Distributed Nonparametric Training Algorithms for Hypothesis Testing Networks

by

John W. Wissinger

B.S. Electrical Engineering, Rice University (1986)
Master Electrical Engineering, Rice University (1988)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

Massachusetts Institute of Technology

May 1994

Author .....
Department of Electrical Engineering and Computer Science
May, 1994

Certified by .....
Michael Athans
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Distributed Nonparametric Training Algorithms for Hypothesis Testing Networks

by

John W. Wissinger

B.S. Electrical Engineering, Rice University (1986)

Master Electrical Engineering, Rice University (1988)

Submitted to the Department of Electrical Engineering and Computer Science
on May, 1994, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

In this report we investigate the nonparametric optimization of Bayesian decentralized binary hypothesis testing (detection) networks. We first adopt a linear threshold parameterization of the decision rules, and study the properties of the cost function which results. We then present several nonparametric training algorithms, and establish sufficient conditions under which they asymptotically achieve the minimum error rate over networks of linear threshold classifiers. The algorithms are implemented in distributed fashion, with the parameters of the decision rules controlled locally at each network node. The methods may be grouped broadly into two classes: model-dependent approaches which comprise sets of coupled and communicating stochastic approximations and model-free approaches requiring only observability of the network output from each node. We interpret all of the algorithms as generalized stochastic descent methods, and investigate their convergence. Finally, we demonstrate that several of the algorithms admit asynchronous implementations.

We suggest that the mathematical models in this study provide a useful paradigm for the study of adaptation in uncertain distributed environments such as those characteristic of human decision making organizations, sensor networks, and even biological neural networks.

Thesis Supervisor: Michael Athans
Title: Professor of Electrical Engineering

Acknowledgments

I would first and foremost like to thank my thesis supervisor, Professor Michael Athans, who provided many things for me in my time at MIT; his friendship and concern for my well-being provided me support, his interest and ideas stimulated my thoughts, and his belief and patience allowed my ideas to mature. I have been privileged to have associated with him.

I would also like to thank my readers, Professors John Tsitsiklis and Alan Willsky of MIT, and Professor Krishna Pattipati of the University of Connecticut. Specifically, I wish to thank Professor Tsitsiklis, whose own thesis and research work provided much of the impetus for the work here, for several helpful discussions, for clarifying some technical points, and for patiently reading this report. I wish to thank Professor Willsky for his reading, helpful comments, but most of all for his good-natured support and friendship throughout my research effort. Professor Pattipati I wish to thank for several visits I made to University of Connecticut, and for his careful reading of this report and subsequent helpful comments. A special debt of gratitude also goes to Dr. David Castañon of ALPHATECH, Inc., who gave this document a detailed reading, and provided many helpful comments as well.

On a personal level, by far my greatest debt is to my lovely wife Elahe, who is my soul and my strength. I thank her for giving me the freedom of mind to complete this work, for her selfless support, even when demands on her own time were enormous, and for her lighthearted nature which never fails to lift my spirits. I also owe great thanks to my mother and father, without whose love and attention through my seemingly endless academic years, I would surely never have reached this point in my life, and to my brother and sister who have shared in my education from the very beginning.

To my friends and colleagues in the office, I am sorry I cannot give each one of you individual attention for the many ways in which you contributed to the completion of this thesis. I can simply say that for their invaluable companionship I would like to thank Frank Aguirre, Mike Ashburn, Ed Bielecki, Mike Daniel, Marcos Escobar, Jose

Lopez, Wesley McDermott, Jason Papastavrou, Steve Patek, Jim Walton, and Peter Young. But special thanks are due to Alan Chao, who spent significant time with me discussing my ideas, and arguing proofs on the blackboard, and to Joel Douglas, who over the years has been extremely helpful with Latex and the computer aspects of my work in general.

A debt of gratitude also goes to the support staff in LIDS, particularly Fifa Monserrate, with whom, unfortunately, most of my interaction concerned my last minute problems, but who always cheerfully helped me out.

This research was carried out at the M.I.T. Laboratory for Information and Decision Systems and was supported in part by the National Science Foundation through an NSF Graduate Fellowship, under grant NSF/Ir1-8902755 (under a subcontract from the University of Connecticut), under grant NSF/ECS-9216531, and by EPRI under contract 8030-10.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preliminary Discussion

We begin this report with a discussion of the main components of the title: *hypothesis testing*, *networks*, *nonparametric training algorithms* and *distributed*. Although the primary purpose of the discussion is to clarify the specific context in which each term is intended, we also wish to overview the setting in which the contributions of this report are made. At present the discussion is kept at a high level; presentation of the mathematical formalism required to make the statements precise begins in Chapter 2.

Hypothesis Testing: A hypothesis test is a particular type of statistical decision test in which the objective is to decide which of a set of mutually exclusive hypotheses was active in the production of some random observable quantity. In this report, we restrict ourselves to the simplest case, in which there are only two competing hypotheses. This is referred to as binary hypothesis testing, or binary detection. The problem setting is as follows. A decision agent or decision maker (DM)[1] observes an *environmental random variable* for which the statistics are available to the DM. In the binary case, the environment assumes one of only two possible values, for example

[1]Decision makers may also be referred to as network nodes, sensors, or classifiers depending on context, although we generally adhere to the terminology DM throughout this report.

as the result of the flip of a biased coin. The DM attempts to decide between the two competing hypotheses on the true value, each corresponding to one of the two possible states of the environment. The information to which the DM has access for making the decision is a noise-corrupted scalar *observation* of the environment, the statistics of which are also known to the DM. These statistics are typically provided in the form of conditional probability density functions[2] describing the distribution of the observation under each hypothesis. Using this noise-corrupted observation, as well as prior knowledge of the statistics involved, the DM makes a decision regarding which of the two possibilities was the true state of the environment. This decision is made according to some prespecified criterion. The function which maps the DM's observation to its decision is known as a *decision rule.* The exact form of the decision rule which is optimal will depend on the particular decision criterion.

Networks: Assume that a collection of DMs is assembled in this setting, each of which receives a different observation of the same environment. In general we allow the quality of each DM's observation to vary, i.e., the noise on each DM's observation can be different.

Suppose that a new *decentralized*[3] version of the above decision problem is formulated in which the set of DMs jointly endeavors to make an overall *team*[4] decision, for example through a specified *primary DM*, in order to optimize a given measure of team performance. A reasonable choice is to have the team attempt to minimize the probability that the primary DM makes an incorrect decision on the true hypothesis. In the absence of any further constraints, the provably optimal action of the collection of DMs is to relay all of the raw observations to the primary DM and let it decide for the team. This *centralized* solution is both mathematically and intuitively the

[2]The existence of the conditional density functions is a fundamental assumption of hypothesis testing.

[3]Throughout this report we will refer to information as being decentralized and algorithms being distributed. This choice is merely a convention which suits our point of view, and is not intended to imply a fundamental difference in meaning between the terms.

[4]The term team is intended to convey the notion that the collection of DMs possesses a common goal.

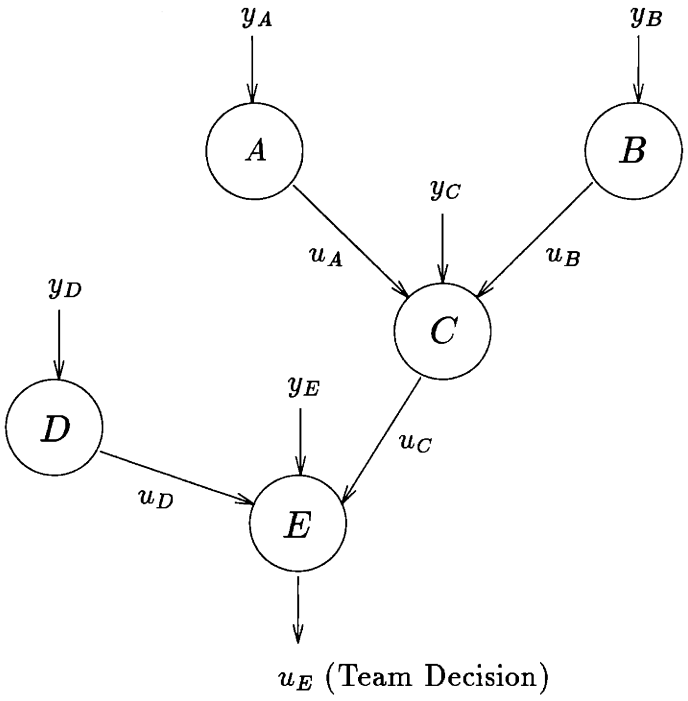


Figure 1-1: Decentralized Decision Network

best since a single DM is permitted to have access to all the available unprocessed information with which to make a decision.

Now consider a modification to this problem setting in which the team of DMs is assembled into an *organization* or *network* with prespecified communication protocols. In particular, suppose each DM is allowed to communicate information to some subset of the group's members through unidirectional communication links. This scenario is depicted graphically in Figure 1-1 for a tree-configured network, where each node of the graph represents a DM, the directed arcs between nodes represent communication links between DMs, and the external input to each node represents a local observation of the environment. There exists a preimposed order to the sequence of communications between DMs which is imposed by the topology of the network. Each DM will take into account what was communicated to it by its immediate predecessors as well as its own observation when it decides what it should communicate to its immediate successors. The decision rule of each DM is now a function from its observation, and any communicated messages it has received, to the message it will communicate to its

successors. For a tree-type topology such as that shown, the primary DM is normally taken to be the root node. The primary DM then maps its observation and incoming messages to the overall team decision on the true hypothesis.

We now make a final modification to the problem setup. Suppose that rather than allowing the DMs to transmit an arbitrary amount of information on the communication links, which would permit the centralized solution to be implemented by the primary DM, communication is restricted, so that each DM is only allowed to transmit to its successors a message out of some finite message set. For example, each DM might be allowed only a single bit with which to communicate with its neighbors. Then the problem of deciding what each DM should transmit, i.e., the form of each DM's decision rule, in order to optimize some measure of organizational performance is what is commonly referred to as the *decentralized* binary hypothesis testing (detection) problem[5]. The problem is a specific brand of *team decision problem* [26]. Throughout this report we will refer to this problem as an organizational or team decision problem, with the context of decentralized binary hypothesis testing understood.

Note that in this setting, information about the environment is distributed throughout the organization and is not allowed to be transmitted freely from DM to DM. As a result, the information of each DM is partial or incomplete. In addition, only prescribed communication pathways are allowed as a result of the fact that, in general, each DM is connected to only a subset of the team's members. These restrictions, in combination with the quantization of the messages, effectively preclude the organization from achieving the same performance as the centralized solution. Thus, for a given decentralized hypothesis testing problem, the corresponding centralized problem always provides an unattainable lower bound on performance.

Nonparametric Training Algorithms: In a broad sense, the term training refers to the dual processes of acquiring information and acting on that information to make favorable adjustments in the execution of some action. In the problem of this

[5]In the sequel we will frequently abbreviate decentralized binary hypothesis testing as DBHT

report, information is acquired through examination of correctly classified sample observations. Performance on these samples is then used to evaluate and subsequently improve the action of team decision making, i.e., to adapt the team decision rules.

In the mathematical formulation of the problem, the quality of the team decision process for a given network topology[6] is quantified by an associated cost functional or scalar-valued performance measure. The quality of the decision process is evaluated on the basis of the corresponding values attained by this functional. If good decision making corresponds to low values of cost, then the best possible decision making process is one which incurs minimum cost. Thus, determining the best decision process requires *optimization* of the cost functional with respect to the network decision rules. The cost we focus on in this report is the Bayes risk, with the majority of attention paid to the special case of the probability of team error.

The difficulty is that optimization of the cost functional in a hypothesis testing environment requires complete knowledge of the underlying statistics of the test. In many instances this information may not be available. The problem discussed in this report assumes that the probability structure of the hypothesis test is unknown to every DM in the team. In some cases it is also assumed that the network topology is unknown. In either case, there is insufficient information available regarding the probability structure of the problem to enable the DMs to analytically compute the value of the criterion function or any of its derivatives. The probability structure of the problem must be inferred through repeated examination of correctly classified observations.

The training problem we have posed is a stochastic optimization. Techniques for performing this optimization in the face of unknown statistics are termed *nonparametric* optimization techniques. Thus, our training algorithms are nonparametric stochastic optimization techniques, frequently referred to as stochastic approximation. In this report we focus exclusively on gradient-based techniques, in which the algorithms all use the data to construct estimates of the gradient. We argue that all

[6]In this report we do not consider optimizing the network topology. For more on this see Papastavrou [43]

of our algorithms possess a generalized stochastic descent property.

In order to apply stochastic approximation to DBHT networks, we must first settle on a suitable parameterization of the decision rules. In this report we choose the simplest such parameterization possible; we parameterize the decision rules as linear threshold rules, or hard-limiting thresholds in observation space. Thus, our training problem could equivalently be described as "threshold learning". Although concerning ourselves exclusively with this parameterization may be viewed as restrictive, it has the advantages of being easily visualized, containing the optimal set of rules for the important case of Gaussian distributions, and being more than rich enough with respect to the goals of this report.

Distributed: The final piece of the title requiring clarification is the term distributed. It refers to the fact that the training algorithms we investigate involve *local* adjustment of the decision rules of each DM. In particular, the decision rule of a given DM is known only to that DM, unless it is communicated through the network to other DMs which may require knowledge of it to perform their updates. Distributed algorithms thus possess added complexity in the form of communication and timing issues. Furthermore, as the training algorithms we consider are of the stochastic approximation variety, our training algorithms consist of collections of coupled, communicating and locally-executing stochastic approximations.

Our primary purpose in considering distributed rather than centralized algorithms is to improve the modeling capabilities of DBHT networks with respect to several areas of application discussed in the following section. In this regard, it is useful to associate each DM with a distinct entity which must communicate to resolve coupling with other DMs and which performs its own computations on locally held parameters. The distributed setting also introduces interesting issues such as how information is distributed throughout the network during training, and timing issues, including the possibility of asynchronous behavior.

In summary, the title of this report indicates that its subject is the problem of

determining, in distributed fashion, the minimum probability of error set of decision rules in a decentralized binary hypothesis testing network of known structure, using only a sequence of correctly classified training examples.

1.2 Motivation

Broadly stated, our interest in this topic is to model and analyze adaptation in coupled organized systems. Toward this goal, we seek to provide in this report a mathematical framework which is rich enough to explore a variety of issues, yet remains mathematically tractable, and which is also qualitatively different from what has been done previously.

The subject matter of this report would seem to be of interest for researchers in many disciplines, including decentralized detection, distributed computation, pattern classification, neural networks, mathematical economics and psychology. It represents a study of DBHT models from a completely novel angle. The problem setting provides a venue for studying the behavior of distributed implementations of classical stochastic approximation algorithms. From the more general point of view of distributed computation, it provides an interesting testbed for examining a variety of fundamental issues such as communication and synchronization. On the AI side, it explores the fundamental limitations of learning in uncertain distributed environments. For obvious reasons this is also pertinent for psychologists who study decision making in organizations, and economists attempting to model optimal strategies in decentralized markets, etc. From the point of view of pattern classification and trainable machines, the work might be said to represent a study of "distributed trainable pattern classifiers". Finally, this work represents a study of training algorithms for a type of network which is qualitatively different from the standard feedforward perceptron neural network, and so would be of interest to researchers in this area as well.

We have chosen decentralized binary hypothesis testing networks as our paradigm because the model is applicable in several areas of interest to us. To impart some

appreciation of the utility of the model, we describe in the following sections several of these applications. Review of the literature concerning these applications appears subsequently in Section 1.4.

1.2.1 Organizational Decision Theory

A setting of particular interest to us is the modeling of human decision making organizations, for example those which arise in command and control environments, or in economic market systems. Both types of organizations often comprise collectives of dispersed rational[7] decision makers that make and share local decisions on the basis of incomplete information, but which are ultimately interested in the satisfaction of some overall team objective. Some of the common elements shared by DBHT networks and structured organizations are already clear in Figure 1-1. The DBHT models possess hierarchical structure, as well as restricted communication and partial information.

Of course, any mathematical model claiming to represent a human decision making process, particularly one concerned with decisions of the yes/no or zero/one variety, is doomed to be inadequate, and will always be subject to criticism on grounds that it represents very few of the factors which ultimately influence a human decision. However, simple mathematical models often prove useful media for studying complex systems, particularly when they are are able to represent what are deemed to be some of the fundamental and defining properties these systems. It is in this sense that DBHT networks provide an especially suitable paradigm for the study of decentralized decision making. These models are mathematically clean and simple to describe, yet despite this simplicity there exists a surprisingly rich class of examples which exhibit interesting behavior. More importantly, the models display behavior, such as hedging, which conforms to intuition about how the members of human organizations actually behave. The decision rules of decision makers in a trained hypothesis testing

[7]Our formulation of decentralized decision making requires that the behavior of each member of the collective be rational when viewed in the context of optimizing a common goal. Issues such as competition are usually addressed by alternative frameworks such as game theory.

team are generally in marked contrast to those of DMs solving the same problem in isolation. This effect results from the coupling between DMs in the team setting. In addition, the centralized counterpart to a given decentralized binary hypothesis testing problem is trivially solved and the solution has a very simple structure. Although the decentralized problem is clearly more difficult to solve, under certain assumptions the mathematical form of the optimal decision rule at each DM is remarkably similar to that of the centralized problem. Hence, in contrast to decentralized versions of many other problems, decentralization in the DBHT problem does not necessarily preclude the problem from having exploitable structure. Indeed, the complexity in solving the DBHT problem is attributable in large part to the decentralization itself, and is not purely a result of combinatorics. As we will see, even very small structures, such as two or three member networks, exhibit complexity of this non-combinatorial variety.

Because of these properties, a sizable body of literature has developed in which DBHT models are used to model organizational decision making. To the present, this modeling effort has focused on the static aspects of the problem, such as parametric studies which investigate the behavior of optimal solutions under various conditions. We believe the present study adds learning dynamics to the modeling effort, making it a suitable framework to develop a normative theory of team training. It is because of the lack of a normative theory which addresses the inherent difficulties and fundamental limitations of learning in distributed unknown and uncertain environments that an understanding of the processes by which organizations adapt to improve performance has been slow in coming. We believe the study undertaken in this report may provide a framework in which such a theory may be developed.

1.2.2 Decentralized Detection and Surveillance Systems

One of the original motivations of the DBHT problem was in the area of multisensor detection problems in which there are a collection of sensors, perhaps geographically distributed, each receiving observations regarding the presence or absence of some target. It may be desired to place some of the computational burden at each sensor's

locale, rather than have all the sensors communicate their raw data to a central location for processing. Reasons for this may be to avoid inundating the central processor with information, or to refrain from producing an abundance of easily detectable communication in a battlefield setting.

To date, the vast majority of work in the field of decentralized detection has centered around formulating specific problems and then deriving optimal solutions to those problems. Certainly, deriving the form of the optimal solution for a particular network topology is a nontrivial exercise. As is presented in detail in Chapter 2, the optimal decision rule at each DM in the organization is coupled with the decision rule being used by every other DM. This results in the optimality criteria of these problems typically being expressed in the form of person-by-person optimality conditions, which specify necessary conditions for optimality of each decision rule given that the remaining decision rules are held fixed. No closed-form analytic expression for decision rules satisfying the necessary conditions for optimality exists. Furthermore, deriving the sufficient conditions for optimality is analytically extremely difficult. Computation of person-by-person optimal decision rules must be done numerically, and may require a slowly converging iterative procedure, solution of a difficult constrained nonlinear optimization problem, or equivalently, solution of a spatial dynamic programming problem. Furthermore, the complete statistics of the problem must be known to each DM in order for the team to collectively compute the optimal set of decision rules. If there is prior bias in the data, that bias must be accounted for explicitly in order to compute the optimal solution. The statistics of the measurement noise at each DM must be accounted for as well. Aside from presupposing that the environmental statistics have been accurately modeled, it is clear that should any of these statistics change, the solution would no longer remain optimal.

One motivation for the study of this training problem is to overcome some of the inherent difficulties involved with computing the optimal solution. If a large body of representative data is available[8], the techniques described in this report may be used to adapt the sensor thresholds to their optimal values without any modeling of the

[8]The data must satisfy certain technical assumptions described in Chapter 5.

statistics required. Adaptations of the methodology may be able to provide adaptive solutions in changing (nonstationary) environments.

1.2.3 Trainable Pattern Classifiers

The ideas of learning and adaptation have historically played a central role in the development of trainable pattern classifiers. An important problem in statistical pattern classification is the problem of optimizing an adaptive pattern classifier to minimize the probability of classification error for two nonseparable pattern classes. When the classifier is parameterized by a linear threshold rule, this problem corresponds to a special case of our team problem for a team of one, i.e., a single decision maker. Our problem can be viewed as a collection of communicating classifiers viewing different realizations of some common underlying pattern, and communicating in an attempt to resolve the pattern. Thus, the training problem of this report could be said to correspond to the nonparametric optimization of a set of coupled binary pattern classifiers.

To our knowledge, no work on the distributed training of decentralized classifiers, or classifiers in which the observation space has been parsed by geography and the computation is performed locally, has appeared.

1.2.4 Biological Neural Systems

An argument can be made that DBHT networks provide a more biologically plausible representation of real biological neural networks than many alternative mathematical models, particularly perceptron neural networks.

We associate each node of the graph depicting a DBHT network with a neuron, and each communication link as a synapse between the neurons. As discussed below, the models are able to represent some of the key properties of biological neural networks, such as all-or-nothing firing (0-1 output) and the ability of each neuron to excite or inhibit other neurons. In addition, the models are probabilistic in nature, a property that is widely thought to be essential in this modeling effort.

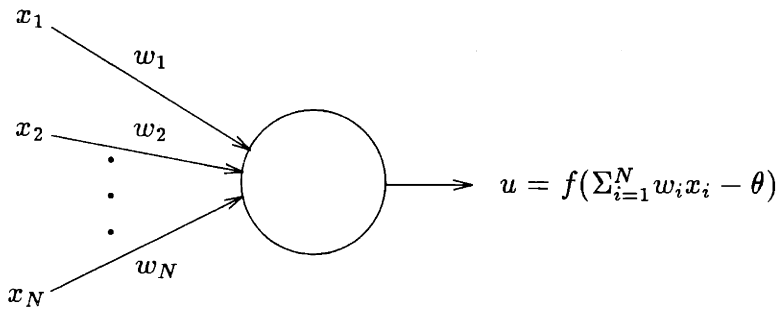


Figure 1-2: Perceptron unit

Relations to Neural Networks

Our initial effort on this problem was prompted by the striking structural similarity of DBHT networks to perceptron neural networks (PNNs). We set out to investigate whether training algorithms similar to back-propagation could be derived which could train a DBHT network to the optimal decision rules. With respect to this endeavor, we must address a reasonable question: What do we hope to learn from developing training algorithms for DBHT networks that cannot already be learned from the study of feedforward perceptron neural networks?

To answer this, we briefly consider some of the similarities and differences between DBHTNs and PNNs. With regard to the similarities, both types of network consist of coupled collections of interconnected nonlinear computational "units". However, there are some qualitative differences in the operations performed by each type of unit.

Figure 1-2 illustrates a single unit of a perceptron network [35]. The output u of a perceptron unit with real-valued inputs $x_1, \ldots, x_N$ is given by

$$u = f\left(\sum_{i=1}^{N} w_i x_i - \theta\right) \tag{1.1}$$

where the w_i are scalar weights multiplying the inputs x_i, θ is a scalar threshold, and f is a nonlinear scalar-valued activation function, also referred to as a gain, transfer,

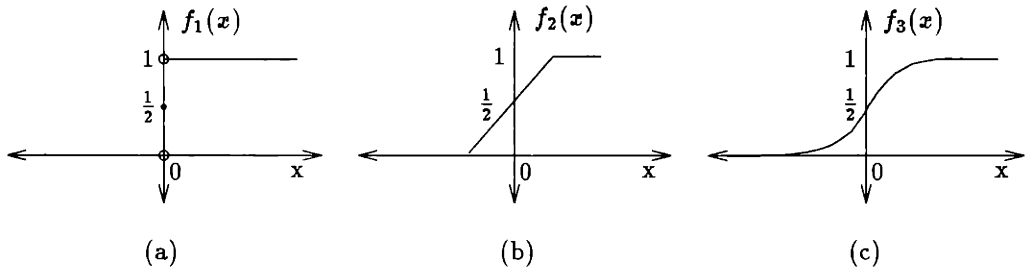


Figure 1-3: Typical Nonlinearities: (a) hard limiting threshold, (b) threshold logic, (c) sigmoid.

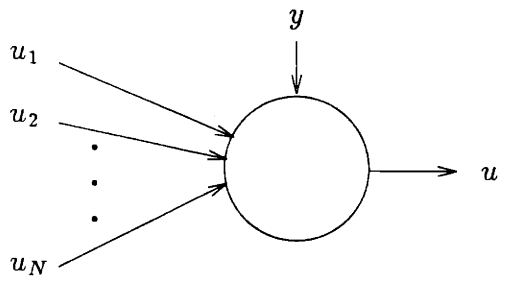


Figure 1-4: DBHT unit

or squashing function[9]. Some typical activation functions are shown in Figure 1-3. The operation performed by the unit is deterministic once the inputs to the node are specified.

In contrast, Figure 1-4 illustrates the operation performed by a unit for the special case of DBHTNs we study in this report. It is derived in Chapter 2. The output u of the DBHT unit (DM) derives from the test

$$y \underset{u=0}{\overset{u=1}{\gtrless}} \begin{cases} \theta_1 & \text{if } u_i = 0,\ \forall i, i = 1, ..., N \\ \theta_2 & \text{if } u_1 = 1,\ u_i = 0, i = 2, ..., N \\ \vdots & \vdots \\ \theta_{2^N} & \text{if } u_i = 1,\ \forall i = 1, ..., N \end{cases} \tag{1.2}$$

[9]Note that the scalar parameter θ may be incorporated into the sum directly by adding a dummy input $x_0 = -1$ and taking $w_0 = \theta$.

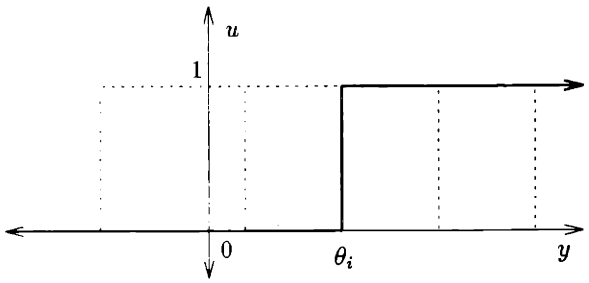


Figure 1-5: DBHT Unit Operation. The bold indicates that observation threshold θ_i has been selected to generate the output. Other possible choices of the threshold are indicated by the dotted curves.

where y is the random observation, u is the binary-valued decision output, and $\{\theta_i\,; i = 1, \ldots, 2^N\}$ is the set of observation thresholds. The output u of a DBHT unit is the result of one of a collection of possible statistical threshold tests on y where the appropriate test is selected by the particular combination of the N incoming binary-valued messages from upstream DMs. The operation of the test is depicted in Figure 1-5.

The particular test or mode of operation is selected from the total of 2^N possibilities. The output of the unit is therefore random given the inputs from other nodes, and is deterministic only once the exogenous input y is specified as well.

In the case of PNNs, the influence of one unit on another is quantified by adaptable multiplicative weights on the interconnections. Specifically, consider the typical topology for a (3-layered) PNN shown in Figure 1-6. The only units receiving external inputs (observations) are the input layer neurons which output to a hidden layer. The output neurons produce the final network output. Internal to the network, the outputs of the first layer nodes are multiplied by scalar weights, and then used as inputs to the hidden layer.

In the case of DBHTNs, the coupling is of a different nature. Units in DBHTNs are coupled with one another through the selection of modes of operation (decision thresholds or operating points). This makes the influence of one unit on another apparent in a way that allows interpretations to be made, whereas interpreting the meaning of the weights in a PNN appears difficult. Specifically, it is not clear that

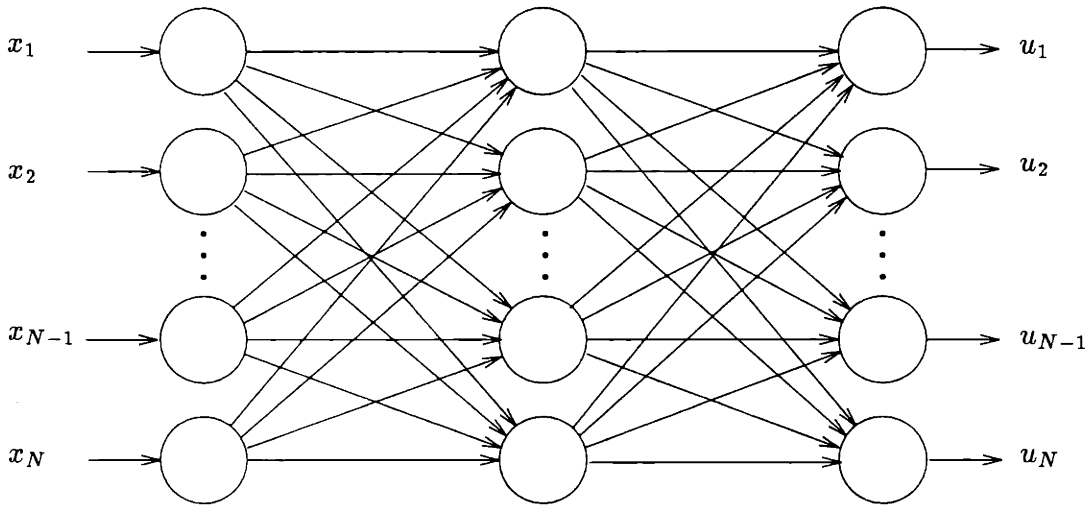


Figure 1-6: Typical 3-Layer Perceptron Network

the weights in a PNN are capturing information about the function the network is approximating in a way that is meaningful from the point of view of drawing behavioral interpretations. However, while it may not be intuitively clear what it means for a weight in a PNN to increase from one training iteration to the next, it certainly is clear what it means for an observation threshold in a local hypothesis test to shift up or down. The ability to interpret behavioral shifts in the network as the parameters are adapted is critical if the network is to be used for organizational modeling. In DBHTNs, all units receive external input in the form of noisy observations. Thus, all the units are affected by the external environment, a property which seems reasonable for massively parallel learning structures, and this allows for the units to be interpreted as having varying degrees of competence. Finally, as we will see in Chapter 2, it is possible in the context of DBHTNs to study arbitrary tree-type topologies which allow for a more flexible study of the relationship between network topologies and training algorithms. On the negative side, we cannot investigate networks with feedback using the techniques of this report, and hence we leave this extension for future work.

As a final point of comparison, both types of networks are trained with labeled training data to optimize some organizational measure of performance. As we will see, while training algorithms for DBHTNs often resemble training algorithms for

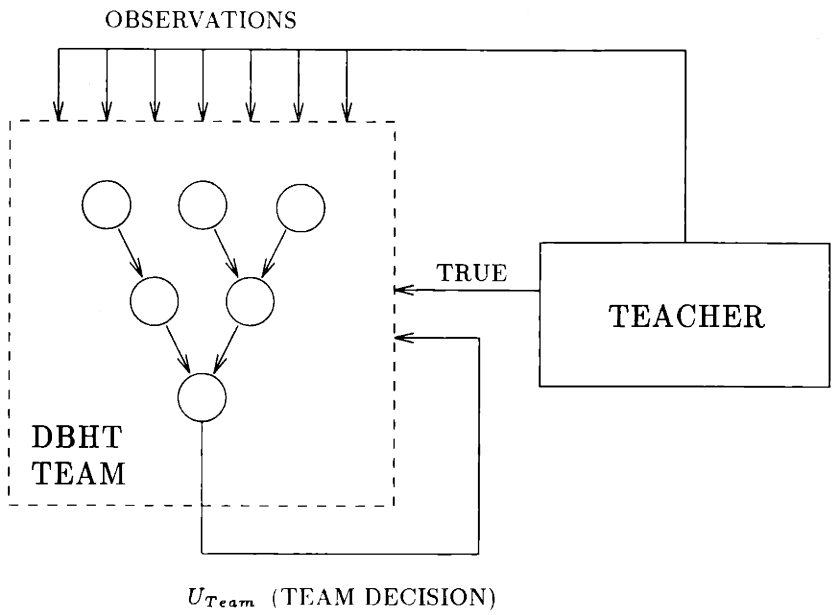


Figure 1-7: Learning System; Global Feedback from Network Output. The teacher provides ground truth to the team by indicating the true acting hypothesis for each set of network observations.

PNNs, there are some pronounced differences.

To summarize, we believe that there is sufficient reason to study DBHTNs due to their interpretive value, and we suggest that the DBHT models of this report, or some variation thereof, might be a source of simple mathematical models which better capture the essential components of biological processes of learning and adaptation.

1.3 Model and Problem Statement

We investigate two models of training in this report. The first model may be represented by the systems diagram in Figure 1-7. The diagram depicts the DBHT network (team) interacting with a so-called "teacher" or "expert" which provides the team with a set of observations along with the true acting hypothesis for the observation set. In other words, the role of the teacher is to make available ground truth to the team. Notice that the model also indicates that the team decision output is fed back to each member of the team as well.

Because of the binary nature of the hypothesis test, the model may be equiva-

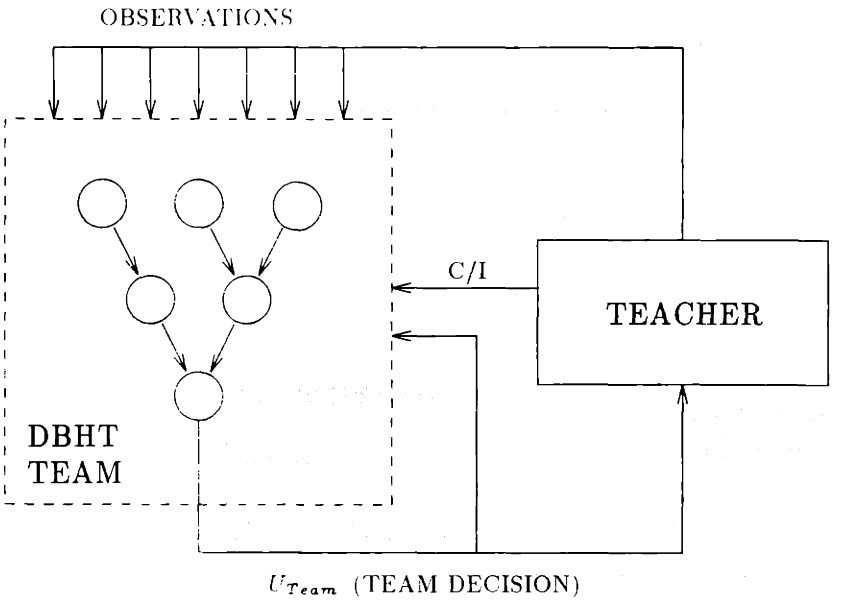


Figure 1-8: Equivalent Representation as a Performance Feedback Loop. Teacher labels as correct or incorrect each decision output of the team.

lently represented as a feedback loop as shown in Figure 1-8, in which the teacher observes the output of the team, and then provides "performance feedback" in the form of labeling as correct or incorrect each decision output of the team. The equivalence of correct/incorrect and ground truth comes from the fact that knowledge of correct/incorrect and the team decision is sufficient to infer the acting hypothesis in the binary problem. Again it is assumed that both the decision of the primary DM and the label are made available to every network DM.

The net effect of the equivalent schemes shown in Figures 1-7 and 1-8 is to provide the team with a set of training examples at each instant of time[10], where each training example consists of a complete set of observations for the network DMs along with the desired network output, i.e., the acting hypothesis for the set of sensor observations obtained at that time instant. In addition, each member of the team is assumed to observe the corresponding team decision. Due to the existence of the teacher/expert, the model is referred to as a *supervised learning* model, or *learning with a teacher.*

An interesting property of the training algorithms which are derived based on this

[10]We assume time evolves discretely

model is that they are *model-free*. No DM requires a representation of the rest of the network to update its decision rules. In particular, each DM may be oblivious to the overall topology of the network to which it is connected. Furthermore, these algorithms require no communication between the DMs to optimize the team decision rules. These properties result from the fact that, under the assumption of feedback from the team output to each DM, each DM is capable of observing the effect of perturbations of its parameter(s) on the output directly. For those readers familiar with stochastic approximation, these training algorithms are of the Kiefer-Wolfowitz variety, and these assumptions are necessary so that each DM may sample the team cost. Notice that at each instant of time, it is necessary that an entire team decision process be executed. During the training phase, the DMs use each training example to execute a decision process which results in an overall team decision, and are then informed as to the true hypothesis active for that set of data. Using this information, the DMs attempt to adapt their decision rules so that the organization will have continually improved performance with respect to a prespecified performance criterion.

A alternative model of training is also possible, in which each DM makes adjustments to its parameter(s) based only on the correctness of its own local output stream. The process is illustrated in Figure 1-9. This type of training process is necessarily *model-dependent*, because a team decision process does not intercede between the execution of an adjustment by a DM and the observation of the effect of that adjustment. In particular, team decision processes are not executed for these schemes, and therefore no team decision output of the network is observed. Each DM must maintain a representation of the current state of the rest of the network DMs which is sufficiently informative for it to perform the proper updates of its parameter(s). Furthermore, communication is required to continually update these representations as the overall state of the network evolves. Again, for those readers familiar with stochastic approximation, these methods explicitly model the partial derivatives locally, and update based on Robbins-Monro type iterations.

The global problem which this report addresses has several components:

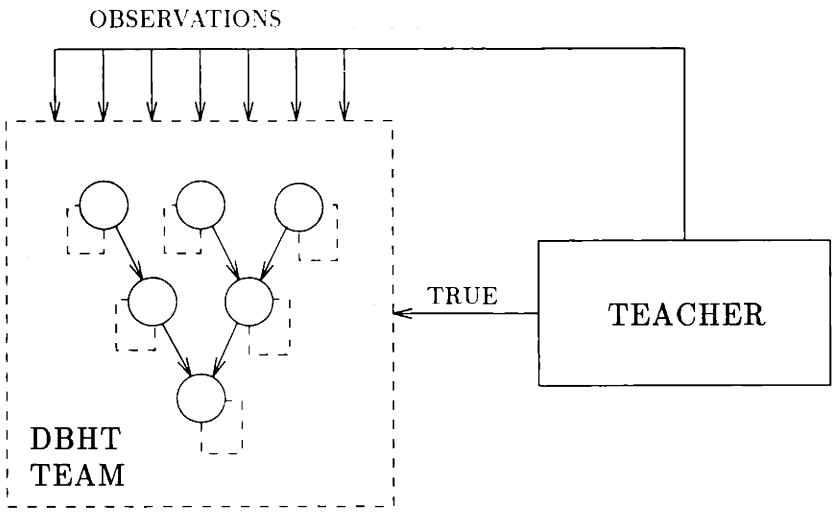


Figure 1-9: Learning System; Local Feedback

- Choose a parameterization of the decision rules which is suitable for nonparametric optimization (linear threshold rules).
- Demonstrate that the team Bayes criterion is sufficiently well-behaved under this parameterization to admit optimization by gradient-based methods.
- Develop nonparametric gradient-based training algorithms, to be implemented in distributed fashion throughout the network, which allow the optimal decision rules to be determined.
- Identify sufficient conditions under which the algorithms yield asymptotic convergence of the decision rules to their optimal values.
- Relax timing restrictions on the distributed training algorithms and identify sufficient conditions for asymptotic convergence of their asynchronous versions.
- Suggest how the modeling framework we have validated can be used to make interpretations in the areas of application already discussed.

We should point out several topics which we do not address in this report. While we suggest a variety of training algorithms, and establish sufficient conditions for

asymptotic convergence for all of them, we do not attempt to compare the algorithms, or provide extensive parametric studies of their performance. In particular, we do not attempt to identify a "best" approach. Our goal was to suggest and validate a variety of approaches. We believe that comparison and parametric studies are better handled in conjunction with rate of convergence analysis, which was beyond the scope of the present study.

1.3.1 Key Issues

We briefly mention some of the questions which are successfully addressed by the methods in this report.

Each distributed training algorithm has associated with it specific information requirements for each DM, (possible) inter-DM communication requirements, and timing specifications according to which the network-wide training must occur, all of which affect the performance of the algorithm.

With respect to information, the key question is: What local information set is required by each DM during training? Specifically, for distributed gradient-based techniques, what information must be made locally available to each DM so that it may compute estimates of its partial derivatives. How accurately must this information be known? How exactly is it obtained in the context of the two models just described?

For algorithms requiring communication, the key issues are summarized by the question "who should communicate to whom, what, when, etc." [66]. In other words, what communication protocol must be established which meets the needs of a particular algorithm? It is of interest to determine how interconnected the communications must be; are there methods for reducing the number of DMs which must communicate?

The issue of how restrictive a timing mechanism is required to ensure convergence of the algorithm is also of interest: Must the DMs make synchronous adjustments to their decision rules? What is the impact if updates are made asynchronously? Do the other DMs have to be informed immediately whenever a particular DM makes

a change to its decision rule? If not, how does the resulting "outdated information" affect the algorithm? How outdated can the information be allowed to become?

In a broader sense, we will focus on whether the intersensor coupling may be effectively resolved through performance feedback, and attempt to distinguish between those difficulties with training which arise as a direct result of the decentralization and those which would arise in a centralized scenario as well. Long term goals, not completely addressed in this report, involve determining the relationship between the effectiveness of certain training schemes and factors such as the network topology, the degree of coupling between the DMs, and the placement of expertise within the network. For example, we would like to determine whether or not some network topologies lend themselves to being more effectively trained than others. Of course, these issues must usually be explored within the context of a specific training algorithm.

1.4 Background Literature

The problem on which this report focuses is multidisciplinary in nature. As a result, the pertinent literature falls into several categories.

Decentralized Detection The decentralized detection problem was first introduced by Tenney and Sandell in 1981 [64], where the optimality of constant threshold strategies for a two-member parallel team was established. Ekchian [19] subsequently derived the optimal decision rules for a variety of small detection networks, as well as a spatial dynamic programming approach for numerically computing the optimal decision rules. Tsitsiklis and Athans [68] established the NP-completeness for a broad class of these problems. Tsitsiklis [65] provides an extensive review of the literature through 1989, so we refer to the interested reader to this work; in addition several results are generalized. Several difficult problems in decentralized detection, such as dependent observations, are addressed by Irving in [27]. A study of the two-member tandem team, on which a significant portion of this report will focus, was undertaken by Poth-

iawala in [52]. A change of perspective was presented by Papastavrou in [43], [44] in which various small topologies were compared to determine whether the performance of some topologies dominates others. Tang [59], and Tang, Pattipati, and Kleinman [61], [62], apply a variety of numerical techniques to the optimization of the team decision rules in the presence of complete statistical information, and suggest an interpretation of the optimization as a deterministic optimal control problem. DBHT networks are investigated as normative models of team decision making in the references by Boettcher and Tenney [12], and Pete, Pattipati, and Kleinman [47], [46], [48].

Stochastic Approximation There is a long history of study on learning problems of the variety we have posed, both Western and Russian, although little of it investigates decentralized problems.

The field of stochastic approximation, which forms the basis of many of our algorithms, has been well-studied since the 50's. Classical stochastic approximation was initiated by Robbins and Monro in 1951 [53] with their seminal paper demonstrating that stochastic (noisy) versions of successive approximation problems could be successfully solved. The method was subsequently shown to satisfy a stronger notion of convergence by Blum in [10]. In 1952 Kiefer and Wolfowitz [30] presented a method for determining the extremum of a function of which only noisy measurements are available. The results of Robbins and Monro, as well as Kiefer and Wolfowitz, were extended to the multidimensional case by Blum [11]. In an important paper by Dvoretzky [18], it was demonstrated that all the previous methods could be interpreted as noisy contraction mappings. Ljung provides an analysis of recursive stochastic algorithms as approximating the solution of an ODE [36]. In this report we also use a result of Kushner [31] regarding the convergence of KW techniques for functions with nonunique stationary points, and discuss in some detail an algorithm suggested by Spall [58].

The Russian school, which was investigating stochastic approximation for modeling learning and adaptation, was led by Tsypkin [69], [70] and Polyak [49], [50]. They provide an interpretation of many iterative stochastic optimization techniques as stochastic descent algorithms, and provide the unifying notion of a "pseudogradient". Further details concerning the approach may be found in the text by Polyak [51].

There are many good expositions on stochastic approximation available. Some of the more classical presentations include the books by Wasan [72], Nevel'son and Has'minskii [40], and Wilde [75] and the survey articles by Sakrison [56], and Kashyap, Blaydon and Fu [29]. More modern presentations include Kushner and Clark [32], and Benveniste et al. [4].

Pattern Classification The literature on adaptive pattern classification and trainable machines contains a good deal of material which is relevant to our training problem. The classical work of Nilsson [41] introduces the issues which arise in trainable pattern classification schemes. The text by Duda and Hart [17] provides an accessible introduction to adaptive pattern classification. Stochastic approximation as a paradigm for more general types of learning has been suggested by Tsypkin in [69],[70], Fu in [21], [22], and Kashyap, Blaydon, and Fu in [29]. Kac [28] analyzes a simple adaptive binary detection scheme. Based on the work of Wassel in [74], Wassel and Sklansky [73] present a stochastic approximation algorithm for the nonparametric training of a one-dimensional binary classifier to minimize the probability of error. In the text by Sklansky and Wassel [57], supervised training algorithms for centralized binary classifiers are analyzed at length. A more general look at the use of stochastic approximation techniques for the nonparametric training of minimum probability of error binary classifiers is presented by Fritz and Gyorfi in [20], and Do-Tu, Installe in [14].

Distributed Computation The textbook by Bertsekas and Tsitsiklis [6] provides a comprehensive introduction to distributed algorithms, particularly distributed

implementations of iterative schemes and the notion of asynchronism. This material is also discussed in Baudet [3] and Bertsekas, Tsitsiklis, and Athans [8].

Distributed asynchronous implementations of stochastic gradient algorithms in particular are covered by Tsitsiklis [66], Tsitsiklis, Bertsekas, and Athans [67], where convergence of a stochastic pseudogradient algorithm is analyzed using martingale arguments, and by Kushner and Yin [33], [34] who attack a similar problem using the ODE approach.

Neural Networks Neural networks are covered in some generality in the textbook by Hertz, Krogh, and Palmer [25], and the survey article by Lippmann [35]. The paper by Rumelhart, Hinton, and Williams [55] introduces the well-known back-propagation algorithm. A learning algorithm similar to those discussed in this report was suggested for use in perceptron neural networks by Nedeljkvic' [39], while philosophically, our approach is similar to that of Dembo and Kailath in [13]. Barnard and Casasent [2] compare various criterion functions for pattern classification in neural networks, and this discussion is pertinent to our work.

1.5 Outline of Report

Our guiding philosophy in organizing the presentation in this report was to introduce additional complexity into the analysis gradually. For instance, discussion of the single DM problem always precedes discussion of the team problem. In a more compact presentation the two would be presented simultaneously. Similarly, discussion of the deterministic optimization precedes discussion of the stochastic optimization, and discussion of the synchronous network training problem precedes the asynchronous case. This organization corresponds to the order in which we researched the problem, and is the order in which we are most comfortable conveying our understanding. With hindsight, the presentation could certainly have been compressed, but we believe it would be less accessible in this form. The resulting effect is that this report reads more like a comprehensive tutorial than a collection of results. In the end, we surren-

dered to this urge completely, and made the document self-contained by including in Appendices virtually all adjunct material necessary for a complete understanding of our research effort.

We frequently opt to convey concepts through specific examples. While this makes for a less abstract presentation, it may also make certain results appear less general than they really are. We try to indicate this where appropriate.

Numerical experiments in this report are intended to be illustrative in nature, and are certainly inadequate as complete characterizations of each algorithm's behavior. The number of algorithms we present, in addition to the number of possible choices of topology, the number of tunable parameters of each algorithm, and the number of combinations of parameters to be chosen for the hypothesis tests, made thorough numerical studies of the algorithms impossible in this presentation. We accordingly focus on a few simple examples which we carry throughout the report.

Because of the multidisciplinary nature of the work in this report, the terminology best suited to describe a concept may differ with context. An example of this would be the terms "team, organization, and network" which, for our purposes are entirely equivalent. However, we feel that one of these words sometimes fits a specific context better than another. We apologize in advance for the drifts in terminology which may result, and have made every effort to be clear in spite of this.

Chapter 2: The Binary Hypothesis Testing Model We mathematically characterize the DBHT model and compare it with its centralized counterpart. We present and discuss the necessary restrictions on the general problem which must be made to guarantee tractable well-structured network problems. We then illustrate the restricted model by way of several examples of small (2-4 member) teams. These small topologies are adequate to illustrate the main features of the model. The coupling of the network decision rules is a central focus of the discussion.

Chapter 3: Optimization using Complete Statistics In this chapter, the underlying structure of the DBHT decision rule optimization problem is revealed,

and methods for exploiting this structure in numerical methods are discussed. We then study a useful parameterization of the network decision rules by linear thresholds, which is particularly well-suited to nonparametric optimization. The differentiability and smoothness properties of the Bayes cost function which result from this parameterization are investigated, and its suitability for optimization by iterative gradient techniques is examined.

Chapter 4: The Single DM Training Problem In this chapter, we formulate the training problem for the single DM case and highlight the relevant issues, all of which are certain to arise for the team training problem as well. Several stochastic approximation-type training algorithms are presented, along with simulations illustrating typical sample paths.

Chapter 5: Synchronous Network Training Algorithms Chapter 5 focuses on the network training problem, under the assumption that activities among the network DMs may be coordinated with respect to a global clock. Several training algorithms are presented which fall into two broad classes, those which require that the network be modeled at each node (model-dependent) and those which do not (model-free). The data processing required by the algorithms in each class, as well as their communication and timing requirements, are discussed.

Chapter 6: Convergence Analysis The asymptotic convergence of most[11] of the algorithms presented in Chapters 4 and 5 is established, using results from martingale convergence theory. The algorithms are all shown to possess a generalized stochastic descent property. The term generalized refers to the fact that the gradient estimates employed by all of the algorithms contain bias which decays asymptotically to zero. It is argued that this bias does not act to destroy convergence.

Chapter 7: Asynchronous Network Training Algorithms This chapter inves-

[11]The remainder are covered in Chapter 7.

tigates asynchronous versions of the previous algorithms, where such versions may be meaningful formulated. Convergence of these asynchronous versions is demonstrated by suitable modification of the martingale methods of Chapter 6.

Chapter 8: Final Remarks We present concluding remarks and suggest several directions for future research.

1.6 Contributions of Report

We indicate here the major contributions of this report, which we break down by chapter.

Chapter 2

- The results of this chapter were known previously, and no novel material is contributed, except possibly the optimal decision rules for the 4-Asym network.

Chapter 3

- Several new interpretations of the optimization of the decision rules are provided.
- A novel methodology is described for deriving the cost for an arbitrary tree structured network with conditionally independent observations, from which the form of the optimal network decision rules immediately follows.
- New results are presented concerning the properties of the Bayes cost and its derivatives which result from a parameterization of the network decision rules by linear threshold rules.

Chapter 4

- Material is assembled concerning the application of stochastic approximation techniques to centralized Bayesian classification problems. This material was sufficient for our needs, and no new results were developed.

Chapter 5

- Seven alternative distributed training algorithms are derived for the DBHT network problem. All seven algorithms represent the novel application of stochastic approximation (SA) ideas in the DBHT setting. Development of the algorithms required the identification and exploitation of some specific properties of DBHT networks. Because the training algorithms are implemented in distributed fashion, many issues applicable to the distributed implementation of SA techniques in general were addressed.

Chapter 6

- A common property is identified which is possessed by all of the previous algorithms, namely a generalized stochastic gradient property, and then a single proof of convergence is provided which encompasses all of the previous algorithms.
- New proofs of convergence are provided for the stochastic approximation methods of references [73],[58].

Chapter 7

- A novel result is provided which indicates that the convergence properties of several of the training algorithms of Chapter 5 are preserved under partial asynchronism. In the process, the class of stochastic approximation techniques known to admit asynchronous implementations is extended to include window and Kiefer-Wolfowitz type algorithms.

Broadly stated, we believe the major contribution of this report to be the creation and subsequent validation of a novel modeling paradigm for exploring the effects of training and adaptation in a decentralized setting. The paradigm is based on the application of the mechanisms of distributed computation to the optimization of the decision rules in a particular class of team decision problem, decentralized binary hypothesis testing.

Chapter 2

The Binary Hypothesis Testing Model

In this chapter, the discussion of Chapter 1 is mathematically formalized. We begin with a brief discussion of the centralized version of the binary hypothesis testing problem at the level of Van Trees [71]. This is primarily for those readers who are unfamiliar with hypothesis testing, although it is also useful for establishing notion and context for the central training problem of this report. We then introduce the decentralized hypothesis testing model and devote significant time to several examples of small networks. These examples are representative of the major topological variations in the class of networks we consider, and evidence the noncombinatorial[1] type complexity that is typical of these problems. We note that more detail can be found in the excellent survey by Tsitsiklis [65], while several of the small teams which concern us here were first examined by Ekchian [19].

2.1 Notational Conventions

We adhere to the following notational conventions throughout this report. Random variables are denoted by upper case letters, while realizations of those variables appear

[1] Meaning the complexity does not arise as a result of large numbers of DMs in the network, but rather from the way they are coupled.

in lower case. For example, if X is a random variable, then its realization is denoted by x, so we would write $\Pr(X = x)$. Notable exceptions to this rule are the use of $H_i, i = 0, 1$ to denote the hypotheses, P_F, P_D which refer to the probabilities of false alarm and detection, respectively, and the criterion functions J_B and P_ϵ which refer to the Bayes risk and probability of error, respectively. These exceptions have been made to correspond to standard usage. Other exceptions include constants, which we also denote with capital letters, as in N for representing the dimension of a vector, M as the number of DMs in a team, and L as the Lipschitz constant. Context will eliminate any confusion in these cases. However, occasionally it will be necessary to use lower case letters for random variables, and in these cases we will indicate the change in notation with a footnote.

Vectors and vector-valued functions[2], both random and deterministic, will be denoted with an underbar, as in $\underline{X}$ and $\underline{f} : \Re^M \mapsto \Re^N$. This convention is chosen so as not to interfere with symbols which frequently have to be placed overhead, such as tilde or hat. Vectors are understood to be columns. Components of a vector will be denoted by subscripts, so that for $\underline{X} \in \Re^N$ we would write

$$\underline{X} = [X_1, X_2, \ldots, X_N]^T \tag{2.1}$$

Time will also be indexed using the subscript k, so that the vector $\underline{X}$ at time k will be written

$$\underline{X}_k = [X_{1(k)}, X_{2(k)}, \ldots, X_{N(k)}]^T \tag{2.2}$$

with the time index shown in parenthesis. The only exception to this will be the use of the random variable H^k to denote the hypothesis at time k. Here, a superscript is adopted so as not to confuse H_k with the standard notation H_0, H_1. In addition, the notation $\|\cdot\|$ will be used to indicate the Euclidean norm throughout this report.

Sets are always denoted with calligraphic letters, for example $\mathcal{G}, \mathcal{T}$, $\mathcal{Y}$, and $\mathcal{F}$.

The conditional probability density function of the random variable Y, given event H_i, is indicated by $p_{Y|H_i}(y|H_i)$. Probability mass functions of discrete random vari-

[2]A notable exception to this is the gradient of a function J, which is written ∇J.

ables are also denoted with small p; for example the prior probabilities $\Pr(H_0)$, $\Pr(H_1)$ are denoted p_0 and p_1, respectively.

The notation $W \sim N(\mu, \sigma^2)$ indicates that the random variable W is distributed normally (Gaussian) with mean μ and variance σ^2, i.e.,

$$p_W(w) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(w-\mu)^2}{2\sigma^2}} \tag{2.3}$$

We also use the error function Φ to express the cumulative distribution of a Gaussian random variable. For example, for the random variable W above we write

$$\begin{aligned} \Pr(W \leq \theta) &= \int_{-\infty}^{\frac{\theta-\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx & (2.4) \\ &= \Phi_\theta(\mu, \sigma) & (2.5) \end{aligned}$$

The expected value of a random variable X is written $E\{X\}$. When taking the expected value of a function of several random variables, we subscript the random variable over which the expectation is taken. For example, if F is a function of two random variables X and Y, the expected value of F taken with respect to X is written $E_X\{F(X,Y)\}$. The conditional expectation of F with respect to X, conditioned on the random variable Y, is the random variable $E_X\{F(X,Y)|Y\}$. The expected value of F with respect to X, conditioned on the realization $Y = y$ is the number $E_X\{F(X,Y)|Y = y\}$. Additional discussion on the properties of conditional expectation may be found in Appendix A.

2.2 The Single DM Problem

We first consider binary hypothesis testing (detection) with a single scalar observation, such as occurs for a single DM. The purpose is to introduce notation as well as give an appreciation for the relationship between the form of the solution for a single DM and the form of solution for the team problem presented in Section 2.4. To avoid being overly pedantic, only the aspects of the theory relevant for subsequent material are presented. Additional information may be found in [71].

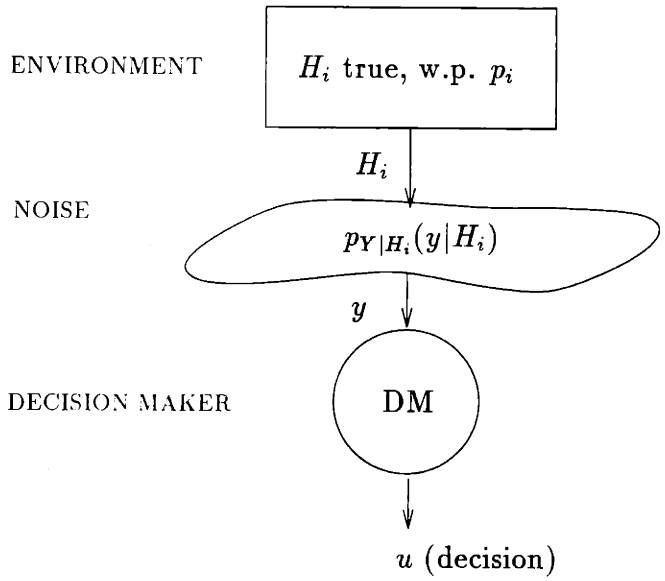


Figure 2-1: Single DM

The basic elements of the binary hypothesis testing problem are as follows (Figure 2-1). There are two distinct and mutually exclusive hypotheses on the state of the environment, H_0 and H_1, each of which occurs with positive prior probability p_0 and $p_1 = 1 - p_0$, respectively. The environment may be viewed as a binary-valued random variable H where

$$H = \begin{cases} H_0 & \text{w.p. } p_0 \\ H_1 & \text{w.p. } p_1 \end{cases} \tag{2.6}$$

The DM attempts to decide on the true state of the environment using a noisy scalar[3] observation of the environment. The hypothesis testing framework assumes the existence of conditional probability density functions $p_{Y|H_i}(y|H_i)$, $i = 0, 1$ according to which realizations of the observations are generated. The noisy observation is then modeled as a random variable $Y \in \mathcal{Y}$ whose realization y is generated according to the corresponding conditional density $p_{Y|H_i}(y|H_i)$, $i = 0, 1$. These densities completely describe the statistical relationship between the observation and the underlying hypotheses, and along with the prior probabilities are assumed known to the DM.

It is assumed throughout this report that Y is a continuous real-valued random

[3]Vector-valued observations are avoided throughout this report for simplicity.

variable. Discrete observation spaces are avoided entirely, due to resulting complications of the optimal decision rules which may result (we will be more specific momentarily). Thus, we take $\mathcal{Y} = \Re$. Furthermore, we assume that the densities are nonseparable, i.e., there is no parsing of the real axis for which all realizations of the observations corresponding to one class are in one region, and all observations corresponding to the other in another.

Let u be a realization of the random variable U denoting the decision of the DM which takes values in the set $u \in \{0, 1\}$, where $u = 0$ and $u = 1$ indicate the decisions H_0 and H_1, respectively. The function which maps the observation of the DM into its decision is known as a *decision rule*, and is to be denoted γ. Thus we write $u = \gamma(y)$ where γ is a function such that $\gamma : \mathcal{Y} \mapsto \{0, 1\}$. The decision is made in order to minimize some cost criterion.

2.2.1 The Optimal Solution

The form of the optimal decision rule γ depends on the particular decision criterion which has been specified. In this report, we are concerned exclusively with minimizing the Bayes risk[4]. This criterion assigns fixed nonnegative costs to each of the four possible decision-event outcomes, and measures the performance of the rule by its expected cost. The cost is a bounded function C defined from decision-event space to the real numbers, i.e., $C(U, H) : \{0, 1\} \times \{H_0, H_1\} \mapsto \Re$. The most general Bayes formulation quantifies the performance of a decision rule γ by the quantity

$$\begin{aligned} J_{Bayes}(\gamma) &= E_{U,H}\{C(U, H)\} \\ &= E_{Y,H}\{C(\gamma(Y), H)\} \\ &= \sum_{j=0}^{1} p_j E_Y\{C(\gamma(Y), H)|H = H_j\} \\ &= \sum_{i=0}^{1}\sum_{j=0}^{1} C(U = i, H = H_j) p_j \Pr(\gamma(Y) = i|H_j) \\ &= C(0, H_0) + C(1, H_1) \end{aligned}$$

[4]The primary alternative is the Neyman-Pearson criterion [71].

$$+[C(1,H_0)-C(0,H_0)]p_0\Pr(\gamma(Y)=1|H=H_0)$$
$$+[C(0,H_1)-C(1,H_1)]p_1\Pr(\gamma(Y)=0|H=H_1) \qquad (2.7)$$

Note that a decision rule γ minimizes $J_{Bayes}(\gamma)$ if and only if it minimizes the related performance $J_B(\gamma)$ given by

$$J_B(\gamma)=\lambda_0 p_0\Pr(\gamma(Y)=1|H=H_0)+\lambda_1 p_1\Pr(\gamma(Y)=0|H=H_1) \qquad (2.8)$$

where $\lambda_0=[C(1,H_0)-C(0,H_0)]$ and $\lambda_1=[C(0,H_1)-C(1,H_1)]$. This form also arises when there are costs on each type of error, and no penalty for correct decisions. Consequently, in the remainder of this report we use J_B rather than J_{Bayes} to represent the Bayes criterion, despite it not being the most general formulation. It is generally assumed that the cost of making an error is strictly higher than the cost of not making an error, namely that

$$C(1,H_0)>C(0,H_0)$$
$$C(0,H_1)>C(1,H_1) \qquad (2.9)$$

so that λ_0 and λ_1 are positive quantities.

We are frequently concerned with a special case of the Bayes criterion resulting from the particular choice of decision-event costs

$$C(i,H_j)=\begin{cases}1 & \text{if } i\neq j\\ 0 & \text{else}\end{cases} \qquad (2.10)$$

This special case is known as the minimum probability of error criterion, since correct decisions are unpenalized and both types of errors receive unit penalty. This assignment of cost corresponds to the indicator function for an error. We denote this criterion with the special symbol P_ϵ. From (2.8) it follows that

$$P_\epsilon = J_B\Big|_{\lambda_0=\lambda_1=1} \qquad (2.11)$$

and the probability of error corresponding to γ is expressible as

$$P_\epsilon(\gamma) = p_0 \Pr(\gamma(Y) = 1|H_0) + p_1 \Pr(\gamma(Y) = 0|H_1) \tag{2.12}$$

The problem of determining the optimal Bayesian decision rule can then be stated precisely as follows.

Problem 2.1 (Minimum Bayes Risk Binary Hypothesis Testing)
Given the conditional probability density functions $p_{Y|H_0}(y|H_0)$, $p_{Y|H_1}(y|H_1)$, positive prior probabilities p_0, p_1, and the positive (bounded) costs λ_0, λ_1, determine the decision rule $\gamma : Y \in \Re \mapsto \{0, 1\}$ which satisfies

$$\gamma^* = \arg\min_{\gamma \in \mathcal{G}} J_B(\gamma) \tag{2.13}$$

where $\mathcal{G}$ denotes the set of all possible decision rules.

This optimization implies a search over the large set of functions $\mathcal{G}$. However, we find that the optimal rule actually lies in a very structured set, making the optimization (2.13) tractable.

For the Bayes Risk criterion, it is a well-known fact [71], [76] that the globally optimal[5] decision rule takes the following special form. If we define the quantities

Definition 2.1 (Likelihood ratio)

$$\Lambda(y) \triangleq \frac{p_{Y|H_1}(y|H_1)}{p_{Y|H_0}(y|H_0)} \tag{2.14}$$

and

Definition 2.2 (Likelihood ratio threshold)

$$\eta \triangleq \frac{\lambda_0 p_0}{\lambda_1 p_1} \tag{2.15}$$

[5]Satisfying both necessary and sufficient conditions for optimality.

then the optimal solution is given by the so-called *likelihood ratio test* (LRT):

$$\Lambda(y) \underset{u=0}{\overset{u=1}{\gtrless}} \eta \tag{2.16}$$

Decision rules of this form, where the likelihood ratio is compared with a constant threshold, are known as *threshold rules* [65]. Thus, we find that the optimal decision rule γ^* for Problem 2.1 is actually contained in the class of threshold rules, which, under the assumption that the statistics are given, admits parameterization by the single scalar parameter η.

There are several points to emphasize concerning rule (2.16). Since y is the realization of a random variable Y, the likelihood ratio defined in (2.14) is a function of a random variable, and is therefore also a random variable. However, once the observation y has been specified, the test becomes deterministic. Also note that the threshold η is a constant with value specified by the prior probabilities and costs. Equation (2.16) may be interpreted as parsing the observation space $\mathcal{Y}$ into disjoint regions, so that to each possible value of the observation y a corresponding assignment of decision is made.

There is a special case of hypothesis test for which the optimal test (2.16) is in fact equivalent to a *linear* threshold test of the form

$$y \underset{u=0}{\overset{u=1}{\gtrless}} \theta \tag{2.17}$$

where θ is a constant observation threshold. Notice that the conditional densities no longer appear explicitly in this decision rule, and the class is parameterized by the single scalar parameter θ. This is the class of decision rules to which we would like to a priori restrict our search in the training problem. A special case for which a linear threshold test is optimal is the so-called Gaussian detection problem, which concerns deciding which of two *known* constant signals, μ_0 or μ_1, is present in zero-mean additive Gaussian noise. For the special case of $\mu_0 = 0$, this problem corresponds to the problem of deciding whether or not a target is present or absent in a radar signal,

or the problem of detecting the presence or absence of a known signal in noise in a communications context. Hence the term detection. For this case, the observation Y is a continuous real-valued random variable of the form

$$Y = \begin{cases} \mu_0 + W & \text{if } H = H_0 \\ \mu_1 + W & \text{if } H = H_1 \end{cases} \tag{2.18}$$

where $W \sim N(0, \sigma^2)$. In other words, for this problem the observations are distributed according to the conditional distributions

$$p_{Y|H_i}(y|H_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-\mu_i)^2}{2\sigma^2}} \tag{2.19}$$

for $i = 0, 1$.[6]

To derive test (2.17), note that since the decision regions specified by the LRT (2.16) are unaffected by the application of a strictly monotonically increasing function to both sides of the inequality, and both sides are nonnegative, the natural log function $\ln(\cdot)$ may be used to reduce (2.16) to the equivalent test

$$y \underset{u=0}{\overset{u=1}{\gtrless}} \theta \tag{2.21}$$

where the observation threshold θ is given by the closed form expression

$$\begin{aligned} \theta &= \frac{\sigma^2}{\mu_1 - \mu_0} \ln \eta + \frac{\mu_1 + \mu_0}{2} \\ &= \frac{\sigma^2}{\mu_1 - \mu_0} \ln \left(\frac{\lambda_0 p_0}{\lambda_1 p_1} \right) + \frac{\mu_1 + \mu_0}{2} \end{aligned} \tag{2.22}$$

For this value to be finite it is of course necessary that $0 < p_0 < 1$, $\lambda_0, \lambda_1 \neq 0$, and

[6]The Gaussian detection problem is not the only problem for which linear threshold rules are optimal. For example, linear threshold rules are also optimal for the case

$$Y \sim \begin{cases} \tau_0 e^{-\tau_0 y} u(y) & \text{if } H = H_0 \\ \tau_1 e^{-\tau_1 y} u(y) & \text{if } H = H_1 \end{cases} \tag{2.20}$$

where $u(y)$ is the unit step, and $\tau_0, \tau_1 > 0$, so that the hypotheses are on exponential distributions with different parameters.

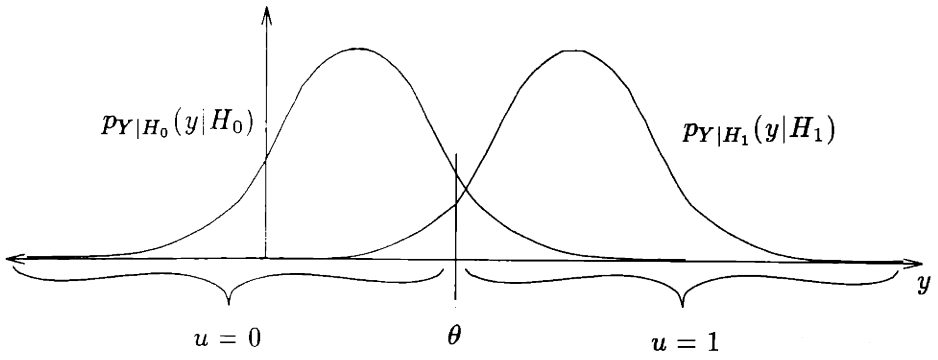


Figure 2-2: Gaussian Binary Detection Problem

$\mu_0 \neq \mu_1$. The observation threshold θ partitions the observation axis into two disjoint connected regions as depicted in Figure 2-2. For all realizations of Y lying above the observation threshold θ, the decision $U = 1$ is made, and for all decisions lying below θ, decision $U = 0$ is made.

For this class of binary hypothesis testing problem, we see that the optimal decision rule is *linear* in the data, and in fact is the simplest such function possible. The only data processing required is a direct comparison of the observation with a fixed threshold. We will refer to this class of decision rules as *linear threshold rules* and denote the class of all linear threshold rules as $\mathcal{T}$. This class of decision rules is also parameterized by a single scalar parameter, with the difference that the statistics are implicit in θ. Thus, for the Gaussian detection problem there is no loss of optimality in restricting the search for the optimal decision rule to the set $\mathcal{T}$, so that

$$\gamma^* = \arg\min_{\gamma \in \mathcal{G}} J_B(\gamma) = \arg\min_{\gamma \in \mathcal{T}} J_B(\gamma) \tag{2.23}$$

where the problem is equivalent to determining the optimal partition of the observation axis.

It should be emphasized that all binary hypothesis testing problems are not reducible to this form, even those involving Gaussian conditional density functions. Consider the problem of deciding between two possible mean-square values for a

Gaussian distribution [76]. For this problem, the observation is of the form

$$Y = \begin{cases} W_0 & \text{if } H = H_0 \\ W_1 & \text{if } H = H_1 \end{cases} \tag{2.24}$$

where $W_0 \sim N(0, \sigma_0^2)$, $W_1 \sim N(0, \sigma_1^2)$, and $\sigma_1^2 > \sigma_0^2 > 0$. In this case the LRT (2.16) is of the form

$$\frac{\frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma_1^2}}}{\frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma_0^2}}} \underset{u=0}{\overset{u=1}{\gtrless}} \eta \tag{2.25}$$

which is reducible to the test

$$y^2 \underset{u=0}{\overset{u=1}{\gtrless}} \frac{2\sigma_0^2\sigma_1^2}{(\sigma_1^2 - \sigma_0^2)} \ln(\eta \frac{\sigma_1}{\sigma_0}) \tag{2.26}$$

This reduced optimal test requires quadratic data processing.

Even for cases where a threshold rule makes better geometric sense, i.e., when the hypotheses are on the means of the conditional densities as in the Gaussian detection problem, the analytic form of the conditional densities usually makes it impossible to reduce the test to a linear one. Examples of alternative densities are Rayleigh, Erlang, Maxwell, or Cauchy, all of which are discussed in [45]. Thus, it is important to keep in mind when discussing linear threshold rules in this report, that obtaining the minimum probability of error linear threshold rule is not the same as determining the best possible minimum error decision rule.

2.2.2 Performance

Two extremely important conditional probabilities may be defined in terms of the likelihood ratio. If we let $p_{\Lambda|H_i}(\lambda|H_i)$ denote the probability density of the likelihood ratio Λ when hypothesis H_i is true, then we may define:

Definition 2.3 (Probability of False Alarm)

$$P_F(\eta) \triangleq \int_\eta^\infty p_{\Lambda|H_0}(\lambda|H_0)\, d\lambda = Pr(U = 1|H_0) \tag{2.27}$$

Definition 2.4 (Probability of Detection)

$$P_D(\eta) \triangleq \int_{\eta}^{\infty} p_{\Lambda|H_1}(\lambda|H_1)\, d\lambda = Pr(U = 1|H_1) \tag{2.28}$$

Typically the dependence of these probabilities on the LRT threshold η is not shown.

In terms of these probabilities, we may also define the following useful complementary conditional probability.

Definition 2.5 (Probability of Miss)

$$P_M \triangleq Pr(U = 0|H_1) = 1 - P_D \tag{2.29}$$

Then the Bayes Risk of a decision rule γ, may be expressed as

$$\begin{aligned} J_B(\gamma) &= \lambda_0 \Pr(\gamma(Y) = 1, H_0) + \lambda_1 \Pr(\gamma(Y) = 0, H_1) \\ &= \lambda_0 p_0 \Pr(\gamma(Y) = 1|H_0) + \lambda_1 p_1 \Pr(\gamma(Y) = 0|H_1) \\ &= \lambda_0 p_0 P_F + \lambda_1 p_1 P_M \end{aligned} \tag{2.30}$$

where λ_0 and λ_1 represent the costs of a false alarm and a miss, respectively. The probability of error is therefore simply

$$P_\epsilon = p_0 P_F + p_1 P_M \tag{2.31}$$

For a linear threshold rule of the form (2.21), we can express the probabilities of false alarm (2.27) and detection (2.28) as

$$P_F(\theta) = \int_{\theta}^{\infty} p_{Y|H_0}(y|H_0)\, dy \tag{2.32}$$

$$P_D(\theta) = \int_{\theta}^{\infty} p_{Y|H_1}(y|H_1)\, dy \tag{2.33}$$

so that

$$J_B(\theta) = \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{2.34}$$

For the Gaussian detection problem, the probabilities of false alarm and detection are expressible in terms of the error function

$$\Phi_\theta(k) = \int_{-\infty}^{\frac{\theta-\mu_k}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\,dx \quad k = 0, 1 \tag{2.35}$$

as

$$P_F(\theta) = \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy = 1 - \Phi_\theta(0) \tag{2.36}$$

$$P_D(\theta) = \int_\theta^\infty p_{Y|H_1}(y|H_1)\,dy = 1 - \Phi_\theta(1) \tag{2.37}$$

Then the Bayes Risk is given by

$$J_B = \lambda_0 p_0 (1 - \Phi_\theta(0)) + \lambda_1 p_1 \Phi_\theta(1) \tag{2.38}$$

ROC Curve

A particularly useful way of characterizing the quality of a DM is through a so-called receiver-operating characteristic (ROC) curve. A sample ROC is shown in Figure 2-3. This curve is a parametric plot of $P_D(\eta)$ vs. $P_F(\eta)$ as the LRT threshold η is varied from 0 to $+\infty$. It may be conveniently defined as:

$$ROC \equiv \{(P_F, P_D); P_F = P_F(\eta), P_D = P_D(\eta), 0 \leq \eta < \infty\} \tag{2.39}$$

For a given value of η, the point $(P_F(\eta), P_D(\eta))$ is referred to as the *operating point* of the DM. The ROC indicates the locus of achievable operating points of the DM, as dictated by the DMs conditional densities. For the binary hypothesis testing problem, the ROC can be likened to a "sufficient statistic" for the DM's observation space. That is, the ROC conveys the same information as the likelihood ratio in (2.14). It captures all the information about the conditional densities which is necessary to

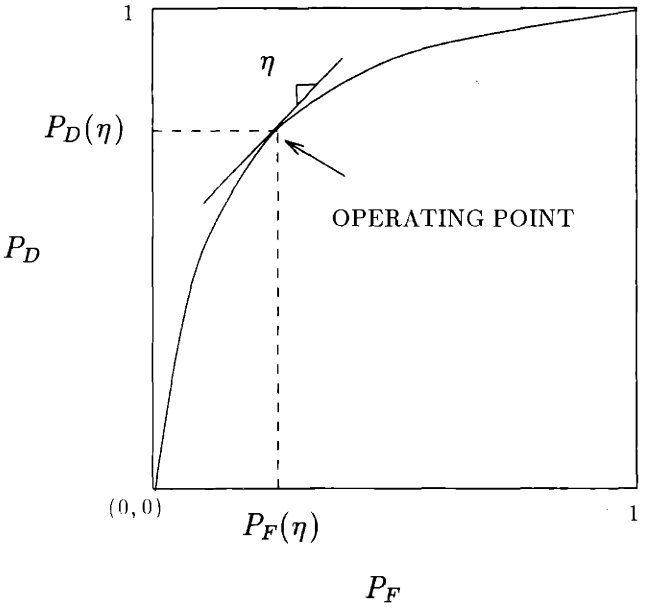


Figure 2-3: ROC curve

make a decision.

The ROC has several important properties of relevance in this report [71],[76]. It is continuous if the underlying conditional densities are continuous, and $P_D \geq P_F$, $\forall \eta$. It is also concave, i.e., given $0 < \alpha < 1$, and two operating points (P_F^1, P_D^1) and (P_F^2, P_D^2), and denoting the ROC function by $g : P_F \rightarrow P_D$, it holds that

$$g(\alpha P_F^1 + (1-\alpha)P_F^2) \geq \alpha g(P_F^1) + (1-\alpha)g(P_F^2) \tag{2.40}$$

so that the chord connecting two points (values of P_D) always lies below the ROC itself. Another important property is that a tangent line to the ROC at the operating point has slope equal to η, the threshold of the LRT, as indicated in Figure 2-3. In particular

$$\frac{dP_D}{dP_F}(\eta) = \eta \tag{2.41}$$

The primary usefulness of the ROC in the decision making context is that it provides a concise characterization of a DM's expertise. For example, if one ROC lies strictly above another, in the sense that for every value of P_F it achieves a higher value of P_D, that DM can be considered superior.

2.3 The Centralized Problem

We now consider the setting in which a collection of scalar-valued observations Y_1,Y_2, ..., Y_M. are obtained of the same environment, and are available for use in a single decision rule γ. Such a case would occur if a single DM made repeated observations of the same environment, or if each of a collection of M DMs received an observation and then all the observations were communicated to one DM to make the decision.

The Bayes Risk in this setting is given by

$$\begin{aligned} J_{Bayes}(\gamma) &= E_{U,H}\{C(U,H)\} \\ &= E_{Y_1,\ldots,Y_M,H}\{C(\gamma(Y_1,\ldots,Y_M),H)\} \\ &= \sum_{j=0}^{j=1} p_j E_{Y_1,\ldots,Y_M}\{C(\gamma(Y_1,\ldots,Y_M),H)|H=H_j\} \\ &= \sum_{i=0}^{1}\sum_{j=0}^{1} C(U=i,H=H_j)p_j \Pr(\gamma(Y_1,\ldots,Y_M)=i|H_j) \\ &= C(0,H_0)+C(1,H_1) \\ &\quad +[C(1,H_0)-C(0,H_0)]p_0\Pr(\gamma(Y_1,\ldots,Y_M)=1|H=H_0) \\ &\quad +[C(0,H_1)-C(1,H_1)]p_1\Pr(\gamma(Y_1,\ldots,Y_M)=0|H=H_1) \end{aligned} \quad (2.42)$$

which we again simplify to

$$J_B(\gamma) = \lambda_0 p_0 \Pr(\gamma(Y_1,\ldots,Y_M)=1|H=H_0) + \lambda_1 p_1 \Pr(\gamma(Y_1,\ldots,Y_M)=0|H=H_1) \quad (2.43)$$

where λ_0 and λ_1 are positive costs.

It is well-known ([71]) that the optimal decision rule is still an LRT, with the joint statistics forming the likelihood ratio. In particular, the optimal rule is given by

$$\Lambda(y) \underset{u=0}{\overset{u=1}{\gtrless}} \eta \quad (2.44)$$

where now

$$\Lambda(y) \triangleq \frac{p_{Y_1,\ldots,Y_M|H_1}(y_1,\ldots,y_M|H_1)}{p_{Y_1,\ldots,Y_M|H_0}(y_1,\ldots,y_M|H_0)} \quad (2.45)$$

and η is still as in (2.15).

If the observations are conditionally independent, given each hypothesis, then the joint densities in (2.45) factor into the marginal densities, and the likelihood ratio becomes

$$\Lambda(y) \triangleq \frac{p_{Y_1|H_1}(y_1|H_1)\cdots p_{Y_M|H_1}(y_M|H_1)}{p_{Y_1|H_0}(y_1|H_0)\cdots p_{Y_M|H_0}(y_M|H_0)} \tag{2.46}$$

For the Gaussian Detection Problem in this setting, the ith observation is of the form

$$Y_i = \begin{cases} \mu_0 + W_i & \text{if } H = H_0 \\ \mu_1 + W_i & \text{if } H = H_1 \end{cases} \tag{2.47}$$

with $W_i \sim N(0, \sigma_i^2)$ and the $\{W_i\}$ statistically independent. Then each observation Y_i is conditionally independent from the set $\{Y_j | j \neq i\}$. Again, application of $\ln(\cdot)$ to both sides of (2.44) reduces the test to the equivalent decision rule

$$\sum_{i=1}^{M} \frac{y_i}{\sigma_i^2} \underset{u=0}{\overset{u=1}{\gtrless}} \theta \tag{2.48}$$

where the centralized observation threshold θ is given by the closed-form expression

$$\begin{aligned} \theta &= \frac{1}{\mu_1 - \mu_0} \ln \eta + \frac{\mu_1 + \mu_0}{2} \sum_{i=1}^{M} \frac{1}{\sigma_i^2} \\ &= \frac{1}{\mu_1 - \mu_0} \ln \left(\frac{\lambda_0 p_0}{\lambda_1 p_1} \right) + \frac{\mu_1 + \mu_0}{2} \sum_{i=1}^{M} \frac{1}{\sigma_i}^2 \end{aligned} \tag{2.49}$$

Note that this decision rule requires joint processing of the observations. The rule (2.48) can be seen to form a hyperplane in M dimensional observation space. For the case $M = 2$, the decision rule is a line in $y_1 - y_2$ space as shown in Figure 2-4.

2.4 The General Decentralized (Team) Problem

The essential components of the decentralized problem are much the same as for the centralized problem. Each of a collection of DMs receives an observation, but now this observation is available only locally because communication between the DMs is restricted. The observations cannot be centralized, so that the limited communication

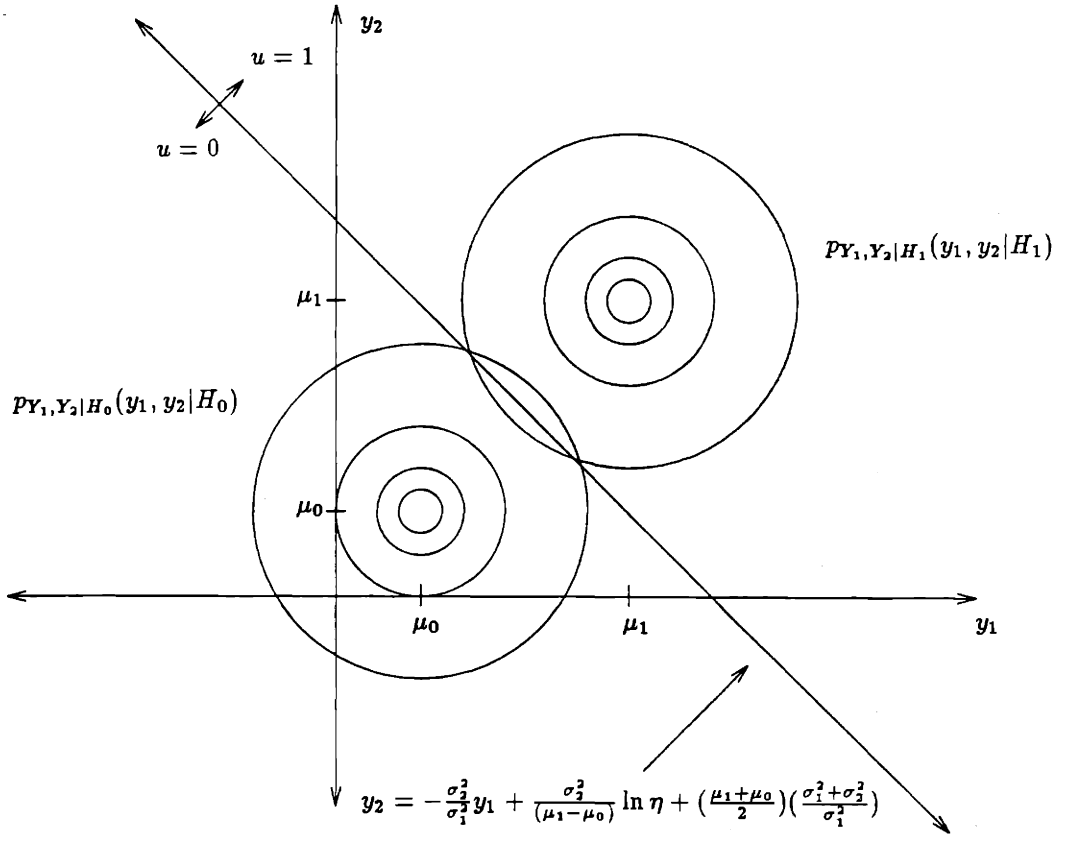


Figure 2-4: Centralized Decision Rule, $M = 2$

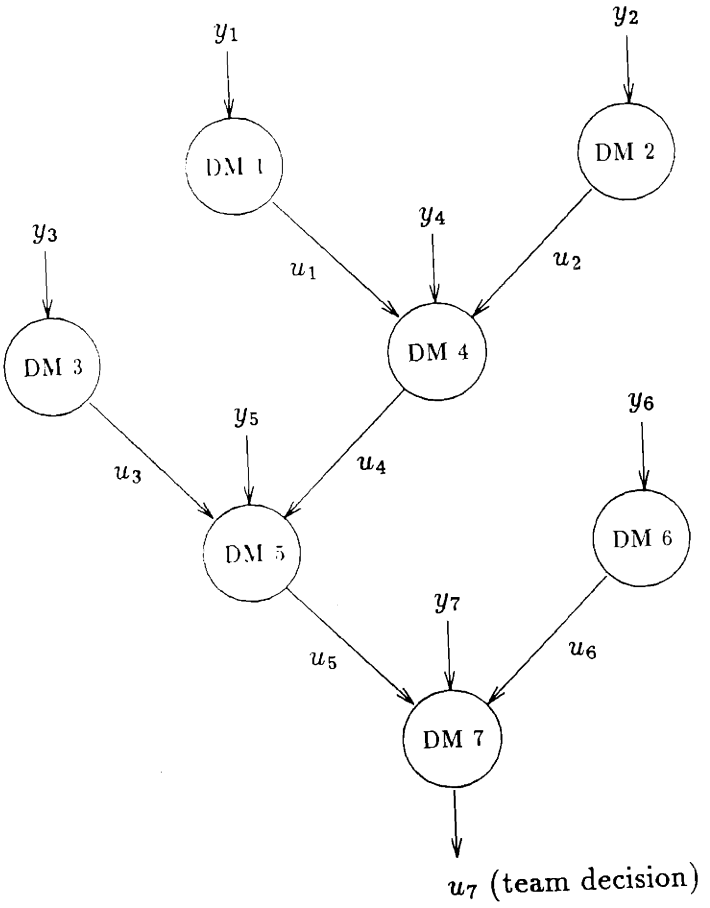


Figure 2-5: Typical DBHT Network

available must be used by each DM to convey as much information about its observation as possible. Information in this sense is measured with respect to optimizing overall team performance. One can think of this operation as an optimal quantization problem, where each DM endeavors to provide the "best" possible "measurement", based on local processing of its own observation and any messages it has received, to those DMs which receive its message. The necessary mathematical formalism needed to make this discussion precise may be established as follows.

Consider a collection of M DMs such as depicted in Figure 2-5 for a case with $M = 7$. Throughout this report we will indicate particular topologies using this graphical representation, where each node in the graph represents a DM, and allowed communication pathways between DMs are indicated graphically by directed arcs. A single decision cycle begins with every DM i, $i = 1, ..., M$, receiving a noisy observa-

tion Y_i of the environment, where Y_i takes values in a set $\mathcal{Y}_i$. Assumed known to each DM i, $i = 1, \ldots, M$ are the topology of the network, the prior probabilities p_0, p_1, the costs λ_0, λ_1, and the conditional probability density functions $p_{Y_i|H_j}(y_i|H_j)$, $j = 0, 1$, which completely capture the statistical relationship between each DM's observation and the underlying hypotheses[7]. DM i makes decision $u_i \in \mathcal{M}_i$, where $\mathcal{M}_i$ is a finite message set containing allowable messages. Each DM's messages are then made available to the other DMs according to the prespecified communication protocol indicated in the graph. The function which maps DM i's inputs, i.e., its observation and any decisions it receives from upstream DMs, to its output decision is still known as a *decision rule*, and will be denoted γ_i. Note that if DM i receives k messages from upstream DMs, γ_i is a function such that $\gamma_i : \mathcal{Y}_i \times \prod_{j=1}^{k} \mathcal{M}_j \mapsto \mathcal{M}_i$. We will refer to the collection of decision rules for the entire network $\underline{\gamma} = (\gamma_1, \ldots, \gamma_M)$ as a *strategy*.

Messages are chosen from the sets $\mathcal{M}_i$, $i = 1, \ldots, M$ in order to optimize some measure of *organizational* performance. For simplicity, in this report we restrict to the case of *binary-valued* message sets, i.e., $\mathcal{M}_i = \{0, 1\}$, $i = 1, \ldots, M$, so that local decisions $u_i \in \{0, 1\}$ are binary-valued as well. One caveat with this notational scheme - by this choice of message labels we do *not* intend to indicate that the message chosen by a DM corresponds to a local decision on the true hypothesis. Rather, the message should be interpreted as a signal, which is chosen in order to optimize team performance. The performance measure we adopt is the team Bayes cost, which in the most general case is defined to be

$$J_{Bayes}(U_1, \ldots, U_M, H) = \sum_{i=1}^{M} \sum_{j=0}^{1} \sum_{k=0}^{1} C_i(k, H_j) \Pr(U_i = k, H = H_j) \tag{2.50}$$

where $C_i(k, H_j)$ is the cost of DM i choosing message k when hypothesis H_j is true, and where the cost assigned depends directly on the decisions of all DMs in the network. Frequently, however, a so-called *primary* DM is specified to make the overall team decision. If we assume DM M to be the primary DM, then this version of the

[7]It should be noted that, in general, the conditional densities may be different at different sensors, meaning that the sensors may have differently distributed noise (varying degrees of "expertise") for a given team decision problem.

team problem is formulated with cost

$$J_{Bayes}(U_1,\ldots,U_M,H) = \sum_{j=0}^{1}\sum_{k=0}^{1} C_M(k,H_j)\Pr(U_M = k, H = H_j) \tag{2.51}$$

which for the simplified version of the Bayes cost we express as

$$\begin{aligned} J_B(U_1,\ldots,U_M,H) &= \lambda_0\Pr(U_M = 1, H = H_0) + \lambda_1\Pr(U_M = 0, H = H_1) \\ &= \lambda_0 p_0 \Pr(U_M = 1 | H = H_0) \\ &\quad +\lambda_1 p_1 \Pr(U_M = 0 | H = H_1) \end{aligned} \tag{2.52}$$

Although the cost depends explicitly only on the decision of the primary DM, the decision rules of all of the other DMs which are intermediate to the process affect the cost implicitly. This dependence can be made apparent by expanding out the dependence of U_M on the other team decisions in (2.52), but this must be done on a case by case basis since the exact functional form of the cost depends on the particular topology being considered.

The primary DM is typically taken to be the last DM in the information pathway[8]. For example, in the tree-type topology depicted in Figure 2-5, the primary DM would correspond to the root node, which in this case is DM 7. We should point out that many formulations in the decentralized detection literature replace the primary DM with a "fusion center", which only receives decision input from the other DMs and receives no observation of its own. We avoid such formulations in this report in the interest of maintaining homogeneity of the nodes in the network which is desirable from our modeling perspective. Thus, in our formulation every DM receives an observation, and the primary DM can be interpreted as a type of "generalized fusion center" which receives an observation of its own.

We are now prepared to pose the decentralized Bayesian hypothesis testing problem, which involves choice of the team strategy in order to optimize the organizational

[8]Strictly speaking this is not necessary, but there is no point in including DMs in the network which have no measurable impact on the outcome.

Bayes cost.

Problem 2.2 Minimum Bayes Risk Decentralized Binary Hypothesis Testing

Given a network topology Team comprising M DMs, the joint conditional density functions $p_{Y_1,\ldots,Y_M|H_i}(y_1,\ldots,y_M|H_i), i=0,1$, the positive prior probabilities p_0, p_1, and the positive (bounded) costs λ_0, λ_1, determine a decision strategy $\underline{\gamma}^* = (\gamma_1^*,\ldots,\gamma_M^*)$ which satisfies

$$\underline{\gamma}^* = \arg\min_{\underline{\gamma}\in\mathcal{G}} J_B(\underline{\gamma}) \tag{2.53}$$

where $\mathcal{G}$ denotes the set of all possible decision strategies and $J_B(\underline{\gamma})$ is the Bayes risk incurred by the team under strategy $\underline{\gamma}$.

Again, the statement of this problem implies optimization over a large set. However, by making several technical assumptions, as described in the following section, we can again force optimal strategies in the team problem to lie in a very structured and tractable set.

2.4.1 Restrictions

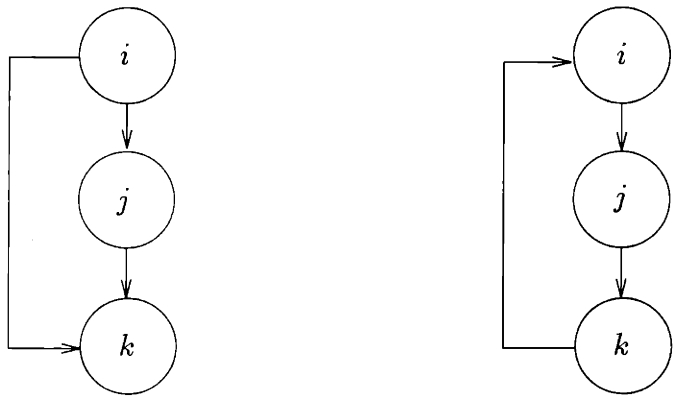
We now make several assumptions which restrict the classes of DBHT models with which we will be concerned in this report.

For the following, assume we have a network comprised of M DMs.

Assumption 2.1 (Network Topology)

Networks of DMs are arranged in trees, that is the networks are representable as singly-connected directed acyclic graphs with nodes connected by directed arcs.

The assumption of tree structure indicates that acceptable network structures are of the form indicated previously in Figures 1-1 and 2-5, where the directed arcs represent unidirectional communication links. What is specifically not permitted are



(a) (b)

Figure 2-6: Network Structures which are Not Permitted. (a) feedforward; not singly connected, (b) feedback; contains cycle

the feedforward and feedback structures of Figure 2-6. In the case of the feedforward structure, the graph is not singly connected, meaning that there exist two different paths connecting nodes i and k, path $i \rightarrow j \rightarrow k$ and path $i \rightarrow k$. For the feedback structure, a cycle $i \rightarrow j \rightarrow k \rightarrow i$ is created. The mathematical difficulties which arise in conjunction with these topological variations will be addressed momentarily in Section 2.4.2.

Assumption 2.2 (Observations)

(a) *The observations* $Y_1, \ldots, Y_M$ *are conditionally independent given either hypothesis, i.e.,*

$$p_{Y_1,\ldots,Y_M|H_j}(y_1, \ldots, y_M|H_j) = p_{Y_1|H_j}(y_1|H_j) \cdots p_{Y_M|H_j}(y_M|H_j), \quad j = 0, 1 \qquad (2.54)$$

(b) Y_i *is a continuous real-valued random variable, i.e.,* $\mathcal{Y}_i = \Re, \forall i = 1, \ldots, M$.

Assumption 2.2 (a) is the most critical and restrictive of our assumptions, and without it it is difficult to make progress. In particular, without the assumption of conditional independence, the problem has been shown by Tsitsiklis and Athans [68]

to be NP-complete. Its other critical implications are discussed in Section 2.4.2.

Assumption 2.2 (b) states that the observations at each DM are real-valued scalar random variables. We do not consider discrete observation spaces because of complications in the decision rules which often arise in discrete problems. For example, it is well known [71], [76], [65], [43], that the optimal decision rules for hypothesis testing problems with discrete observation spaces often involve some form of randomization. The choice of scalar observations enables the decision spaces to be more easily visualized (Section 3.2.1), results in simpler simulations, and does not result in any loss of generality with respect to the goals of this report.

Assumption 2.3 (Message Set)

The message set of each DM is binary-valued, i.e.,

$$\mathcal{M}_i = \{0, 1\}, \forall\, i = 1, \ldots, M \tag{2.55}$$

Thus, we restrict each DM to use the simplest nontrivial message set possible. Each DM is allowed only one bit of communication capacity. Allowing more messages clearly improves the performance of the team since, in the limit, each DM is able to transmit its entire observation [52]. In this case, the centralized solution, which provides a lower bound on the performance of the team, would be attainable.

Assumption 2.4 (Cost Function)

The cost is taken to be the probability of error of the primary DM (root node of tree), so that if DM M is the primary DM, the team cost is given by

$$P_e^{Team} = p_0 P_F^M + p_1 P_M^M \tag{2.56}$$

Rather than consider the general Bayes criterion, we restrict to the minimum probability of error performance metric for all the team hypothesis testing problems in this report. We do not do this so much for simplicity as for the fact that it is the natural performance metric for the modeling effort discussed in Chapter 1, as well as for many problems in decision making and pattern classification. This restriction in no way limits the applicability of our results, all of which are easily adapted to handle unequal costs on the team errors.

2.4.2 Significance of these Restrictions

The significance of Assumptions 2.1 and 2.2(a) to the results of this report cannot be overstated. These assumptions combine to give Problem 2.2 sufficient structure so that the problem of determining the optimal team decision rules becomes tractable. First, we must quantify what we mean by optimal. Necessary and sufficient conditions for optimality of strategies in team hypothesis testing problems do not exist. Rather, one has to settle for determining strategies that satisfy only the necessary conditions, and these are not expressible in closed-form. They are typically specified by coupled systems of equations referred to as person-by-person optimality conditions, since they specify the necessary conditions for optimality for each decision rule, given that the other decision rules in the network are held fixed. A person-by-person optimal strategy, or a strategy in which each component decision rule is person-by-person optimal, is a strategy whose performance cannot be improved by perturbing any single decision rule. Thus, a person-by-person optimal strategy is not even guaranteed to be locally optimal, since it is not clear that the strategy cannot be improved by perturbing several decision rules simultaneously. It is also clear that all globally optimal strategies are necessarily person-by-person optimal, but the converse is not true.

It has been shown by Tsitsiklis [65] that conditionally independent observations and tree-type topologies ensure that the person-by-person optimality conditions for decentralized binary hypothesis testing networks are expressible in the form of coupled likelihood ratio tests. In particular, the optimal decision rule of a DM i receiving messages from k upstream DMs indexed 1-k, is expressible in the general form

For DM i, given fixed $\gamma_j, j = 1, \ldots, M, j \neq i$:

$$\Lambda_i(y_i) \triangleq \frac{p_{Y_i|H_1}(y_i|H_1)}{p_{Y_i|H_0}(y_i|H_0)} \underset{u_i=0}{\overset{u_i=1}{\gtrless}} \begin{cases} \eta_1^i & \text{if } u_j = 0,\ \forall j,\ j = 1, \ldots, k \\ \eta_2^i & \text{if } u_1 = 1,\ u_j = 0,\ \forall j,\ j = 2, \ldots, k \\ \vdots & \vdots \\ \eta_{2^k}^i & \text{if } u_j = 1,\ \forall j,\ j = 1, \ldots, k \end{cases} \tag{2.57}$$

where the $\eta_1^i - \eta_{2^k}^i$ are a set of 2^k possible LRT thresholds, one of which is selected by the particular combination of messages received from upstream DMs. As discussed in the next section, the numerical values of the LRT thresholds normally depend on the current decision rules being employed by the other network DMs, resulting in a coupled set of necessary conditions for optimality.

Decision rules of this form result from the fact that, under Assumptions 2.1 and 2.2(a), the local observation Y_i of a given DM i is guaranteed to be statistically independent from all other incoming information received from upstream DMs. This allows each local decision rule to be structured as an LRT over the local information set. This property may fail to hold if the observations are statistically dependent, since this introduces correlation between the local observation and incoming messages. In the absence of this restriction, the decision rules can become very messy, no longer necessarily being threshold rules. Furthermore, it has been shown by Tsitsiklis and Athans [68] that the problem of determining the optimal strategy in a decentralized hypothesis testing problem without the conditional independence assumption is an inherently intractable combinatorial problem. In the same vein, if we do not restrict to tree-type topologies, similar problems may arise, since dependence is then introduced through the communication scheme.

In summary, Assumptions 2.1 and 2.2(a) combine to guarantee that the search for optimal team strategies may be restricted to rules of the form (2.57), i.e., coupled LRTs, with no loss of optimality. This places optimal network strategies in a very

structured set, and suggests a parameterization of the decision rules which describes the optimum for the Gaussian detection problem and is suitable for optimization as discussed in Chapter 3.

2.5 Examples of Small Teams

The best way to clarify the previous description of the restricted DBHT model is by presenting examples. In this section, we present four examples of small teams that are fairly representative of the major topological variations that can arise in tree-structured networks. We illustrate the form of the optimal decision rules (LRTs) for each network. We then demonstrate that for the Gaussian detection problem the optimal decision rules admit parameterization by linear threshold tests, where the values of the optimal thresholds are expressed as a system of coupled nonlinear equations. We hope that illustrating the form of the decision rules which arise for these small teams will enhance the intuition behind the properties we later claim to hold true for general tree-structures with conditionally independent observations. We also feel that the inherent complexity of the DBHT problem is made evident by writing out the optimal decision rules for each of these rather simple-looking examples.

2.5.1 Example 1: Two-Member Tandem (2-Tand)

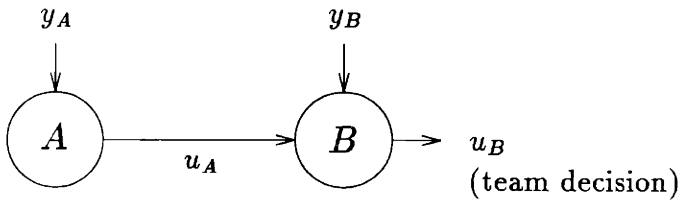


Figure 2-7: 2-Tand

A topology of particular importance to us is illustrated in Figure 2-7. This type of structure is referred to as *series* or *tandem*, and the two member tandem team will

hereafter be denoted 2-Tand. 2-Tand is of special interest because it is the smallest team in which all of the fascinating properties of decentralized detection problems arise.[9] The operation of 2-Tand may be described as follows: DM A receives an observation y_A which it uses to choose a message $u_A \in \{0,1\}$ to send to DM B. DM B takes into account the message u_A from DM A as well as its own observation y_B to compute the overall team decision $u_B \in \{0,1\}$. Thus, DM B is the primary DM for this team. The decision rules employed by DM A and DM B are of the form $\gamma_A : \mathcal{Y}_A \mapsto u_A \in \{0,1\}$ and $\gamma_B : \mathcal{Y}_B \times \{0,1\} \mapsto u_B \in \{0,1\}$, respectively. It remains to determine just what γ_A and γ_B should be in view of the minimum probability of error criterion.

The *necessary* conditions for optimality of the decision rules γ_A and γ_B, under Assumptions 2.2(a), 2.3, and 2.4 were derived in [19]. Perhaps surprisingly, as in the centralized case, they are again likelihood ratio tests (LRTs).

For DM B, given γ_A:

$$\Lambda_B(y_B) \triangleq \frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \dfrac{p_0 \Pr(U_A=0|H_0)}{p_1 \Pr(U_A=0|H_1)} & \text{if } u_A = 0 \\ \dfrac{p_0 \Pr(U_A=1|H_0)}{p_1 \Pr(U_A=1|H_1)} & \text{if } u_A = 1 \end{cases} \tag{2.58}$$

For DM A, given γ_B:

$$\Lambda_A(y_A) \triangleq \frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \frac{p_0[\Pr(U_B=1|U_A=1,H_0) - \Pr(U_B=1|U_A=0,H_0)]}{p_1[\Pr(U_B=1|U_A=1,H_1) - \Pr(U_B=1|U_A=0,H_1)]} \tag{2.59}$$

In the derivation of these rules, it is normally assumed that

$$\begin{aligned} \Pr(U_B=1|U_A=1,H_0) &\geq \Pr(U_B=1|U_A=0,H_0) \\ \Pr(U_B=1|U_A=1,H_1) &\geq \Pr(U_B=1|U_A=0,H_1) \end{aligned} \tag{2.60}$$

[9]The two-member *parallel* topology involves a "fusion center" for combining the two decisions which we wish to avoid.

so that

$$0 \leq \frac{p_0[\Pr(U_B = 1|U_A = 1, H_0) - \Pr(U_B = 1|U_A = 0, H_0)]}{p_1[\Pr(U_B = 1|U_A = 1, H_1) - \Pr(U_B = 1|U_A = 0, H_1)]} \tag{2.61}$$

If this is not the case, the opposite assignment of message U_A may be made.

As discussed previously, these conditions are also referred to as the *person-by-person* optimality conditions since they give the conditions for optimality of each DM's decision rule, given that the decision rule of the other DM is held fixed. The rules specify the optimal action of each DM as a function of the other DM's rule. The rules are coupled, and cannot be expressed in closed-form.

The form of the rules is somewhat intuitive. The decision rule of the downstream DM B is simply the centralized rule over the new "measurement" set $\{U_A, Y_B\}$. This is made obvious by flipping the probability mass ratio $\frac{\Pr(U_A=i|H_0)}{\Pr(U_A=i|H_1)}, i = 0, 1$ to the other side. The threshold used by DM A can be interpreted as the cost to be incurred downstream by the primary DM B as a result of its decision [19].

Notice that the impact of each DM on the other appears as bias in the LRT thresholds. This bias enters in the same fashion that unequal costs on each type of error enter the general Bayes problem. However, these costs depend directly on the operating point of the other DM.

It should be emphasized that U_A is not to be interpreted as a best local decision by DM A as to the true acting hypothesis. Rather, the message sent by A is a *signal* to DM B, with which A attempts to provide B with the best possible information with which to make a decision.

We may rewrite these LRTs in more compact form by introducing the following notation. Let P_F^A, P_D^A denote the probabilities of false alarm and detection of A and P_F^{Bi}, P_D^{Bi} the probabilities of false alarm and detection of B when A transmits message $i = 0, 1$. Then if we define $\eta \triangleq \frac{p_0}{p_1}$, these LRTs can be expressed in terms of the network operating points as:

For DM B, given fixed γ_A:

$$\frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \eta\frac{1-P_F^A}{1-P_D^A} = \eta_0 & \text{if } u_A = 0 \\ \eta\frac{P_F^A}{P_D^A} = \eta_1 & \text{if } u_A = 1 \end{cases} \tag{2.62}$$

For DM A, given fixed γ_B:

$$\frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \eta\frac{P_F^{B1} - P_F^{B0}}{P_D^{B1} - P_D^{B0}} = \eta_A \tag{2.63}$$

Each DM's LRT threshold(s) correspond to so-called *operating points* on its receiver operating characteristic (ROC) curve as illustrated in Figure 2-8. Since DM B employs two thresholds, it has two associated operating points, one of which is selected by the upstream decision u_A.

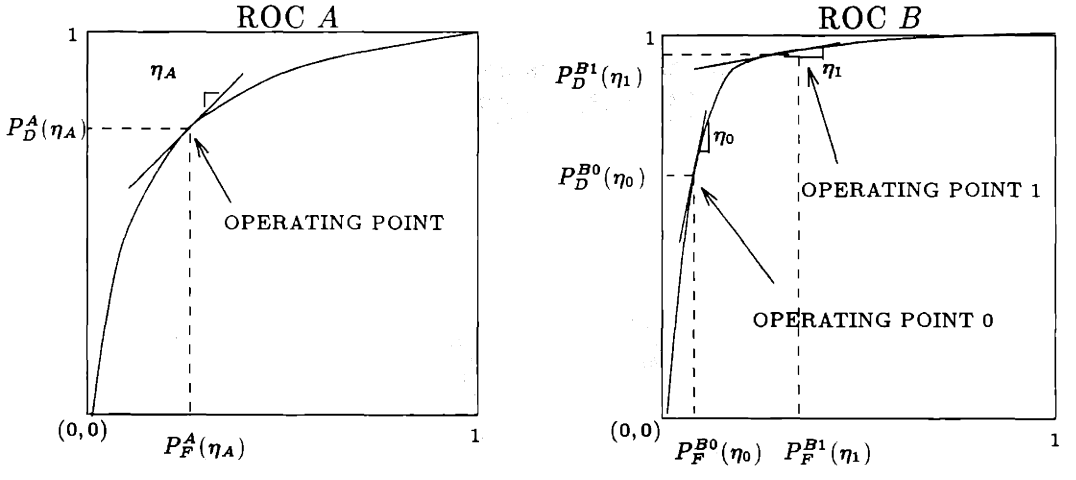


Figure 2-8: 2-Tand ROCs

Performance

The probability of error associated with this optimal solution may be parameterized

by the probabilities of false alarm and detection of A and B and expressed as

$$\begin{aligned} P_\epsilon^{2-Tand} &= p_0 \overbrace{[(1-P_F^A)P_F^{B0} + P_F^A P_F^{B1}]}^{P_F^{2-Tand}} \\ &\quad + p_1 \underbrace{[(1-P_D^A)(1-P_D^{B0}) + P_D^A(1-P_D^{B1})]}_{P_M^{2-Tand}} \end{aligned} \tag{2.64}$$

where by analogy to (2.31) we may define a team operating point by

$$\begin{aligned} P_F^{2-Tand} &= [(1-P_F^A)P_F^{B0} + P_F^A P_F^{B1}] \\ P_M^{2-Tand} &= [(1-P_D^A)(1-P_D^{B0}) + P_D^A(1-P_D^{B1})] \end{aligned} \tag{2.65}$$

This result was derived in [19], but we will see how it may be easily rederived in Section 3.2.2.

A significant reduction in the decision rules of equations (2.62) - (2.63) can be effected if we work with a particular class of decision problems, namely those in which the network's objective is to decide which of two constant signals occurred with each DM's measurement corrupted by zero-mean Gaussian noise (team Gaussian detection). For this case the observations at DMs A and B are of the form

$$y_A = \begin{cases} \mu_0 + W_A & : H_0 \\ \mu_1 + W_A & : H_1 \end{cases} \tag{2.66}$$

$$y_B = \begin{cases} \mu_0 + W_B & : H_0 \\ \mu_1 + W_B & : H_1 \end{cases} \tag{2.67}$$

where W_A and W_B are conditionally statistically independent and $W_A \sim N(0, \sigma_A^2)$, $W_B \sim N(0, \sigma_B^2)$.

Under the assumption that the ratios

$$\frac{1-P_F^A}{1-P_D^A}, \quad \frac{P_F^A}{P_D^A}, \quad \frac{P_F^{B1} - P_F^{B0}}{P_D^{B1} - P_D^{B0}} \tag{2.68}$$

are all positive, we can use the $\ln(\cdot)$ function to reduce (2.62) - (2.63) to simpler tests

as before. The only questionable term is the third one. However, it is easily verified that the numerator and denominator are either both positive or both negative so long as $P_F^{B0} \neq P_F^{B1}$ and $P_D^{B0} \neq P_D^{B1}$. Then application of $\ln(\cdot)$ yields the following set of equivalent tests, which are now *linear* in the observations.

For DM B, given fixed γ_A:

$$y_B \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \beta_0 & \text{if } u_A = 0 \\ \beta_1 & \text{if } u_A = 1 \end{cases} \tag{2.69}$$

For DM A, given fixed γ_B:

$$y_A \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \alpha \tag{2.70}$$

where the fixed observation axis thresholds α, β_0, β_1 must satisfy the system of coupled nonlinear equations

$$\begin{aligned} \beta_0 &= \tfrac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\tfrac{\Phi_\alpha(0)}{\Phi_\alpha(1)}\right) + \tfrac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\ \beta_1 &= \tfrac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\tfrac{1-\Phi_\alpha(0)}{1-\Phi_\alpha(1)}\right) + \tfrac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\ \alpha &= \tfrac{\sigma_A^2}{\mu_1-\mu_0} \ln\left(\tfrac{\Phi_{\beta_0}(0)-\Phi_{\beta_1}(0)}{\Phi_{\beta_0}(1)-\Phi_{\beta_1}(1)}\right) + \tfrac{\sigma_A^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \end{aligned} \tag{2.71}$$

which specify the necessary conditions for optimality, where the functions $\Phi_\alpha(k)$, $\Phi_{\beta_0}(k)$, and $\Phi_{\beta_1}(k)$ are given by

$$\Phi_\alpha(k) = \int_{-\infty}^{\frac{\alpha-\mu_k}{\sigma_A}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx \tag{2.72}$$

$$\Phi_{\beta_0}(k) = \int_{-\infty}^{\frac{\beta_0-\mu_k}{\sigma_B}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx \tag{2.73}$$

$$\Phi_{\beta_1}(k) = \int_{-\infty}^{\frac{\beta_1-\mu_k}{\sigma_B}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx \tag{2.74}$$

for $k = 0, 1$.

Comparison of the functional form of these equations with (2.22) indicates that each team threshold is specified by an equation of the same form as the single DM case, except for the additional cost through which the equations are coupled. There is no closed form solution to this system; it must be solved numerically using an iterative technique such as successive approximation.

DM A employs one observation axis threshold α, while DM B requires two (for the binary message case, a DM employs 2^k where k is the number of immediate predecessors). DM B will use threshold β_0 if it receives message $u_A = 0$ and threshold β_1 if it receives message $u_A = 1$. Figures (2-9)-(2-11) illustrate the relative positions of DM A's threshold α and the two thresholds β_0 and β_1 of DM B for some sample operating conditions. The $\times$'s denote the locations of the optimal thresholds for each DM in isolation. In Figure 2-9, both DMs have $\sigma_A^2 = \sigma_B^2 = 100$. With reference to the modeling of human decision makers, we say that DM A and DM B have equal decision making capability or are equally "smart". Part (a) illustrates the symmetry of the equal prior case, while (b) and (c) illustrate how the thresholds shift right or left in response to prior bias in the data. A particularly interesting effect is the development of a "lying region" along the observation axis, corresponding to those values of the observation for which DM A sends the opposite message to DM B than it would select in isolation. This portion of the axis is highlighted between the position of α and the $\times$ denoting the optimal isolated threshold position. It is clear evidence of the coupling that exists between DM A and DM B. In Figure 2-10, DM A is "smarter" than DM B, i.e., it has smaller noise variance. The optimal positions of DM B's thresholds have now spread apart to reflect the increased confidence in the decision of DM A. In the limit, these thresholds would spread to $+\infty$ and $-\infty$, indicating that DM B would choose to always agree with DM A. In Figure 2-11 the opposite effect may be observed. Now it may be seen that the thresholds of DM B have moved together, indicating reduced confidence on DM B's part, and a tendency to ignore what DM A says. In the limit, DM B would perform as it would in isolation. More discussion along these lines is presented in [52].

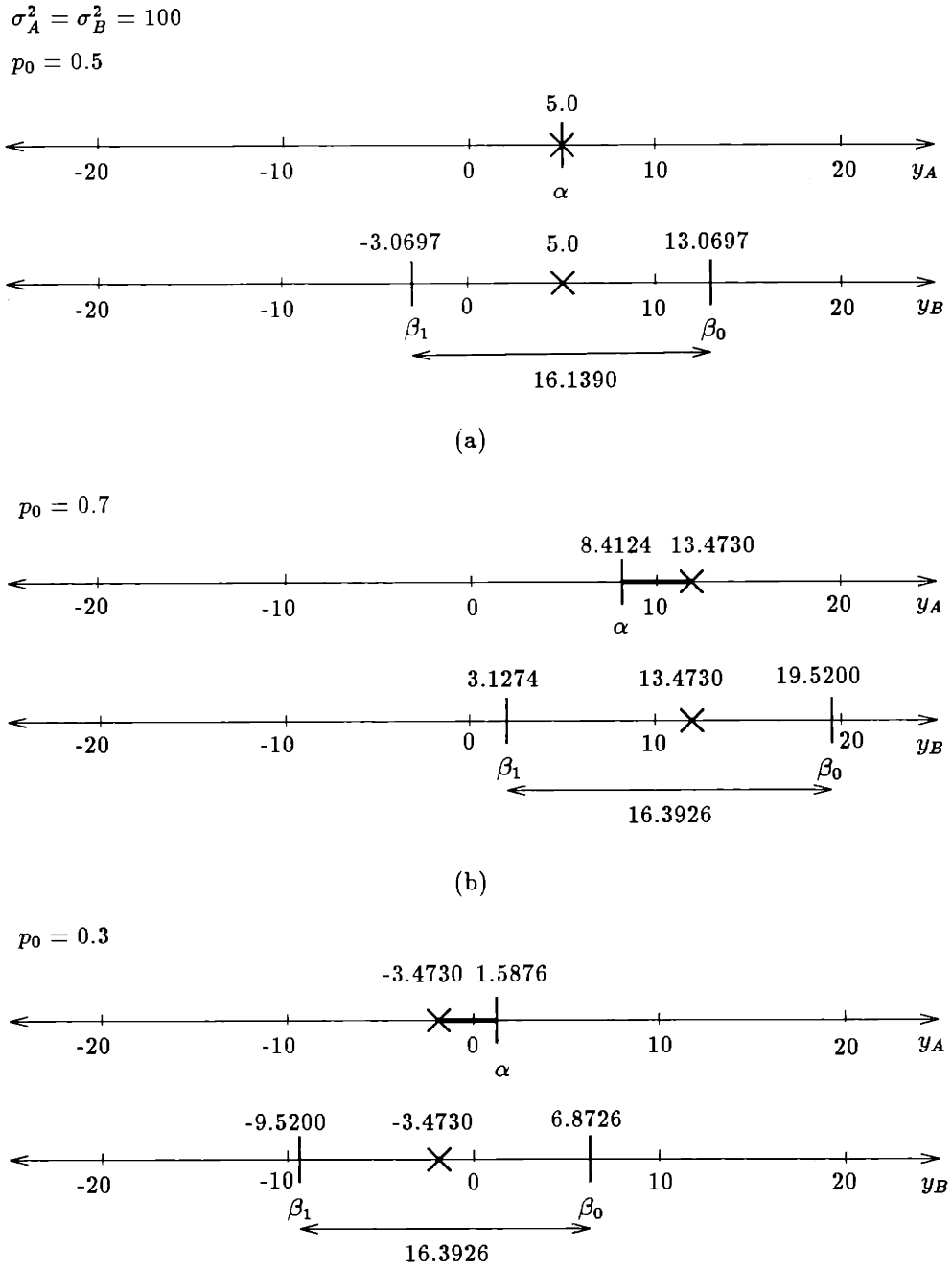
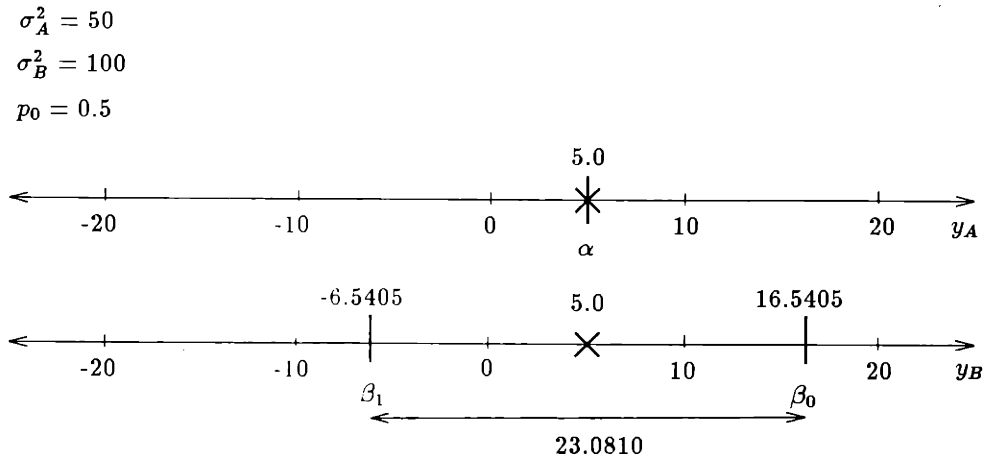
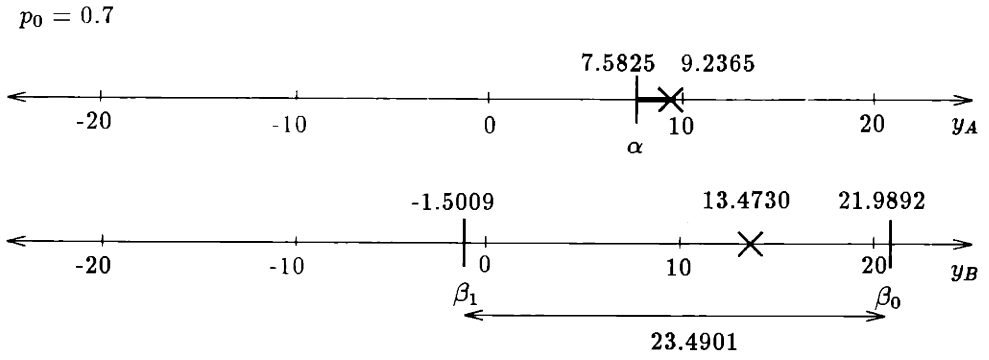


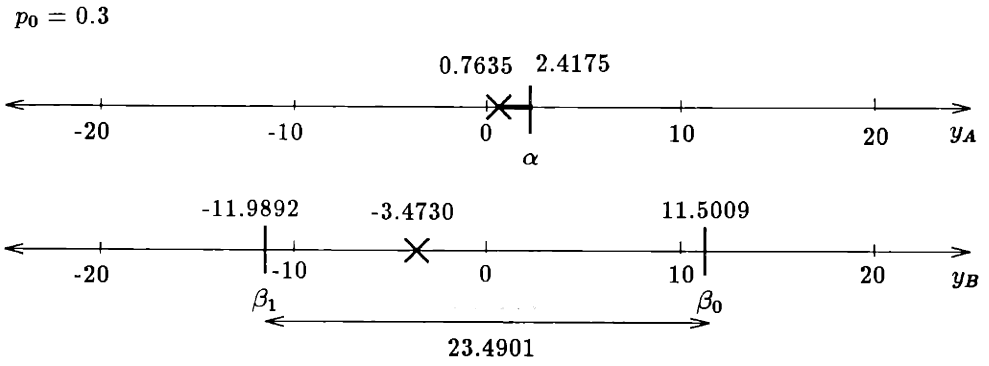
Figure 2-9: 2-Tand thresholds: Equally smart (equal variance) case, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = \sigma_B^2 = 100$. (a) $p_0 = p_1 = 0.5$, (b) $p_0 = 0.7$, $p_1 = 0.3$, (c) $p_0 = 0.3$, $p_1 = 0.7$



(a)



(b)



(c)

Figure 2-10: 2-Tand threshholds: DM A smarter ($\sigma_A^2 = 50$), DM B dumber ($\sigma_B^2 = 100$), $\mu_0 = 0$, $\mu_1 = 10$. (a)$p_0 = p_1 = 0.5$, (b) $p_0 = 0.7$, $p_1 = 0.3$, (c) $p_0 = 0.3$, $p_1 = 0.7$

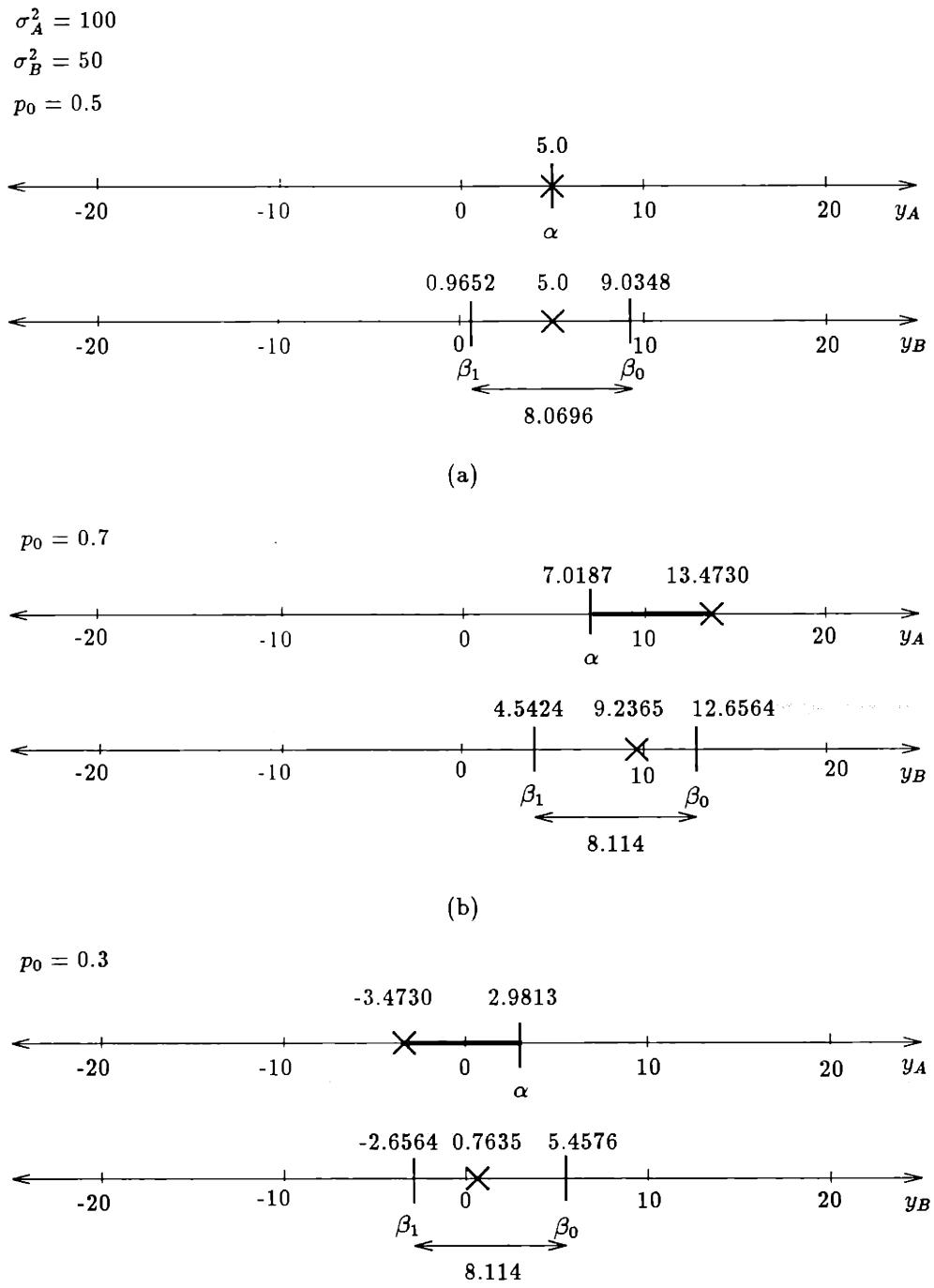


Figure 2-11: 2-Tand threshholds: DM A dumber ($\sigma_A^2 = 100$), DM B smarter ($\sigma_B^2 = 50$), $\mu_0 = 0$, $\mu_1 = 10$. (a) $p_0 = p_1 = 0.5$, (b) $p_0 = 0.7$, $p_1 = 0.3$, (c) $p_0 = 0.3$, $p_1 = 0.7$

2.5.2 Example 2: Three-Member Vee (3-Vee)

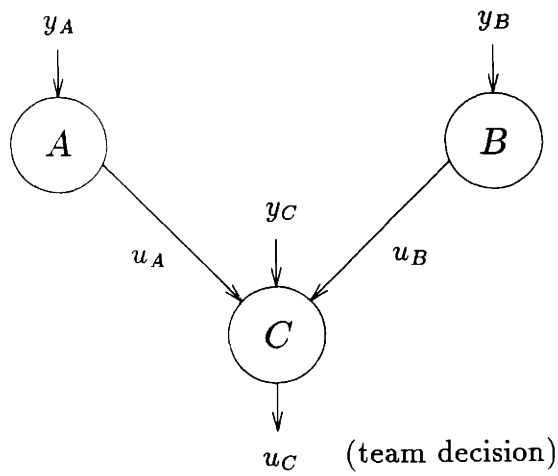


Figure 2-12: 3-Vee

The three-member Vee structure is illustrated in Figure 2-12. In contrast to the previous tandem structure, this structure represents a *parallel* structure. Since our formulation requires that a DM which receives an observation of its own act as the fusion center, this is the smallest parallel structure we can consider. The decision rules employed by DM A and DM B are of the form $\gamma_A : \mathcal{Y}_A \mapsto \{0,1\}$ and $\gamma_B : \mathcal{Y}_B \to \{0,1\}$, while the decision rule of DM C is of the form $\gamma_C : \mathcal{Y}_C \times \{0,1\} \times \{0,1\} \mapsto \{0,1\}$. DM C is the primary DM in the topology.

The necessary conditions for optimality of the decision rules γ_A, γ_B, and γ_C, under Assumptions 2.2(a), 2.3, and 2.4 were first derived in [19], and are expressed as person-by-person optimality conditions of the form:

For DM C, given fixed γ_A,γ_B:

$$\Lambda_C(y_C) = \frac{p_{Y_C|H_1}(y_C|H_1)}{p_{Y_C|H_0}(y_C|H_0)} \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \frac{p_0 \Pr(U_A=0|H_0)\Pr(U_B=0|H_0)}{p_1 \Pr(U_A=0|H_1)\Pr(U_B=0|H_1)} & \text{if } u_A=0, u_B=0 \\ \frac{p_0 \Pr(U_A=0|H_0)\Pr(U_B=1|H_0)}{p_1 \Pr(U_A=0|H_1)\Pr(U_B=1|H_1)} & \text{if } u_A=0, u_B=1 \\ \frac{p_0 \Pr(U_A=1|H_0)\Pr(U_B=0|H_0)}{p_1 \Pr(U_A=1|H_1)\Pr(U_B=0|H_1)} & \text{if } u_A=1, u_B=0 \\ \frac{p_0 \Pr(U_A=1|H_0)\Pr(U_B=1|H_0)}{p_1 \Pr(U_A=1|H_1)\Pr(U_B=1|H_1)} & \text{if } u_A=1, u_B=1 \end{cases} \tag{2.75}$$

For DM A, given fixed γ_B and γ_C:

$$\Lambda_A(y_A) = \frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \frac{p_0 \Pr(U_C=1|U_A=1,H_0) - \Pr(U_C=1|U_A=0,H_0)}{p_1 \Pr(U_C=1|U_A=1,H_1) - \Pr(U_C=1|U_A=0,H_1)} \tag{2.76}$$

For DM B, given fixed γ_A and γ_C:

$$\Lambda_B(y_B)\frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \frac{p_0 \Pr(U_C=1|U_B=1,H_0) - \Pr(U_C=1|U_B=0,H_0)}{p_1 \Pr(U_C=1|U_B=1,H_1) - \Pr(U_C=1|U_B=0,H_1)} \tag{2.77}$$

As before, we can give an equivalent representation of these decision rules in terms of the network operating points as:

For DM C, given fixed γ_A,γ_B:

$$\frac{p_{Y_C|H_1}(y_C|H_1)}{p_{Y_C|H_0}(y_C|H_0)} \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \eta \frac{(1-P_F^A)(1-P_F^B)}{(1-P_D^A)(1-P_D^B)} & \text{if } u_A=0, u_B=0 \\ \eta \frac{(1-P_F^A)P_F^B}{(1-P_D^A)P_D^B} & \text{if } u_A=0, u_B=1 \\ \eta \frac{P_F^A(1-P_F^B)}{P_D^A(1-P_D^B)} & \text{if } u_A=1, u_B=0 \\ \eta \frac{P_F^A P_F^B}{P_D^A P_D^B} & \text{if } u_A=1, u_B=1 \end{cases} \tag{2.78}$$

For DM A, given fixed γ_B and γ_C:

$$\frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \eta \frac{(1-P_F^B)[P_F^{C(10)}-P_F^{C(00)}]+P_F^B[P_F^{C(11)}-P_F^{C(01)}]}{(1-P_D^B)[P_D^{C(10)}-P_D^{C(00)}]+P_D^B[P_D^{C(11)}-P_D^{C(01)}]} \tag{2.79}$$

For DM B, given fixed γ_A and γ_C:

$$\frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \eta \frac{(1-P_F^A)[P_F^{C(01)}-P_F^{C(00)}]+P_F^A[P_F^{C(11)}-P_F^{C(10)}]}{(1-P_D^A)[P_D^{C(01)}-P_D^{C(00)}]+P_D^A[P_D^{C(11)}-P_D^{C(10)}]} \tag{2.80}$$

where P_F^A, P_D^A denote the probabilities of false alarm and detection of DM A, P_F^B, P_D^B denote the probabilities of false alarm and detection of DM B, and $P_F^{C(ij)}, P_D^{C(ij)}$ denote the probabilities of false alarm and detection of DM C when receiving messages $u_A = i, u_B = j$ with $i, j \in \{0,1\}$.

For the Gaussian problem, the optimal decision rules of equations (2.78) - (2.80) may be reduced, under the appropriate positivity conditions, to the following equivalent linear threshold rules.

For DM C, given fixed γ_A and γ_B:

$$y_C \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \xi_{00} & \text{if } u_A = 0, u_B = 0 \\ \xi_{01} & \text{if } u_A = 0, u_B = 1 \\ \xi_{10} & \text{if } u_A = 1, u_B = 0 \\ \xi_{11} & \text{if } u_A = 1, u_B = 1 \end{cases} \tag{2.81}$$

For DM A, given fixed γ_B and γ_C:

$$y_A \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \alpha \tag{2.82}$$

For DM B, given fixed γ_A and γ_C:

$$y_B \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \beta \tag{2.83}$$

where the fixed observation axis thresholds $\alpha, \beta, \xi_{00}, \xi_{01}, \xi_{10}, \xi_{11}$ satisfy the nonlinear system of equations

$$\begin{aligned}
\xi_{00} &= \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{\Phi_\alpha(0)\Phi_\beta(0)}{\Phi_\alpha(1)\Phi_\beta(1)}\right) + \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\
\xi_{01} &= \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{\Phi_\alpha(0)(1-\Phi_\beta(0))}{\Phi_\alpha(1)(1-\Phi_\beta(1))}\right) + \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\
\xi_{10} &= \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{(1-\Phi_\alpha(0))\Phi_\beta(0)}{(1-\Phi_\alpha(1))\Phi_\beta(1)}\right) + \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\
\xi_{11} &= \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{(1-\Phi_\alpha(0))(1-\Phi_\beta(0))}{(1-\Phi_\alpha(1))(1-\Phi_\beta(1))}\right) + \tfrac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\
\alpha &= \tfrac{\sigma_A^2}{\mu_1-\mu_0} \ln\left(\tfrac{\Phi_\beta(0)[\Phi_{\xi_{00}}(0)-\Phi_{\xi_{10}}(0)]+(1-\Phi_\beta(0))[\Phi_{\xi_{01}}(0)-\Phi_{\xi_{11}}(0)]}{\Phi_\beta(1)[\Phi_{\xi_{00}}(1)-\Phi_{\xi_{10}}(1)]+(1-\Phi_\beta(1))[\Phi_{\xi_{01}}(1)-\Phi_{\xi_{11}}(1)]}\right) \\
&\quad + \tfrac{\sigma_A^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2} \\
\beta &= \tfrac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\tfrac{\Phi_\alpha(0)[\Phi_{\xi_{00}}(0)-\Phi_{\xi_{01}}(0)]+(1-\Phi_\alpha(0))[\Phi_{\xi_{10}}(0)-\Phi_{\xi_{11}}(0)]}{\Phi_\alpha(1)[\Phi_{\xi_{00}}(1)-\Phi_{\xi_{01}}(1)]+(1-\Phi_\alpha(1))[\Phi_{\xi_{10}}(1)-\Phi_{\xi_{11}}(1)]}\right) \\
&\quad + \tfrac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\tfrac{p_0}{p_1}\right) + \tfrac{\mu_0+\mu_1}{2}
\end{aligned} \tag{2.84}$$

where the functions Φ are defined analogously to those in equations (2.72)-(2.74)

2.5.3 Example 3: Three-Member Tandem (3-Tand)

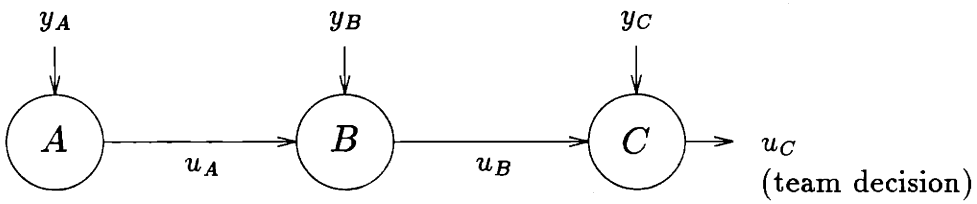


Figure 2-13: 3-Tand

The three-member tandem topology is illustrated in Figure 2-13. Our interest in this structure is that it is the first topology we have seen that contains a DM which is not in direct contact with the primary DM. In particular, DM A affects the primary DM C only through B. It also represents an alternative arrangement of 3 DMs which can be compared to 3-Vee. The decision rule employed by DM A is of the form $\gamma_A : \mathcal{Y}_A \mapsto \{0,1\}$, while the rules of DM B and DM C are of the forms $\gamma_B : \mathcal{Y}_B \times \{0,1\} \mapsto \{0,1\}$ and $\gamma_C : \mathcal{Y}_C \times \{0,1\} \mapsto \{0,1\}$, respectively. DM C is the primary DM in the topology.

The necessary conditions for optimality of the decision rules γ_A, γ_B, and γ_C, under Assumptions 2.2(a), 2.3, and 2.4, are given by

For DM C, given fixed γ_A, γ_B:

$$\Lambda_C(y_C) \triangleq \frac{p_{Y_C|H_1}(y_C|H_1)}{p_{Y_C|H_0}(y_C|H_0)} \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \frac{p_0 \Pr(U_B=0|H_0)}{p_1 \Pr(U_B=0|H_1)} & \text{if } u_B = 0 \\ \\ \frac{p_0 \Pr(U_B=1|H_0)}{p_1 \Pr(U_B=1|H_1)} & \text{if } u_B = 1 \end{cases} \tag{2.85}$$

For DM B, given fixed γ_A, γ_C:

$$\Lambda_B(y_B) \triangleq \frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}}$$

$$\begin{cases} \frac{p_0 \Pr(U_A=0|H_0)[\Pr(U_C=1|U_B=1,H_0) - \Pr(U_C=1|U_B=0,H_0)]}{p_1 \Pr(U_A=0|H_1)[\Pr(U_C=1|U_B=1,H_1) - \Pr(U_C=1|U_B=0,H_1)]} & \text{if } u_A = 0 \\ \\ \frac{p_0 \Pr(U_A=1|H_0)[\Pr(U_C=1|U_B=1,H_0) - \Pr(U_C=1|U_B=0,H_0)]}{p_1 \Pr(U_A=1|H_1)[\Pr(U_C=1|U_B=1,H_1) - \Pr(U_C=1|U_B=0,H_1)]} & \text{if } u_A = 1 \end{cases} \tag{2.86}$$

For DM A, given fixed γ_B, γ_C:

$$\Lambda_A(y_A) \triangleq \frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \frac{p_0[\Pr(U_C=1|U_A=1,H_0) - \Pr(U_C=1|U_A=0,H_0)]}{p_1[\Pr(U_C=1|U_A=1,H_1) - \Pr(U_C=1|U_A=0,H_1)]} \tag{2.87}$$

Note that the decision rules of DM C and DM A are of the same form as they were for DMs B and A in 2-Tand. To express these rules in operating point form, let P_F^A, P_D^A denote the probabilities of false alarm and detection of DM A, P_F^{Bi}, P_D^{Bi}

denote the probabilities of false alarm and detection of DM B given that it receives message $u_A = i;\ i \in \{0,1\}$, and P_F^{Ci}, P_D^{Ci} the probabilities of false alarm and detection of DM C when DM B selects message $u_B = i;\ i \in \{0,1\}$. Then the decision rules are expressible as follows.

For DM C, given fixed γ_A, γ_B:

$$\frac{p_{Y_C|H_1}(y_C|H_1)}{p_{Y_C|H_0}(y_C|H_0)} \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \eta \frac{(1-P_F^A)(1-P_F^{B0})+P_F^A(1-P_F^{B1})}{(1-P_D^A)(1-P_D^{B0})+P_D^A(1-P_D^{B1})} & \text{if } u_B = 0 \\ \eta \frac{(1-P_F^A)P_F^{B0}+P_F^A P_F^{B1}}{(1-P_D^A)P_D^{B0}+P_D^A P_D^{B1}} & \text{if } u_B = 1 \end{cases} \tag{2.88}$$

For DM B, given fixed γ_A, γ_C:

$$\frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \eta \frac{(1-P_F^A)[P_F^{C1}-P_F^{C0}]}{(1-P_D^A)[P_D^{C1}-P_D^{C0}]} & \text{if } u_A = 0 \\ \eta \frac{P_F^A[P_F^{C1}-P_F^{C0}]}{P_D^A[P_D^{C1}-P_D^{C0}]} & \text{if } u_A = 1 \end{cases} \tag{2.89}$$

For DM A, given fixed γ_B, γ_C:

$$\frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \eta \frac{(P_F^{B1}-P_F^{B0})(P_F^{C1}-P_F^{C0})}{(P_D^{B1}-P_D^{B0})(P_D^{C1}-P_D^{C0})} \tag{2.90}$$

For the linear Gaussian problem, these rules can be reduced, under the appropriate positivity conditions, to the following equivalent set of rules.

For DM C, given fixed γ_A and γ_B:

$$y_C \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \xi_0 & \text{if } u_B = 0 \\ \xi_1 & \text{if } u_B = 1 \end{cases} \tag{2.91}$$

For DM B, given fixed γ_A and γ_C:

$$y_B \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \beta_0 & \text{if } u_A = 0 \\ \\ \beta_1 & \text{if } u_A = 1 \end{cases} \tag{2.92}$$

For DM A, given fixed γ_B and γ_C:

$$y_A \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \alpha \tag{2.93}$$

where the fixed *observation axis* thresholds $\alpha, \beta_0, \beta_1, \xi_0, \xi_1$ satisfy the system of non-linear equations

$$\begin{aligned}
\xi_0 &= \frac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\frac{\Phi_\alpha(0)\Phi_{\beta_0}(0)+(1-\Phi_\alpha(0))\Phi_{\beta_1}(0)}{\Phi_\alpha(1)\Phi_{\beta_0}(1)+(1-\Phi_\alpha(1))\Phi_{\beta_1}(1)}\right) + \frac{\sigma_C^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2} \\
\xi_1 &= \frac{\sigma_C^2}{\mu_1-\mu_0} \ln\left(\frac{\Phi_\alpha(0)(1-\Phi_{\beta_0}(0))+(1-\Phi_\alpha(0))(1-\Phi_{\beta_1}(1))}{\Phi_\alpha(1)(1-\Phi_{\beta_0}(1))+(1-\Phi_\alpha(1))(1-\Phi_{\beta_1}(1))}\right) + \frac{\sigma_C^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2} \\
\beta_0 &= \frac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\frac{\Phi_\alpha(0)(\Phi_{\xi_0}(0)-\Phi_{\xi_1}(0))}{\Phi_\alpha(1)(\Phi_{\xi_0}(1)-\Phi_{\xi_1}(1))}\right) + \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2} \\
\beta_1 &= \frac{\sigma_B^2}{\mu_1-\mu_0} \ln\left(\frac{(1-\Phi_\alpha(0))(\Phi_{\xi_0}(0)-\Phi_{\xi_1}(0))}{(1-\Phi_\alpha(1))(\Phi_{\xi_0}(1)-\Phi_{\xi_1}(1))}\right) + \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2} \\
\alpha &= \frac{\sigma_A^2}{\mu_1-\mu_0} \ln\left(\frac{(\Phi_{\beta_0}(0)-\Phi_{\beta_1}(0))(\Phi_{\xi_0}(0)-\Phi_{\xi_1}(0))}{(\Phi_{\beta_0}(1)-\Phi_{\beta_1}(1))(\Phi_{\xi_0}(1)-\Phi_{\xi_1}(1))}\right) + \frac{\sigma_A^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2}
\end{aligned} \tag{2.94}$$

specifying the necessary conditions for optimality, where the functions Φ are defined analogously to those in equations (2.72)-(2.74).

2.5.4 Example 4: Four-Member Asymmetric (4-Asym)

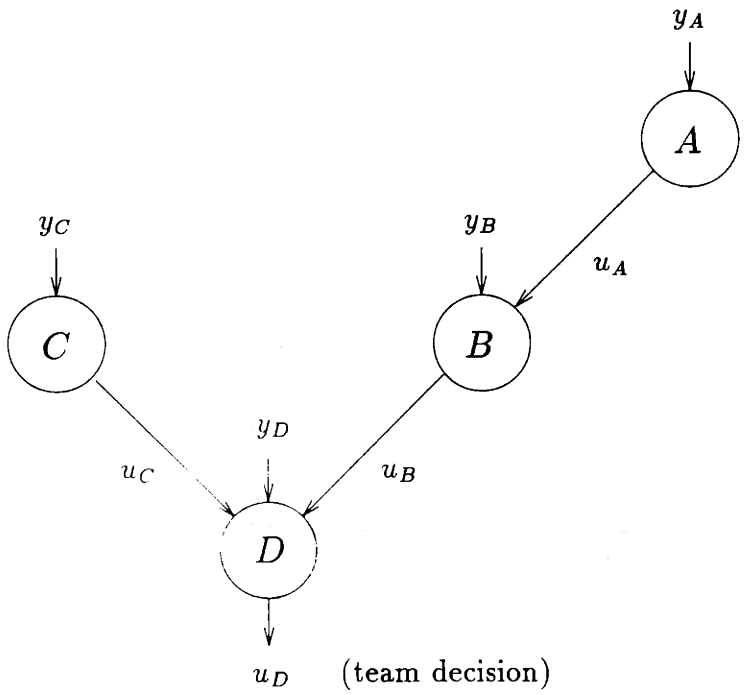


Figure 2-14: 4-Asym

The four-member asymmetric topology is illustrated in Figure 2-14. This last team is included so that at least one asymmetric topology is examined, and so that the explosion of complexity in the decision rules as DMs are added may be illustrated. The decision rule employed by DMs A and C are of the form $\gamma_A : \mathcal{Y}_A \mapsto \{0,1\}$ and $\gamma_C : \mathcal{Y}_C \mapsto \{0,1\}$, while the decision rule of DM B is of the form $\gamma_B : \mathcal{Y}_B \times \{0,1\} \mapsto \{0,1\}$, and the decision rule for DM D is of the form $\gamma_D : \mathcal{Y}_D \times \{0,1\} \times \{0,1\} \mapsto \{0,1\}$. DM D is the primary DM in the topology.

For this team it is easiest to express the decision rules directly in operating point form at the outset[10]. Let P_F^A, P_D^A denote the probabilities of false alarm and detection of DM A, P_F^C, P_D^C denote the probabilities of false alarm and detection of DM C, P_F^{Bi}, P_D^{Bi} denote the probabilities of false alarm and detection of DM B given that it

[10]The reason for this is that the rules are more easily derived in this form as discussed in Chapter 3.

receives message $u_A = i$; $i \in \{0,1\}$, and $P_F^{D(ij)}, P_D^{D(ij)}$ the probabilities of false alarm and detection of DM D when it receives the incoming messages $u_B = i, u_C = j$, $i,j \in \{0,1\}$. Then the necessary conditions for optimality of the decision rules γ_A, γ_B, γ_C, and γ_D, under Assumptions 2.2(a), 2.3, and 2.4, are given by

For DM D, given fixed γ_A, γ_B, and γ_C:

$$\Lambda_D(y_D) \triangleq \frac{p_{Y_D|H_1}(y_D|H_1)}{p_{Y_D|H_0}(y_D|H_0)} \underset{u_D=0}{\overset{u_D=1}{\gtrless}}$$

$$\begin{cases} \eta \frac{(1-P_F^A)(1-P_F^{B0})(1-P_F^C)+P_F^A(1-P_F^{B1})(1-P_F^C)}{(1-P_D^A)(1-P_D^{B0})(1-P_D^C)+P_D^A(1-P_D^{B1})(1-P_D^C)} & \text{if } u_B = 0,\ u_C = 0 \\ \eta \frac{(1-P_F^A)(1-P_F^{B0})P_F^C+P_F^A(1-P_F^{B1})P_F^C}{(1-P_D^A)(1-P_D^{B0})P_D^C+P_D^A(1-P_D^{B1})P_D^C} & \text{if } u_B = 0,\ u_C = 1 \\ \eta \frac{(1-P_F^A)P_F^{B0}(1-P_F^C)+P_F^A P_F^{B1}(1-P_F^C)}{(1-P_D^A)P_D^{B0}(1-P_D^C)+P_D^A P_D^{B1}(1-P_D^C)} & \text{if } u_B = 1,\ u_C = 0 \\ \eta \frac{(1-P_F^A)P_F^{B0}P_F^C+P_F^A P_F^{B1}P_F^C}{(1-P_D^A)P_D^{B0}P_D^C+P_D^A P_D^{B1}P_D^C} & \text{if } u_B = 1,\ u_C = 1 \end{cases} \tag{2.95}$$

For DM B, given fixed γ_A, γ_C, γ_D:

$$\Lambda_B(y_B) \triangleq \frac{p_{Y_B|H_1}(y_B|H_1)}{p_{Y_B|H_0}(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}}$$

$$\begin{cases} \eta \frac{(1-P_F^A)[(1-P_F^C)(P_F^{D(10)}-P_F^{D(00)})+P_F^C(P_F^{D(11)}-P_F^{D(01)})]}{(1-P_D^A)[(1-P_D^C)(P_D^{D(10)}-P_D^{D(00)})+P_D^C(P_D^{D(11)}-P_D^{D(01)})]} & \text{if } u_A = 0 \\ \eta \frac{P_F^A[(1-P_F^C)(P_F^{D(10)}-P_F^{D(00)})+P_F^C(P_F^{D(11)}-P_F^{D(01)})]}{P_D^A[(1-P_D^C)(P_D^{D(10)}-P_D^{D(00)})+P_D^C(P_D^{D(11)}-P_D^{D(01)})]} & \text{if } u_A = 1 \end{cases} \tag{2.96}$$

For DM C, given fixed γ_A, γ_B, γ_D:

$$\Lambda_C(y_C) \triangleq \frac{p_{Y_C|H_1}(y_C|H_1)}{p_{Y_C|H_0}(y_C|H_0)} \underset{u_C=0}{\overset{u_C=1}{\gtrless}}$$

$$\eta \frac{(1-P_F^A)((1-P_F^{B0})(P_F^{D(01)}-P_F^{D(00)})+P_F^{B0}(P_F^{D(11)}-P_F^{D(10)}))}{(1-P_D^A)((1-P_D^{B0})(P_D^{D(01)}-P_D^{D(00)})+P_D^{B0}(P_D^{D(11)}-P_D^{D(10)}))} \cdots$$

$$\cdots \frac{+P_F^A((1-P_F^{B1})(P_F^{D(01)}-P_F^{D(00)})+P_F^{B1}(P_F^{D(11)}-P_F^{D(10)}))}{+P_D^A((1-P_D^{B1})(P_D^{D(01)}-P_D^{D(00)})+P_D^{B1}(P_D^{D(11)}-P_D^{D(10)}))} \tag{2.97}$$

For DM A, given fixed γ_B, γ_C, γ_D:

$$\Lambda_A(y_A) \triangleq \frac{p_{Y_A|H_1}(y_A|H_1)}{p_{Y_A|H_0}(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}}$$

$$\eta \frac{-(1-P_F^{B0})((1-P_F^C)P_F^{D(00)}+P_F^C P_F^{D(01)})}{(1-P_D^{B0})((1-P_D^C)(1-P_D^{D(00)})+P_D^C(1-P_D^{D(01)}))} \cdots$$
$$\cdots \frac{-P_F^{B0}((1-P_F^C)P_F^{D(10)}+P_F^C P_F^{D(11)})}{+P_D^{B0}((1-P_D^C)(1-P_D^{D(10)})+P_D^C(1-P_D^{D(11)}))} \cdots$$
$$\cdots \frac{+(1-P_F^{B1})((1-P_F^C)P_F^{D(00)}+P_F^C P_F^{D(01)})}{-(1-P_D^{B1})(P_D^C(1-P_D^{D(01)})+(1-P_D^C)(1-P_D^{D(00)}))} \cdots$$
$$\cdots \frac{+P_F^{B1}((1-P_F^C)P_F^{D(10)}+P_F^C P_F^{D(11)})}{+P_D^{B1}((1-P_D^C)(1-P_D^{D(10)})+P_D^C(1-P_D^{D(11)}))} \tag{2.98}$$

For the linear Gaussian problem, these rules can be reduced, under the appropriate positivity assumptions, to the following equivalent set of rules.

For DM D, given fixed γ_A, γ_B, and γ_C:

$$y_D \underset{u_D=0}{\overset{u_D=1}{\gtrless}} \begin{cases} \zeta_{00} & \text{if } u_B=0,\ u_C=0 \\ \zeta_{01} & \text{if } u_B=0,\ u_C=1 \\ \zeta_{10} & \text{if } u_B=1,\ u_C=0 \\ \zeta_{11} & \text{if } u_B=1,\ u_C=1 \end{cases} \tag{2.99}$$

For DM B, given fixed γ_A, γ_C, γ_D:

$$y_B \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \beta_0 & \text{if } u_A=0 \\ \beta_1 & \text{if } u_A=1 \end{cases} \tag{2.100}$$

For DM C, given fixed γ_A, γ_B, γ_D:

$$y_C \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \xi \tag{2.101}$$

For DM A, given fixed γ_B, γ_C, γ_D:

$$y_A \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \alpha \tag{2.102}$$

where the fixed *observation axis* thresholds α,β_0,β_1,ξ, ζ_{00},ζ_{01},ζ_{10} and ζ_{11} satisfy the system of nonlinear equations

$$\begin{aligned}
\zeta_{00} &= \frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)\Phi_{\beta_0}(0)\Phi_\xi(0)+(1-\Phi_\alpha(0))\Phi_{\beta_1}(0)\Phi_\xi(0)}{\Phi_\alpha(1)\Phi_{\beta_0}(1)\Phi_\xi(1)+(1-\Phi_\alpha(1))\Phi_{\beta_1}(1)\Phi_\xi(1)}\right) \\
&\quad +\frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2} \\
\zeta_{01} &= \frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)\Phi_{\beta_0}(0)(1-\Phi_\xi(0))+(1-\Phi_\alpha(0))\Phi_{\beta_1}(0)(1-\Phi_\xi(0))}{\Phi_\alpha(1)\Phi_{\beta_0}(1)(1-\Phi_\xi(1))+(1-\Phi_\alpha(1))\Phi_{\beta_1}(1)(1-\Phi_\xi(1)0}\right) \\
&\quad +\frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2} \\
\zeta_{10} &= \frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)(1-\Phi_{\beta_0}(0))\Phi_\xi(0)+(1-\Phi_\alpha(0))(1-\Phi_{\beta_1}(0))\Phi_\xi(0)}{\Phi_\alpha(1)(1-\Phi_{\beta_0}(1))\Phi_\xi(1)+(1-\Phi_\alpha(1))(1-\Phi_{\beta_1}(1))\Phi_\xi(1)}\right) \\
&\quad +\frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2} \\
\zeta_{11} &= \frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)(1-\Phi_{\beta_0}(0))(1-\Phi_\xi(0))+(1-\Phi_\alpha(0))(1-\Phi_{\beta_1}(0))(1-\Phi_\xi(0))}{\Phi_\alpha(1)(1-\Phi_{\beta_0}(1))(1-\Phi_\xi(1))+(1-\Phi_\alpha(1))(1-\Phi_{\beta_1}(1))(1-\Phi_\xi(1))}\right) \\
&\quad +\frac{\sigma_D^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2}
\end{aligned} \tag{2.103}$$

$$\begin{aligned}
\xi &= \frac{\sigma_C^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)(\Phi_{\beta_0}(0)(\Phi_{\zeta_{00}}(0)-\Phi_{\zeta_{01}}(0))+(1-\Phi_{\beta_0}(0))(\Phi_{\zeta_{10}}(0)-\Phi_{\zeta_{11}}(0)))}{\Phi_\alpha(1)(\Phi_{\beta_1}(1)(\Phi_{\zeta_{00}}(1)-\Phi_{\zeta_{01}}(1))+(1-\Phi_{\beta_0}(1))(\Phi_{\zeta_{10}}(1)-\Phi_{\zeta_{11}}(1)))}\cdots\right. \\
&\quad \left.\cdots\frac{+(1-\Phi_\alpha(0))(\Phi_{\beta_1}(0)(\Phi_{\zeta_{00}}(0)-\Phi_{\zeta_{01}}(0))+(1-\Phi_{\beta_1}(0))(\Phi_{\zeta_{10}}(0)-\Phi_{\zeta_{11}}(0)))}{(1-\Phi_\alpha(1))(\Phi_{\beta_1}(\Phi_{\zeta_{00}}(1)-\Phi_{\zeta_{01}}(1))+(1-\Phi_{\beta_1}(1))(\Phi_{\zeta_{10}}(1)-\Phi_{\zeta_{11}}(1)))}\right) \\
&\quad +\frac{\sigma_C^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2}
\end{aligned} \tag{2.104}$$

$$\beta_0 = \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)[\Phi_\xi(0)(\Phi_{\zeta_{00}}(0)-\Phi_{\zeta_{10}}(0))+(1-\Phi_\xi(0))(\Phi_{\zeta_{01}}(0)-\Phi_{\zeta_{11}}(0))]}{\Phi_\alpha(1)[\Phi_\xi(1)(\Phi_{\zeta_{00}}(1)-\Phi_{\zeta_{10}}(1))+(1-\Phi_\xi(1))(\Phi_{\zeta_{01}}-\Phi_{\zeta_{11}}(1))]}\right) + \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2} \tag{2.105}$$

$$\beta_1 = \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{(1-\Phi_\alpha(0))[\Phi_\xi(0)(\Phi_{\zeta_{00}}(0)-\Phi_{\zeta_{10}}(0))+(1-\Phi_\xi(0))(\Phi_{\zeta_{01}}(0)-\Phi_{\zeta_{11}}(0))]}{(1-\Phi_\alpha(1))[\Phi_\xi(1)(\Phi_{\zeta_{00}}(1)-\Phi_{\zeta_{10}}(1))+(1-\Phi_\xi(1))(\Phi_{\zeta_{01}}-\Phi_{\zeta_{11}}(1))]}\right) + \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right) + \frac{\mu_0+\mu_1}{2}$$

$$\begin{aligned}\alpha = \frac{\sigma_A^2}{\mu_1-\mu_0}\ln\Bigg(&\frac{-\Phi_{\beta_0}(0)(\Phi_\xi(0)(1-\Phi_{\zeta_{00}}(0))+(1-\Phi_\xi(0))(1-\Phi_{\zeta_{01}}(0)))}{\Phi_{\beta_0}(1)(\Phi_\xi(1)\Phi_{\zeta_{00}}(1)+(1-\Phi_\xi(1))\Phi_{\zeta_{01}}(1))}\cdots\\ &\cdots\frac{-(1-\Phi_{\beta_0}(0))(\Phi_\xi(0)(1-\Phi_{\zeta_{10}}(0))+(1-\Phi_\xi(0))(1-\Phi_{\zeta_{11}}(0)))}{+(1-\Phi_{\beta_0}(1))(\Phi_\xi(1)\Phi_{\zeta_{10}}(1)+(1-\Phi_\xi(1))\Phi_{\zeta_{11}}(1))}\cdots\\ &\cdots\frac{+\Phi_{\beta_1}(0)(\Phi_\xi(0)(1-\Phi_{\zeta_{00}}(0))+(1-\Phi_\xi(0))(1-\Phi_{\zeta_{01}}(0)))}{-\Phi_{\beta_1}(1)((1-\Phi_\xi(1))\Phi_{\zeta_{01}}(1)+\Phi_\xi(1)\Phi_{\zeta_{00}}(1))}\cdots\\ &\cdots\frac{+(1-\Phi_{\beta_1}(0))(\Phi_\xi(0)(1-\Phi_{\zeta_{10}}(0))+(1-\Phi_\xi(0))(1-\Phi_{\zeta_{11}}(0)))}{+(1-\Phi_{\beta_1}(1))(\Phi_\xi(1)\Phi_{\zeta_{10}}(1)+(1-\Phi_\xi(1))\Phi_{\zeta_{11}}(1))}\Bigg)\\ &+\frac{\sigma_A^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2}\end{aligned} \tag{2.106}$$

specifying the necessary conditions for optimality, where the functions Φ are defined analogously to those in equations (2.72)-(2.74).

2.6 Chapter Conclusions

In this chapter we examined in detail the optimal decision rules for the single DM problem, and a special class of network problem in which each node receives a conditionally independent observation, is allowed to transmit a binary-valued message, and where the network topology is a tree configuration with the root node making the overall team decision in order to minimize the probability of team error. This class was found to possess several interesting properties.

The necessary conditions for optimality for such networks were not expressible in closed form, but instead had to be expressed in the form of person-by-person optimality conditions, which specified the necessary condition for optimality for each decision rule, given that the rest of the network decision rules were held fixed. The decision rules were found to comprise coupled likelihood ratio tests. Each DM uses one of a set of LRTs, where the test used is selected by the particular combination

of incoming messages from upstream DMs. The coupling enters in the form of cost coefficients which affect the LRT in the same way as costs enter the LRT for general Bayes hypothesis testing. If the LRT(s) being used by a given DM are viewed from a local point of view, then all the information regarding the rest of the network that is necessary for the DM to make an optimal decision, in the context of team performance, is captured by these costs. The specific information required was found to be the conditional operating points of all of the other network DMs. We will see in the next chapter that the analytic form of these coupling costs is completely determined by the topology of the network, while the actual values of the costs require knowledge of the current set of conditional operating points of the rest of the network DMs. The operating point information can be thought of as capturing "state" information regarding the current performance of the rest of the teams members, while the exact way the members are interconnected determines how that information is fused together in computing a particular DM's cost. It was clearly evident that the computations become complicated, even for the small examples we considered, thus giving evidence of the noncombinatorial-type complexity that characterizes the underlying optimization of the decision rules.

From a broader perspective, what the DBHT framework provides is a class of stochastic team decision problem with particularly nice mathematical structure. The optimal action of each member of the team is to perform a local statistical decision test on a local observation, where the local test is appropriately biased to take into consideration how the DM fits into the overall topology of the team, as well as the current state of the rest of the team's members. The framework also has a notion of capability or expertise which is mathematically quantifiable by the ROC curve. This makes it easy to separate the notions of a DM's capability from its current performance. Coupling in the form of local cost changes manifests as adjustments of operating point along the ROC, i.e., adjustments of performance within a DMs range of capability. Furthermore, the restricted message set of each DM in our particular DBHT setting forces each DM to operate under conditions of partial or incomplete information.

Even more favorable is the fact that, for the Gaussian detection problem, the optimal team decision rules take an extremely simple form. In particular, they are given by linear threshold tests in which each DM's observation is compared with an observation axis threshold. Necessary conditions for optimality of the network observation thresholds were not computable in closed form, but instead were expressible as the solution to a system of coupled nonlinear equations. In the next chapter, we adopt this linear threshold parameterization of the decision rules for the remainder of this report. It is extremely useful for numerical experiments to have a parameterization include the optimal rules for a class of problems of interest, since we may then compare the rules generated by the training algorithms with the rules we know a priori to be optimal.

Chapter 3

Optimization Using Complete Statistics

From the point of view of optimizing the network decision rules, the results of the previous chapter give reason for optimism in several respects. In the first place, the class of problems in which we are interested has optimal decision rules which are likelihood ratio tests with constant thresholds, so that the problem of determining the optimal decision rules may be immediately reduced to searching over the class of threshold rules. Secondly, we found that for Gaussian detection problems, the search could be further restricted, without loss of optimality, to the class of linear threshold rules, in which the statistics do not appear explicitly, and for which each rule is parameterized by a single real-valued scalar. On the negative side, we discovered that no closed-form analytic expression for the necessary conditions for optimality existed, but that they instead had to be expressed as person-by-person optimality conditions, specifying the necessary condition for optimality of a single network decision rule, given that the other decision rules of the network were held fixed. Person-by-person optimal solutions must therefore be computed using iterative numerical techniques.

Computation of the optimal decision thresholds in a decentralized binary hypothesis testing problem with M DMs requires the following pieces of information

1. Prior probabilities p_0, p_1

2. One of the following, for all $i = 1, \ldots, M$:

   - $p_{Y_i|H_0}(y_i|H_0), p_{Y_i|H_1}(y_i|H_1)$
   - ROC_i
   - $\Lambda_i(y_i)$

3. Network topology

4. Cost structure (in general Bayes case)

Given complete knowledge of the problem statistics and network structure, there are a variety of approaches for obtaining the optimal decision rules, each with its own advantages and disadvantages. The schemes differ depending on which variables of optimization are selected and the form in which the problem statistics are available.

For example, one possible approach is to formulate the problem as a deterministic nonlinear optimization of the probability of error P_ϵ over whatever parameterization is chosen. If only the likelihood ratios are known, then unconstrained optimization over the likelihood ratio thresholds may be performed. If the ROC curves are known, then a constrained optimization of the operating points (P_F^i, P_D^i) along each ROC i may be performed. If the functional forms of the conditional densities are known, then all options are available, since the likelihood ratios or ROCs are then easily computed. Tang [59] examines a variety of iterative numerical methods suitable for nonlinear programs, such as steepest descent, conjugate gradient, and nonlinear Gauss-Seidel cyclic coordinate descent, given complete characterization of the statistics. Further discussion of the nonlinear Gauss-Seidel algorithm may be found in [60]. Tang performs the optimization over the network operating points in all cases. One of the main disadvantages of the nonlinear programming approach is that the programs can become quite large, even for small teams, due to the combinatorics involved. And of course, it is generally unknown whether the solution obtained is a local or global minimum.

A second approach is to formulate the problem as a deterministic nonlinear optimal control problem, as suggested by Tang [59] and as we present in Section 3.2.3,

and utilize a technique from optimal control theory such as gradient descent using the adjoint method, min-H, or spatial dynamic programming. The advantage of the optimal control approach is that the stagewise nature of the problem is directly exploited to permit solution of a series of smaller dimensional problems in place of the large dimensional nonlinear program.

A third approach is to try and directly solve the system of equations that define the necessary conditions for optimality. The difficulty of this approach is that this system of equations is nonlinear and highly coupled, so that this problem is no easier than solving the original optimization.

The present chapter overviews each of these approaches assuming the availability of full knowledge of the problem statistics. The purpose of this discussion is to establish certain properties of the optimization before moving on to the stochastic problem in Chapter 4. In Section 3.1 we define the parameterization of the decision rules as linear thresholds. In Section 3.2 we present several alternative perspectives on the optimization which we believe illuminate some of its relevant structure, and suggest how it might be exploited for optimization. In Section 3.3 we take an in-depth look at the probability of error criterion function which results from the parameterization described in Section 3.1. Our goal here is twofold; we wish to investigate its differentiability and smoothness properties as a function of the properties of the underlying conditional densities of the hypothesis test, as well as examine the shape of its surface through numerical experiments. Our overall purpose is to assess whether or not it can be expected to admit optimization by gradient-based methods. In Section 3.4 we then briefly discuss the application of gradient methods to the deterministic problem in order to indicate the pitfalls we expect to be encountered by the stochastic gradient methods we consider in Chapters 4 and 5. Finally, in Section 3.5 we discuss the issues involved with determining the optimal network thresholds by performing a fixed point iteration on the necessary conditions.

3.1 Parameterization by Linear Threshold Rules

In this section, we describe in detail the particular parameterization of the decision rules with which we will be exclusively concerned in the remainder of this report.

As alluded to earlier, if the decision rules are restricted to be linear threshold tests, then *unconstrained* optimization of the observation thresholds can be performed. This parameterization lends itself to nonparametric optimization because the statistics are not explicitly modeled, and because it is the simplest parameterization possible, with each component LRT parameterized by a single real-valued scalar. Furthermore, this class of rules includes the optimal rules for the Gaussian problem, and restriction to linear thresholds is consistent with the use of 0-1 messages.

A point on which we must be clear is that the parameterization we require does not involve each DM using a single such linear threshold rule, but rather a number of rules determined by the number of upstream DMs from which it receives decisions. We introduce the following formalism, cf. [65], to make this statement precise. For a given DM i, define the set of predecessors of DM i to be $\mathcal{P}(i)$. Then the number of thresholds utilized by DM i is $2^{|\mathcal{P}(i)|}$, where the notation $|\mathcal{P}|$ denotes cardinality of the set $\mathcal{P}$. For a network of M DMs, the total number of thresholds required, and thus the dimension of the associated parameter vector for the network, is given by

$$\sum_{i=1}^{M} 2^{|\mathcal{P}(i)|} \tag{3.1}$$

For example, for the 4-Asym network (Figure 2-14)

$$\begin{aligned} \mathcal{P}(A) &= \{\emptyset\}, \quad & \mathcal{P}(B) &= \{A\} \\ \mathcal{P}(C) &= \{\emptyset\}, \quad & \mathcal{P}(D) &= \{B, C\} \end{aligned} \tag{3.2}$$

so that

$$\begin{aligned} 2^{|\mathcal{P}(A)|} &= 1, \quad & 2^{|\mathcal{P}(B)|} &= 2 \\ 2^{|\mathcal{P}(C)|} &= 1, \quad & 2^{|\mathcal{P}(D)|} &= 4 \end{aligned} \tag{3.3}$$

and a linear threshold parameterization of the network decision rules requires a total of 8 observation thresholds. Notice that the dimension of the parameter vector N necessarily exceeds the number of network DMs.

Thus far, we have used the greek letters $\alpha, \beta, \xi, \zeta$ to denote the observation thresholds. In the remainder of this report, we adopt the notation $\underline{\theta}$ to denote a general vector of network threshold parameters. For example, for the 2-Tand network, $\underline{\theta} \in \Re^3$ represents

$$\begin{aligned} \underline{\theta} &= [\theta_1, \theta_2, \theta_3]^T \\ &= [\alpha, \beta_0, \beta_1] \end{aligned} \tag{3.4}$$

For our purposes, the exact order of assignment of the components of $\underline{\theta}$ will be unimportant. We should also note that the thresholds are treated as independently tunable parameters by our algorithms, despite that fact that the form of the optimal decision rule often imposes a relationship between the values (see for example (2.62)).

When we discuss performance, either probability of error or Bayes cost, we will make the argument $\underline{\theta}$ explicit to indicate that we are referring to the cost under the parameterization just described. Thus, the probability of error of a single DM using a linear threshold rule will be denoted $P_\epsilon(\theta)$ while the probability of error for a general network parameterized by linear threshold rules as described above will be indicated by $P_\epsilon(\underline{\theta})$. When a particular network is indicated, it will be specified with a superscript, such as in $P_\epsilon^{2-Tand}(\underline{\theta})$. We extend the definition of the set $\mathcal{T}$, defined in Section 2.2.1 for a single DM ($M = 1$), to include the class of linear threshold parameterizations for networks ($M > 1$).

It should be emphasized that parameterization by linear threshold rules does *not* imply a linear parameterization of the cost $J_B(\theta)$ or $P_\epsilon(\underline{\theta})$, both of which are highly nonlinear functions of the parameters θ or $\underline{\theta}$, respectively. It is also important to note that restriction of the optimization to the set $\mathcal{T}$ means that the performance achieved by the DM or team of DMs will in general be suboptimal with respect to the best performance possible, except for the case of the Gaussian detection problem.

3.2 Alternative Perspectives

In this section we suggest two alternative ways of visualizing the optimization of the network decision rules which we believe provide significant intuition into the operation of the networks. We then present an alternative formulation of the DBHT problem as a deterministic nonlinear optimal control problem.

3.2.1 Observation Space Geometry

It is instructive to view the decision boundaries established by the optimal decentralized solution vis-a-vis the corresponding decision boundaries of the centralized solution. For simplicity, we examine the decision boundaries in observation space, although the same sort of insight could be gained in the more general setting of likelihood ratio space. The entire observation space may be viewed in this way only for two-dimensional problems. Accordingly, we illustrate the viewpoint for the 2-Tand network as it is the only team with a two-dimensional observation space. For teams comprised of more than two DM's, only two-dimensional slices of the higher dimensional observation space can be viewed in this manner.

Figure 3-1 illustrates the effective parsing of the $y_A - y_B$ observation space created by the 2-Tand team. The threshold α corresponds to the vertical line $y_A = \alpha$, while the thresholds β_0 and β_1 correspond to the parallel horizontal lines $y_B = \beta_0$ and $y_B = \beta_1$. The separating surface is piecewise linear in nature since the scalar thresholds of each DM yield separating hyperplanes which are perpendicular to the observation axis of that DM. The parameterization described in Section 3.1 effectively constrains the separating surface of the general team problem to be of this piecewise linear[1] nature as well. The problem of optimally parsing the space can be viewed as a quantization problem.

In Figure 3-2, a typical configuration of the decision regions of 2-Tand for the Gaussian detection problem are shown with respect to typical decision regions of the

[1]We retain the terminology piecewise linear although in higher dimensional problems the lines are hyperplanes.

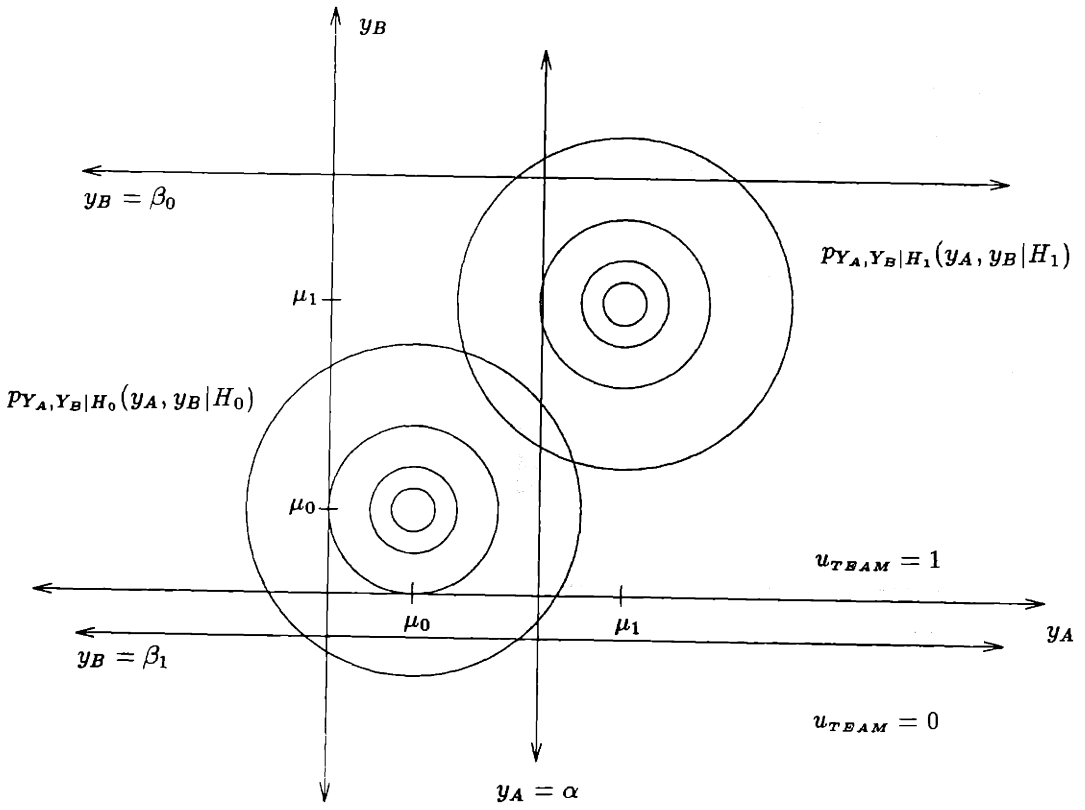


Figure 3-1: 2-Tand Decision Regions. Shading indicates (y_A, y_B) pairs for which the team decision $u_{Team} = u_B = 1$ is assigned.

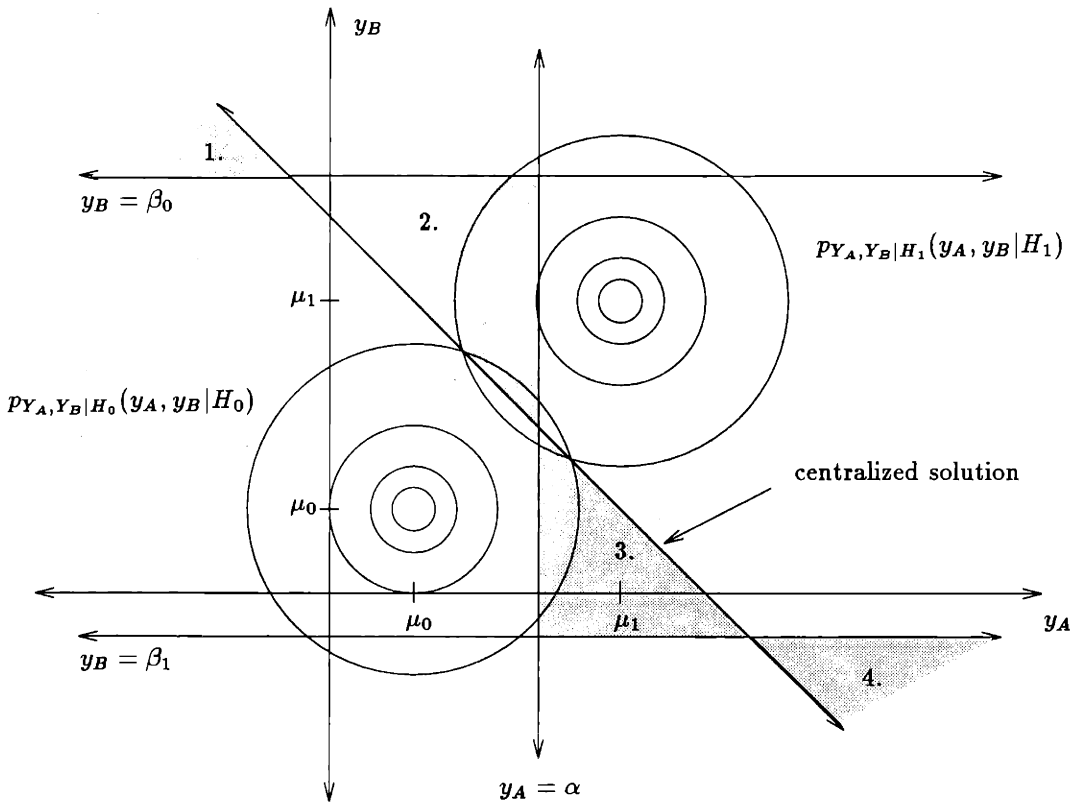


Figure 3-2: Typical placement of 2-Tand decision regions with respect to optimal centralized hyperplane. Shaded regions indicate regions of disagreement between 2-Tand and the centralized solution.

corresponding centralized solution. The centralized solution corresponds to a single hyperplane in y_A-y_B space. The shaded triangular areas $\triangle 1 - \triangle 4$ represent regions for which (y_A, y_B) pairs result in the 2-Tand network making a decision (team decision is u_B) which disagrees with the decision made by the centralized solution on those observation pairs. For (y_A, y_B) pairs in $\triangle 1$, $u_A = 0$, $u_B = 1$, and $u_{cent} = 0$. In $\triangle 2$, $u_A = 0$, $u_B = 0$,and $u_{cent} = 1$. In $\triangle 3$, $u_A = 1$, $u_B = 1$,and $u_{cent} = 0$. And in $\triangle 4$, $u_A = 1$, $u_B = 0$,and $u_{cent} = 1$.

A reasonable conjecture, upon inspection of Figure 3-2, would be that the problem of determining the optimal thresholds α^*, β_0^*, and β_1^* of 2-Tand is equivalent to the piecewise linear approximation problem in which the lines $y_A = \alpha$, $y_B = \beta_0$, and $y_B = \beta_1$ are placed so as to best fit the centralized solution, where the measure of fit is to minimize the probability of observations (y_A, y_B) falling into the four triangular regions of disagreement with the centralized. In fact, a simple argument indicates that this conjecture, for any DBHT team, is false, although it may still be of value to us. First, proof that the conjecture is false.

Proposition 3.1 (Piecewise Linear Approximation)
The probability of error of a DBHT Team is not minimized if the network observation thresholds are chosen to minimize probability of disagreement with the corresponding centralized solution.

Proof. Express the probability of error of the team as a function of the probability of error of the centralized solution

$$\begin{aligned} P_\epsilon^{Team} &= P_\epsilon^{Cent} + (P_\epsilon^{Team} - P_\epsilon^{Cent}) \\ &= P_\epsilon^{Cent} + (\Pr(\text{outcomes for which } u_{Team} \text{ incorrect, } u_{Cent} \text{ correct}) \\ &\quad - \Pr(\text{outcomes for which } u_{Team} \text{ correct, } u_{Cent} \text{ incorrect})) \end{aligned} \tag{3.5}$$

Then the optimal team thresholds $\underline{\theta}^*$ are given by

$$\underline{\theta}^* = \arg\min P_\epsilon^{Team}(\underline{\theta}) = \arg\min_{\underline{\theta}} (\Pr(\text{outcomes for which } u_{Team} \text{ incorrect, } u_{Cent} \text{ correct}) - \Pr(\text{outcomes for which } u_{Team} \text{ correct, } u_{Cent} \text{ incorrect})) \quad (3.6)$$

Clearly, minimizing the probability of disagreement with the centralized solution would minimize the quantity

$$(\Pr(\text{outcomes for which } u_{Team} \text{ incorrect, } u_{Cent} \text{ correct}) + \Pr(\text{outcomes for which } u_{Team} \text{ correct, } u_{Cent} \text{ incorrect})) \quad (3.7)$$

■

This argument can be made geometrically clear on the typical layout depicted in Figure 3-2. Label the point of intersection of $y_B = \beta_0$ with the centralized solution as $int.1$, the point of intersection of $y_A = \alpha$ with the centralized solution as $int.2$, and the point of intersection of $y_B = \beta_1$ with the centralized solution as $int.3$.

Since the areas of disagreement are triangular, it is not difficult to write out the corresponding probabilities of a data pair (y_A, y_B) falling into each of these four regions:

$$P(\triangle 1) = p_0[\int_{\beta_0}^{cent}\int_{-\infty}^{int.1} p_{Y_A,Y_B|H_0}(y_A, y_B|H_0)\, dy_A\, dy_B] + p_1[\int_{\beta_0}^{cent}\int_{-\infty}^{int.1} p_{Y_A,Y_B|H_1}(y_A, y_B|H_1)\, dy_A\, dy_B]$$

$$P(\triangle 2) = p_0[\int_{int.2}^{\beta_0}\int_{cent}^{\alpha} p_{Y_A,Y_B|H_0}(y_A, y_B|H_0)\, dy_A\, dy_B] + p_1[\int_{int.2}^{\beta 0}\int_{cent}^{\alpha} p_{Y_A,Y_B|H_1}(y_A, y_B|H_1)\, dy_A\, dy_B]$$

$$P(\triangle 3) = p_0[\int_{\beta_1}^{int.2}\int_{\alpha}^{cent} p_{Y_A,Y_B|H_0}(y_A, y_B|H_0)\, dy_A\, dy_B] + p_1[\int_{\beta_1}^{int.2}\int_{\alpha}^{cent} p_{Y_A,Y_B|H_1}(y_A, y_B|H_1)\, dy_A\, dy_B]$$

$$P(\triangle 4) = p_0[\int_{cent}^{\beta 1}\int_{int.3}^{+\infty} p_{Y_A,Y_B|H_0}(y_A, y_B|H_0)\, dy_A\, dy_B] + p_1[\int_{cent}^{\beta_1}\int_{int.3}^{+\infty} p_{Y_A,Y_B|H_1}(y_A, y_B|H_1)\, dy_A\, dy_B] \quad (3.8)$$

It can be seen that the probability in each of the four triangles is the sum of two terms. To simplify the notation, let us refer to the terms as $term1 - term8$ (taken in sequential order). Then after some manipulation

$$P_\epsilon^{2-Tand} = P_\epsilon^{Cent} + [(term1 + term4 + term5 + term8) - (term2 + term3 + term6 + term7)] \quad (3.9)$$

Terms 1,4,5,8 represent disagreement with the centralized solution in which the centralized solution was correct, and terms 2,3,6,7 represent disagreement with the centralized solution in which the centralized solution is incorrect. So actually, we have the following relationship:

$$(\alpha^*, \beta_0^*, \beta_1^*) = \arg\min P_\epsilon^{2-Tand}(\alpha, \beta_0, \beta_1) =$$
$$\arg\min_{\alpha,\beta_0,\beta_1} [(term1 + term4 + term5 + term8) - (term2 + term3 + term6 + term7)]$$

Clearly, minimizing disagreement with the centralized solution would actually minimize the *sum* of all eight terms.

Intuitively, minimizing probability of disagreement with the centralized solution will not result in minimum probability of error for the team since for some (y_A, y_B) pairs the team is correct and the centralized solution is incorrect, although the long run proportion of errors made by the centralized solution will certainly be lower.

Our interest in this conjecture was prompted by the question of whether or not the centralized solution could play the role of the teacher in the feedback loop. Rather than using measurements in which ground truth is provided, one might conceive of using measurements in which the centralized solution was provided instead. The previous argument indicates that no nice interpretation as an approximate line fitting problem to minimize disagreement would back up such a scheme. But it does raise the question of whether a bound on the relative deterioration might be established if

the thresholds of the DBHT team were in fact chosen to minimize the probability of disagreement. We leave this issue for future research.

In spite of the above discussion, numerical experiments on some cases of interest seem to indicate that the optimal positions of the threshold segments are generally close, both spatially and when measured by the probability enclosed in their difference, to the positions of the thresholds placed so as to minimize disagreement with the centralized. Thus, for many cases the bound referred to above may be small. This observation in turn provides some geometric intuition about some of the interesting behavior of 2-Tand described in Chapter 2. Gaussian simulation studies performed in [52] indicated that even for difficult problems[2] beyond 2 bits of communication capacity there is actually not much room for improvement. Some geometric justification for this is given in Figure 3-3 where it can be seen that allowing DM A more messages quickly reaches a point of diminishing return as a good fit is achieved with just a few segments. In Figures 3-4 and 3-5, the threshold spreading effect discussed in Section 2.5.1 is illustrated.

[2]hypotheses are close in a statistical sense, such as that measured by Kullback-Leibler distance for example

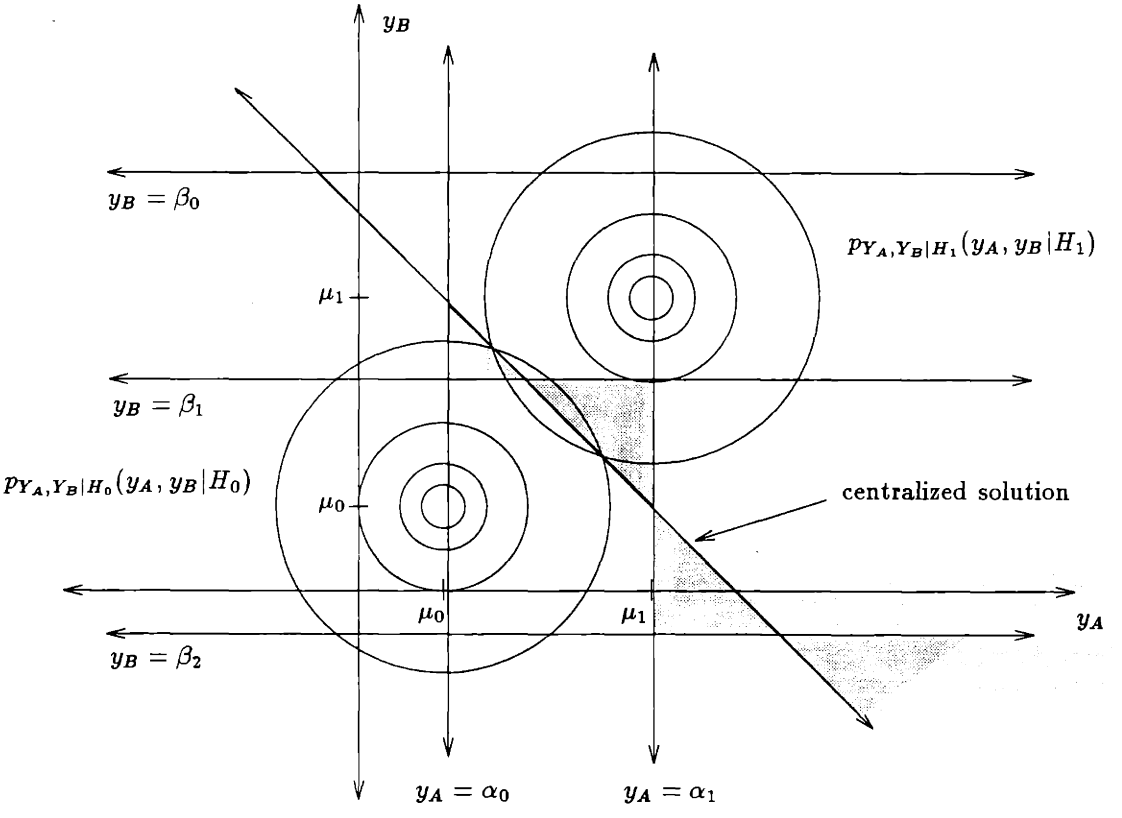


Figure 3-3: 2-Tand Observation Geometry: Multimessage case. The diagram indicates that good "approximations" to the centralized solution are obtained with very few messages.

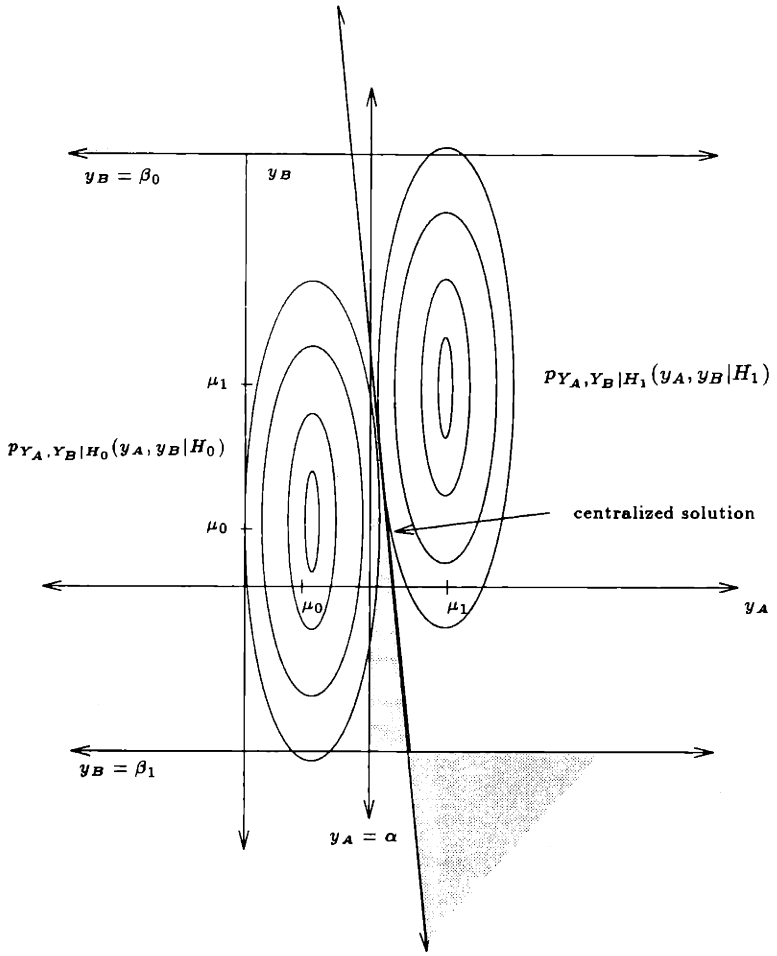


Figure 3-4: 2-Tand Observation Geometry: Spreading of thresholds when DM A smart (lower variance), DM B dumb (higher variance). As the disparity in variance increases, team performance approaches performance of A.

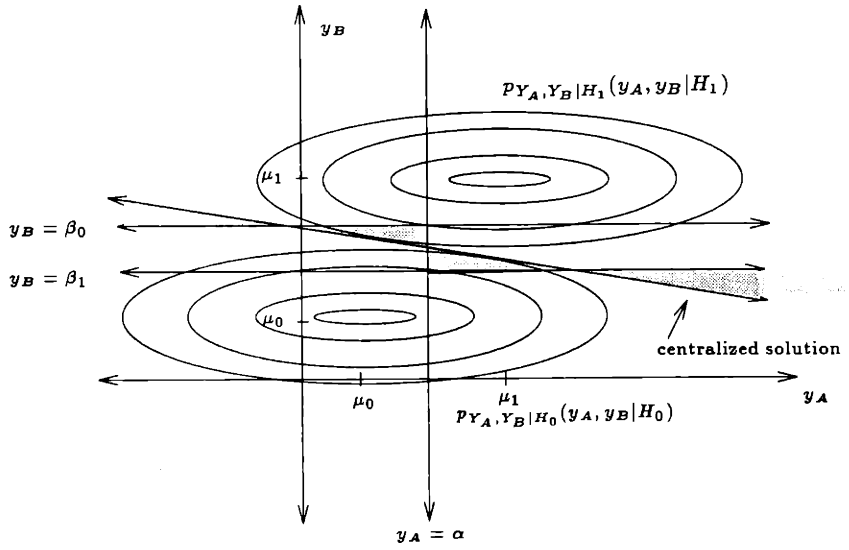


Figure 3-5: 2-Tand Observation Geometry: Closure of thresholds when DM A dumb (higher variance), DM B smart (lower variance). As disparity in variance increases, team performance approaches performance of B.

3.2.2 Sequential Probability Tree

Causality constraints dictate that decisions and communications in DBHT networks occur sequentially from those DMs of greatest distance from the primary DM, as measured in number of connecting arcs, back toward the primary DM, until the primary DM is reached and a team decision is made. Thus, there is an inherent feedforward nature to the decision process. Because of the tree structure of the topology, decisions of a particular DM affect only subsequent DMs, and the conditional independence of the observations at each DM makes the decision event at each stage, given the event of the previous stage, an independent event. The usefulness of these facts is that they allow for a complete sequential enumeration of the sample space on a binary probability tree. This is extremely useful as it provides a graphical depiction of the team decision process which makes clear the functional interdependence of the variables to be optimized, and provides a straightforward mechanism for generating parameterizations of the team error probability and the corresponding derivatives.

Example 1: 2-Tand

In order to illustrate, consider 2-Tand (Figure 2-7). The sample space for a single decision cycle of 2-Tand may be exhaustively enumerated in the form of a sequential probability tree as illustrated in Figure 3-6. This exercise makes directly available the functional form of the error probability, parameterized in terms of the *operating points* of each DM in the network. If we define a *path* in the enumeration tree as an event tuple (H, u_A, u_B), then we see that the probability of error may be read directly off the tree by summing the probability of those paths which terminate in errors. For Figure 3-6, if we sum the probability associated with the four erroneous paths $(0,0,1),(0,1,1),(1,0,0)$ and $(1,1,0)$ this yields

$$\begin{aligned} P_{\epsilon}^{2-Tand} &= p_0[(1-P_F^A)P_F^{B0} + P_F^A P_F^{B1}] \\ &\quad + p_1[(1-P_D^A)(1-P_D^{B0}) + P_D^A(1-P_D^{B1})] \end{aligned} \tag{3.10}$$

which we see is in agreement with (2.64). Furthermore, we immediately obtain the team probability of false alarm and miss by making the association

$$\begin{aligned} P_F^{2-Tand} &= [(1-P_F^A)P_F^{B0} + P_F^A P_F^{B1}] \\ P_M^{2-Tand} &= [(1-P_D^A)(1-P_D^{B0}) + P_D^A(1-P_D^{B1})] \end{aligned} \tag{3.11}$$

The team operating point is then given by $(P_F^{2-Tand}, P_D^{2-Tand})$ where $P_D^{2-Tand} = (1 - P_M^{2-Tand})$.

With the criterion function parameterized in this operating point form, we can formulate the problem of finding the minimum probability of error decision rules as one of two equivalent nonlinear optimization problems, one of which is constrained and the other unconstrained. The unconstrained optimization is parameterized in terms of the observation axis thresholds, while the constrained optimization is in terms of the operating points of each DM.

To formulate the unconstrained optimization, we rewrite (2.64) to make the de-

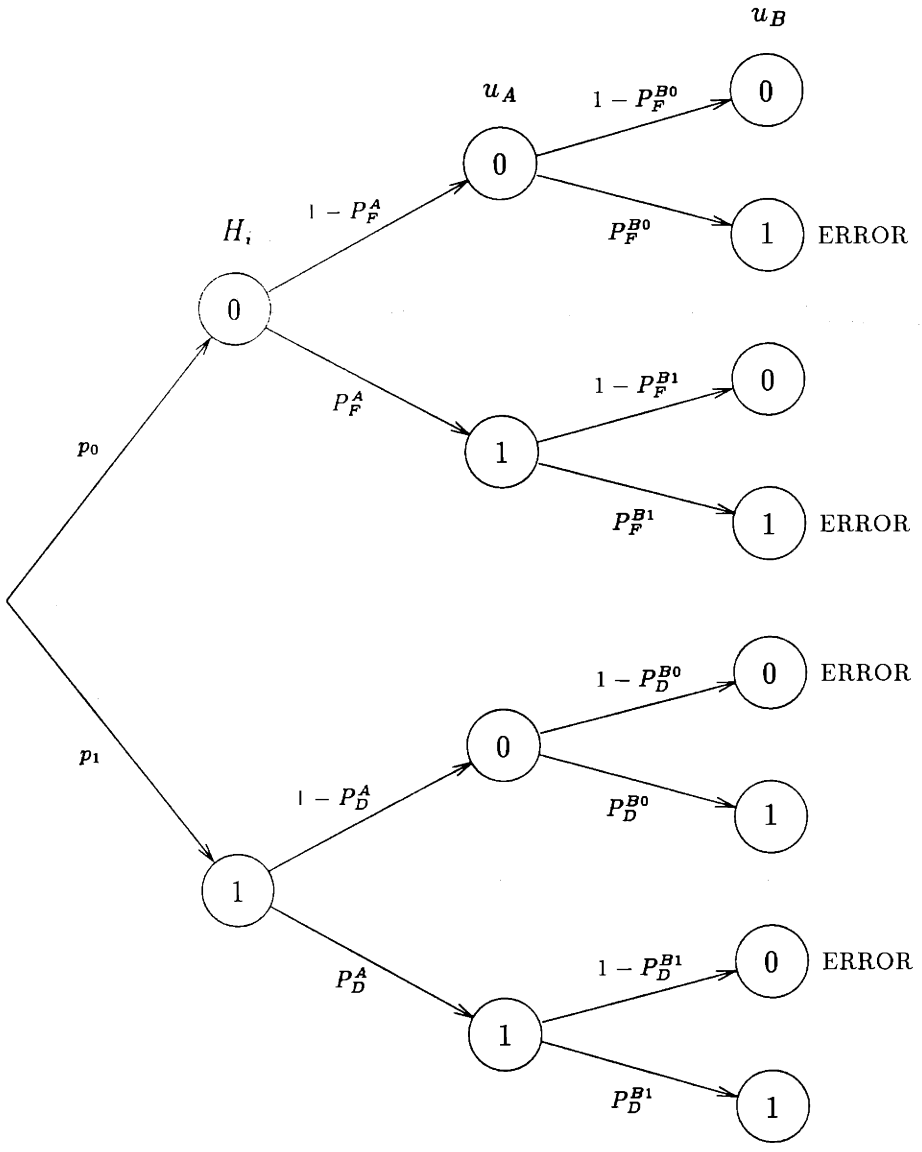


Figure 3-6: Sample Space for 2-Tand

pendence on the thresholds explicit as

$$\begin{aligned} P_\epsilon^{2-Tand}(\alpha,\beta_0,\beta_1) &= p_0\left[\left(1-\int_\alpha^{+\infty} p_{Y_A|H_0}(y_A|H_0)dy_A\right)\int_{\beta_0}^{+\infty} p_{Y_B|H_0}(y_B|H_0)dy_B\right. \\ &\quad \left.+\int_\alpha^{+\infty} p_{Y_A|H_0}(y_A|H_0)dy_A\int_{\beta_1}^{+\infty} p_{Y_B|H_0}(y_B|H_0)dy_B\right] \\ &\quad +p_1\left[\left(1-\int_\alpha^{+\infty} p_{Y_A|H_1}(y_A|H_1)dy_A\right)\left(1-\int_{\beta_0}^{+\infty} p_{Y_B|H_1}(y_B|H_1)dy_B\right)\right. \\ &\quad \left.+\int_\alpha^{+\infty} p_{Y_A|H_1}(y_A|H_1)dy_A\left(1-\int_{\beta_1}^{+\infty} p_{Y_B|H_1}(y_B|H_1)dy_B\right)\right] \end{aligned} \tag{3.12}$$

and then specify the problem of obtaining the optimal thresholds as

$$(\alpha^*,\beta_0^*,\beta_1^*) = \arg\min P_\epsilon^{2-Tand}(\alpha,\beta_0,\beta_1) \tag{3.13}$$

Here the conditional density functions are taken into account directly in the integrals.

The equivalent constrained problem may be formulated by taking the operating points of each DM as the variables of optimization, with the ROC curves as the constraint sets. In this approach, the conditional densities are taken into account indirectly (see Section 2.2.2). Assuming that the ROCs are provided as the constraint sets, (2.64) can be optimized over the choice of operating points (P_F^A, P_D^A), (P_F^{B0}, P_D^{B0}), and (P_F^{B1}, P_D^{B1}), where (P_F^A, P_D^A) is constrained to lie on ROC_A and (P_F^{B0}, P_D^{B0}), (P_F^{B1}, P_D^{B1}) are constrained to lie on ROC_B. Specifically, we perform

$$((P_F^{A*}, P_D^{A*}),(P_F^{B0*}, P_D^{B0*}),(P_F^{B1*}, P_D^{B1*})) =$$

$$\arg\min_{\substack{(P_F^A,P_D^A)\in \mathrm{ROC}_A \\ (P_F^{B0},P_D^{B0}),(P_F^{B1},P_D^{B1})\in \mathrm{ROC}_B}} \left\{ \begin{array}{l} p_0[(1-P_F^A)P_F^{B0}+P_F^A P_F^{B1}] \\ +p_1[(1-P_D^A)(1-P_D^{B0})+P_D^A(1-P_D^{B1})] \end{array} \right\}$$

Of course, this is nothing more than a restatement of the same problem, but this form may be more insightful from the training perspective. For example, from the point of view of human organizations, psychophysical experimental evidence [24] suggests that the ROC curve for a human executing a binary decision task may be effectively estimated point by point. If a good set of pointwise estimates of ROC_A and ROC_B can be obtained, the optimization of team performance can be reduced

to a straightforward combinatorial optimization over these finite feasible sets. The approach is limited only by the accuracy with which the ROCs can be estimated. In this scenario, it is not necessary to have explicit analytic expressions for the conditional densities. A truly nonparametric solution can be obtained. The only missing information required to minimize (3.10) is the prior probabilities, and these can also be estimated from the problem data. Nothing has been lost in this process since the requisite *coupling* between the DMs enters through the product terms in (3.10), the *prior probabilities* appear explicitly, and the *noise* enters through the constraining ROCs. Thus, a two step process is indicated in which the DMs first determine their capabilities in isolation (ROC curves), and then those DM's are assembled into an optimal team.

In a broader sense, viewing the problem in this way is intuitively pleasing. The structure of the P_ϵ cost function is a function only of the topology of the team. In other words, the way in which the DMs are interconnected determines the analytic form of P_ϵ. It has previously been argued that the ROCs can be thought of as characterizing the capability or expertise of the DMs. Thus, the problem of minimizing P_ϵ is equivalent to determining the point at which each DM should operate, within its capability, in order to be part of the optimal team. In general, the point(s) at which it must operate to be part of the optimal team will differ from where it would operate to be optimal in isolation.

This has implications for the problem of optimal topologies, although we do not address this topic in this report[3]. In principle, given a set of DMs and their corresponding ROCs, along with a specified topology, a system designer can locate every DM in every possible position, minimizing P_ϵ for each combination, and thereby establish where each DM should be placed within the given topology to best take advantage of its capabilities. Thus a DM can be optimally placed within the topology. Then the problem of determining a globally optimal configuration for a set of DMs can be phrased as a nested optimization in which the inner loop optimizes over the arrangement within a given topology, and the outer loop iterates over possible topolo-

[3]For more on the study of optimal topologies see Papastavrou [43], [44].

gies. Determining so-called dominant topologies, which are uniformly better than the alternatives for a given number of DMs has proven difficult, with most results being negative results in the form of counterexamples [43].

It is important to note that the usefulness of the sample space enumeration scheme is that it is completely general, in the sense that a such a tree may be generated for an *arbitrary* tree-structured organization with conditionally independent observations. In practice, its use is limited only by the resultant combinatorial explosion as the teams become larger. However, the recipe for generating the tree is straightforward and could easily be programmed. In this manner, the functional form of the probability of error, parameterized in terms of the operating points of each DM, can be readily found.

The partial derivatives of the cost with respect to each network observation threshold are readily obtained from (3.10) and (3.12) by application of Leibniz' rule and the chain rule as[4]

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \alpha} &= \frac{\partial P_\epsilon^{2-Tand}}{\partial P_F^A}\frac{dP_F^A}{d\alpha} + \frac{\partial P_\epsilon^{2-Tand}}{\partial P_D^A}\frac{dP_D^A}{d\alpha} \\
&= p_0\left[\int_{\beta_1}^{+\infty} p_{Y_B|H_0}(y_B|H_0)dy_B - \int_{\beta_0}^{+\infty} p_{Y_B|H_0}(y_B|H_0)dy_B\right]\frac{dP_F^A}{d\alpha} \\
&\quad + p_1\left[\int_{\beta_0}^{+\infty} p_{Y_B|H_1}(y_B|H_1)dy_B - \int_{\beta_1}^{+\infty} p_{Y_B|H_1}(y_B|H_1)dy_B\right]\frac{dP_D^A}{d\alpha} \\
&= p_0[P_F^{B1} - P_F^{B0}]\frac{dP_F^A}{d\alpha} + p_1[P_D^{B0} - P_D^{B1}]\frac{dP_D^A}{d\alpha} \\
&= -[P_F^{B1} - P_F^{B0}]p_0 p_{Y_A|H_0}(\alpha|H_0) \\
&\quad + [P_D^{B1} - P_D^{B0}]p_1 p_{Y_A|H_1}(\alpha|H_1) \qquad (3.14)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_0} &= \frac{\partial P_\epsilon^{2-Tand}}{\partial P_F^{B0}}\frac{dP_F^{B0}}{d\beta_0} + \frac{\partial P_\epsilon^{2-Tand}}{\partial P_D^{B0}}\frac{dP_D^{B0}}{d\beta_0} \\
&= p_0\left[1 - \int_{\alpha}^{+\infty} p_{Y_A|H_0}(y_A|H_0)dy_A\right]\frac{dP_F^{B0}}{d\beta_0} \\
&\quad - p_1\left[1 - \int_{\alpha}^{+\infty} p_{Y_A|H_1}(y_A|H_1)dy_A\right]\frac{dP_D^{B0}}{d\beta_0}
\end{aligned}$$

[4]Note: Setting these partial derivatives to zero gives the necessary conditions for optimality in terms of satisfying the LRTs with equality

$$
\begin{aligned}
&= p_0[1 - P_F^A]\frac{dP_F^{B0}}{d\beta_0} + p_1[P_D^A - 1]\frac{dP_D^{B0}}{d\beta_0} \\
&= -[1 - P_F^A]p_0 p_{Y_B|H_0}(\beta_0|H_0) \\
&\quad +[1 - P_D^A]p_1 p_{Y_B|H_1}(\beta_0|H_1)
\end{aligned} \tag{3.15}
$$

$$
\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_1} &= \frac{\partial P_\epsilon^{2-Tand}}{\partial P_F^{B1}}\frac{dP_F^{B1}}{d\beta_1} + \frac{\partial P_\epsilon^{2-Tand}}{\partial P_D^{B1}}\frac{dP_D^{B1}}{d\beta_1} \\
&= p_0 \int_\alpha^{+\infty} p_{Y_A|H_0}(y_A|H_0)dy_A \frac{dP_F^{B1}}{d\beta_1} \\
&\quad -p_1 \int_\alpha^{+\infty} p_{Y_A|H_1}(y_A|H_1)dy_A \frac{dP_D^{B1}}{d\beta_1} \\
&= p_0 P_F^A \frac{dP_F^{B1}}{d\beta_1} - p_1 P_D^A \frac{dP_D^{B1}}{d\beta_1} \\
&= -P_F^A p_0 p_{Y_B|H_0}(\beta_1|H_0) + P_D^A p_1 p_{Y_B|H_1}(\beta_1|H_1)
\end{aligned} \tag{3.16}
$$

Note that each partial derivative is comprised of terms which can be considered "individual" and terms which can be considered "organizational". For example, we can identify the terms in (3.14) as

$$
\frac{\partial P_\epsilon}{\partial \alpha} = \overbrace{\frac{\partial P_\epsilon}{\partial P_F^A}}^{org}\overbrace{\frac{dP_F^A}{d\alpha}}^{ind} + \overbrace{\frac{\partial P_\epsilon}{\partial P_D^A}}^{org}\overbrace{\frac{dP_D^A}{d\alpha}}^{ind} \tag{3.17}
$$

since the terms identified as organizational depend only on the prior probabilities, the organization's architecture and current state (in this case current performance of DM B) while the individual terms depend only on how DM A's own performance varies with its threshold. This of course is a function of the noise environment or ROC curve of A alone. [comment on coupling probabilities]

Note also that the derivatives demonstrate clearly the coupling discussed previously. In order for DM A to compute its partial derivative, it must have the conditional operating points (P_F^{B0}, P_D^{B0}) and (P_F^{B1}, P_D^{B1}) of DM B. And DM B cannot compute its partial derivatives without the operating point (P_F^A, P_D^A) of DM A.

The following sections illustrate the trees for the other small teams we have considered. There are several reasons for presenting this material. In the first place, we

wish to add credence to our claim that such a tree may be generated for an arbitrary tree-structured network with conditionally independent observations, as we will rely on this fact to argue that certain structural properties of the cost and derivatives hold true in general. We hope that this structure will be readily observable in these examples. Secondly, the specific forms of the cost functions and derivatives we derive here are required in order to implement some of the stochastic gradient-based training algorithms of Chapter 5 on these networks. Finally, the examples serve to make clear the source of the added complexity in the decision rules presented in Chapter 2.

Example 2: 3-Vee

For the 3-Vee network (Figure 2-12) we obtain the tree shown in Figure 3-7.

In this case, paths are event tuples of the form $(H,(u_A,u_B),u_C)$, where the inner parenthesis indicates that the parallel decisions at A and B are generated simultaneously. In similar fashion, we may obtain directly from this tree the following operating point parameterization of the probability of error

$$\begin{aligned} P_\epsilon^{3-Vee} &= p_0[(1-P_F^A)((1-P_F^B)P_F^{C(00)}+P_F^BP_F^{C(01)}) \\ &\quad +P_F^A((1-P_F^B)P_F^{C(10)}+P_F^BP_F^{C(11)})] \\ &\quad +p_1[(1-P_D^A)((1-P_D^B)(1-P_D^{C(00)})+P_D^B(1-P_D^{C(01)})) \\ &\quad +P_D^A((1-P_D^B)(1-P_D^{C(10)})+P_D^B(1-P_D^{C(11)}))] \end{aligned} \tag{3.18}$$

from which the team probabilities of false alarm and miss are expressible as

$$\begin{aligned} P_F^{3-Vee} &= [(1-P_F^A)((1-P_F^B)P_F^{C(00)}+P_F^BP_F^{C(01)}) \\ &\quad +P_F^A((1-P_F^B)P_F^{C(10)}+P_F^BP_F^{C(11)})] \end{aligned}$$

$$\begin{aligned} P_M^{3-Vee} &= [(1-P_D^A)((1-P_D^B)(1-P_D^{C(00)})+P_D^B(1-P_D^{C(01)})) \\ &\quad +P_D^A((1-P_D^B)(1-P_D^{C(10)})+P_D^B(1-P_D^{C(11)}))] \end{aligned} \tag{3.19}$$

and the team operating point given by $(P_F^{3-Vee},P_D^{3-Vee})$ where $P_D^{3-Vee}=(1-P_M^{3-Vee})$.

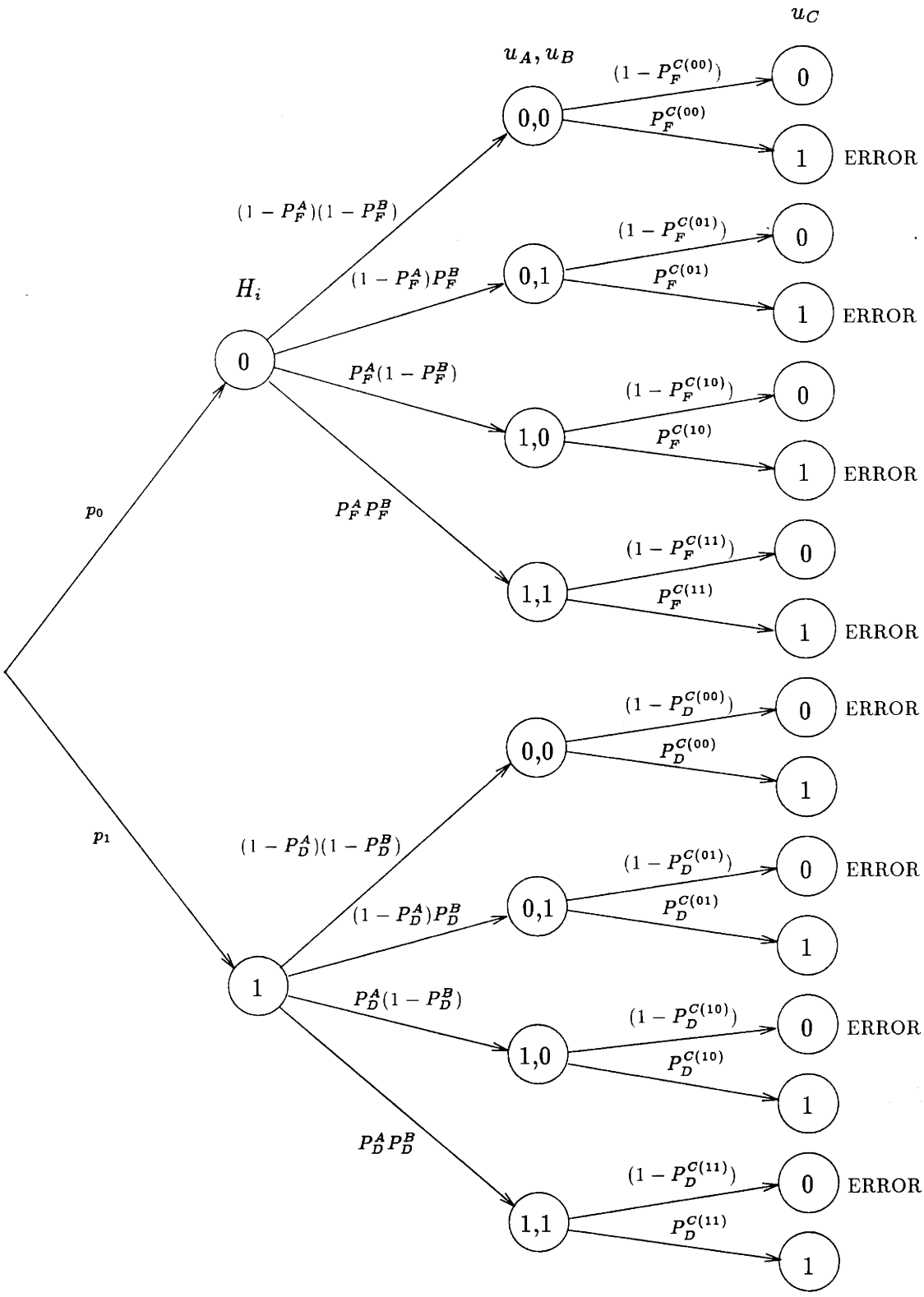


Figure 3-7: Sample Space for 3-Vee

The partial derivatives for the linear threshold parameterization are readily computed from (3.18) as

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Vee}}{\partial \alpha} &= -[(1-P_F^B)(P_F^{C(10)} - P_F^{C(00)}) \\
&\quad + P_F^B(P_F^{C(11)} - P_F^{C(01)})]p_0 p_{Y_A|H_0}(\alpha|H_0) \\
&\quad + [(1-P_D^B)(P_D^{C(10)} - P_D^{C(00)}) \\
&\quad + P_D^B(P_D^{C(11)} - P_D^{C(01)})]p_1 p_{Y_A|H_1}(\alpha|H_1) \qquad (3.20)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Vee}}{\partial \beta} &= -[(1-P_F^A)(P_F^{C(01)} - P_F^{C(00)}) \\
&\quad + P_F^A(P_F^{C(11)} - P_F^{C(10)})]p_0 p_{Y_B|H_0}(\beta|H_0) \\
&\quad + [(1-P_D^A)(P_D^{C(01)} - P_D^{C(00)}) \\
&\quad + P_D^A(P_D^{C(11)} - P_D^{C(10)})]p_1 p_{Y_B|H_1}(\beta|H_1) \qquad (3.21)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Vee}}{\partial \xi_{00}} &= -[(1-P_F^A)(1-P_F^B)]p_0 p_{Y_C|H_0}(\xi_{00}|H_0) \\
&\quad + [(1-P_D^A)(1-P_D^B)]p_1 p_{Y_C|H_1}(\xi_{00}|H_1) \qquad (3.22)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Vee}}{\partial \xi_{01}} &= -[(1-P_F^A)P_F^B]p_0 p_{Y_C|H_0}(\xi_{01}|H_0) \\
&\quad + [(1-P_D^A)P_D^B]p_1 p_{Y_C|H_1}(\xi_{01}|H_1) \qquad (3.23)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Vee}}{\partial \xi_{10}} &= -[P_F^A(1-P_F^B)]p_0 p_{Y_C|H_0}(\xi_{10}|H_0) \\
&\quad + [P_D^A(1-P_D^B)]p_1 p_{Y_C|H_1}(\xi_{10}|H_1) \qquad (3.24)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Vee}}{\partial \xi_{11}} &= -[P_F^A P_F^B]p_0 p_{Y_C|H_0}(\xi_{11}|H_0) \\
&\quad + [P_D^A P_D^B]p_1 p_{Y_C|H_1}(\xi_{11}|H_1) \qquad (3.25) \\
& \qquad (3.26)
\end{aligned}$$

Example 3: 3-Tand

For the 3-Tand network (Figure 2-13) we obtain the tree shown in Figure 3-8.

Event tuples in this tree are of the form (H, u_A, u_B, u_C). From the tree we read

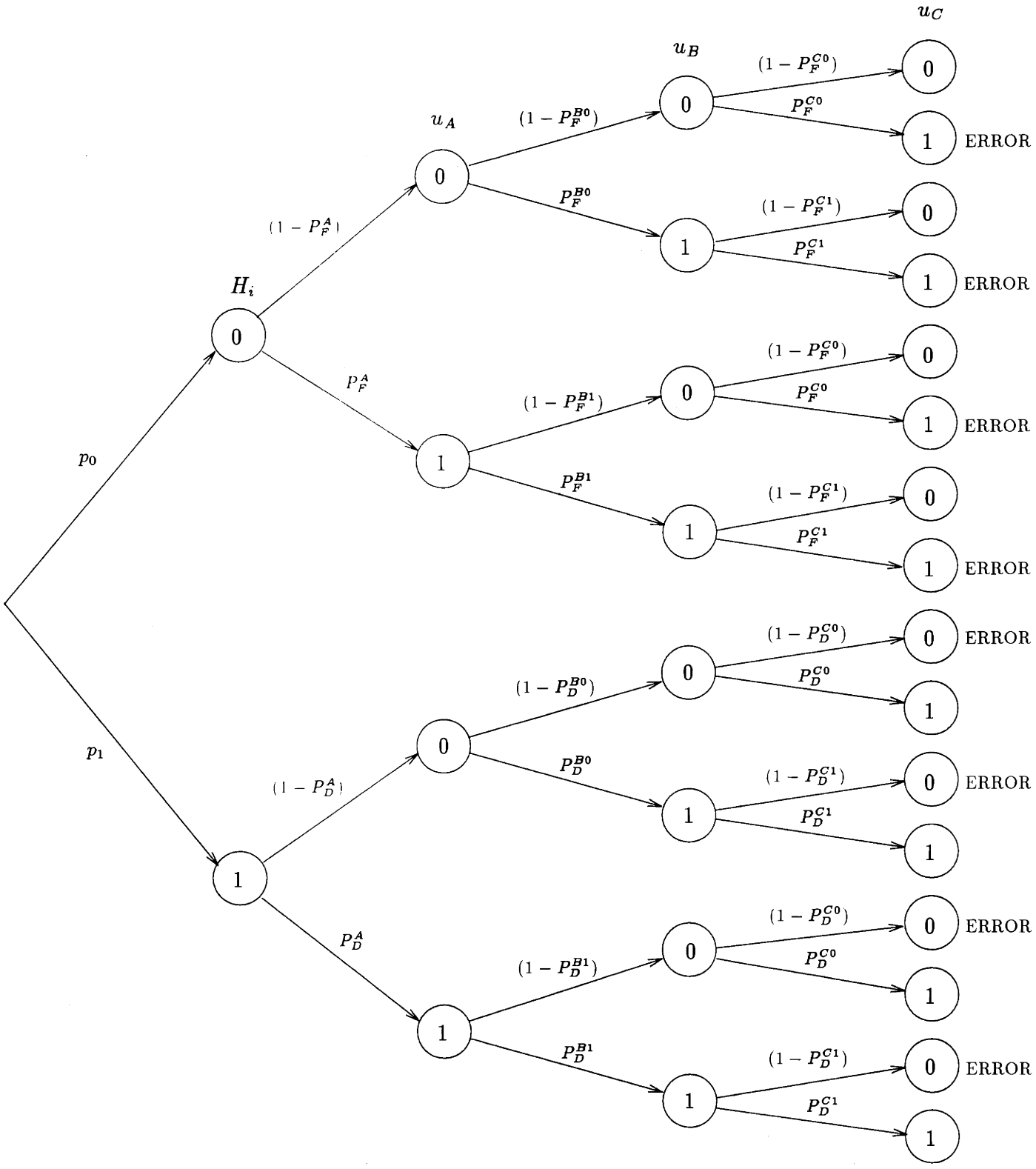


Figure 3-8: Sample Space for 3-Tand

off the operating point parameterization of the probability of error as

$$\begin{aligned} P_\epsilon^{3-Tand} &= p_0[(1-P_F^A)((1-P_F^{B0})P_F^{C0}+P_F^{B0}P_F^{C1}) \\ &\quad +P_F^A((1-P_F^{B1})P_F^{C0}+P_F^{B1}P_F^{C1})] \\ &\quad +p_1[(1-P_D^A)((1-P_D^{B0})(1-P_D^{C0})+P_D^{B0}(1-P_D^{C1})) \\ &\quad +P_D^A((1-P_D^{B1})(1-P_D^{C0})+P_D^{B1}(1-P_D^{C1}))] \end{aligned} \tag{3.27}$$

from which the team probabilities of false alarm and miss are given by

$$\begin{aligned} P_F^{3-Tand} &= [(1-P_F^A)((1-P_F^{B0})P_F^{C0}+P_F^{B0}P_F^{C1}) \\ &\quad +P_F^A((1-P_F^{B1})P_F^{C0}+P_F^{B1}P_F^{C1})] \end{aligned}$$

$$\begin{aligned} P_M^{3-Vee} &= [(1-P_D^A)((1-P_D^{B0})(1-P_D^{C0})+P_D^{B0}(1-P_D^{C1})) \\ &\quad +P_D^A((1-P_D^{B1})(1-P_D^{C0})+P_D^{B1}(1-P_D^{C1}))] \end{aligned} \tag{3.28}$$

and the team operating point is given by $(P_F^{3-Tand}, P_D^{3-Tand})$ where $P_D^{3-Tand} = (1 - P_M^{3-Tand})$.

The partial derivatives for the linear threshold parameterization are readily computed from (3.18) as

$$\begin{aligned} \frac{\partial P_\epsilon^{3-Tand}}{\partial \alpha} &= -[-((1-P_F^{B0})P_F^{C0}+P_F^{B0}P_F^{C1}) \\ &\quad +((1-P_F^{B1})P_F^{C0}+P_F^{B1}P_F^{C1})]p_0 p_{Y_A|H_0}(\alpha|H_0) \\ &\quad +[((1-P_D^{B0})(1-P_D^{C0})+P_D^{B0}(1-P_D^{C1})) \\ &\quad -((1-P_D^{B1})(1-P_D^{C0})+P_D^{B1}(1-P_D^{C1}))]p_1 p_{Y_A|H_1}(\alpha|H_1) \\ &= -[(P_F^{B1}-P_F^{B0})(P_F^{C1}-P_F^{C0})]p_0 p_{Y_A|H_0}(\alpha|H_0) \\ &\quad +[(P_D^{B1}-P_D^{B0})(P_D^{C1}-P_D^{C0})]p_1 p_{Y_A|H_1}(\alpha|H_1) \end{aligned} \tag{3.29}$$

$$\begin{aligned} \frac{\partial P_\epsilon^{3-Tand}}{\partial \beta_0} &= -[(1-P_F^A)(P_F^{C1}-P_F^{C0})]p_0 p_{Y_B|H_0}(\beta_0|H_0) \\ &\quad +[(1-P_D^A)((1-P_D^{C0})-(1-P_D^{C1}))]p_1 p_{Y_B|H_1}(\beta_0|H_1) \\ &= -[(1-P_F^A)(P_F^{C1}-P_F^{C0})]p_0 p_{Y_B|H_0}(\beta_0|H_0) \end{aligned}$$

$$+[(1-P_D^A)(P_D^{C1}-P_D^{C0})]p_1 p_{Y_B|H_1}(\beta_0|H_1) \tag{3.30}$$

$$\begin{aligned}\frac{\partial P_\epsilon^{3-Tand}}{\partial \beta_1} &= -[P_F^A(P_F^{C1}-P_F^{C0})]p_0 p_{Y_B|H_0}(\beta_1|H_0) \\ &\quad +[P_D^A((1-P_D^{C0})-(1-P_D^{C1}))]p_1 p_{Y_B|H_1}(\beta_1|H_1) \\ &= -[P_F^A(P_F^{C1}-P_F^{C0})]p_0 p_{Y_B|H_0}(\beta_1|H_0) \\ &\quad +[P_D^A(P_D^{C1}-P_D^{C0})]p_1 p_{Y_B|H_1}(\beta_1|H_1)\end{aligned} \tag{3.31}$$

$$\begin{aligned}\frac{\partial P_\epsilon^{3-Tand}}{\partial \xi_0} &= -[(1-P_F^A)(1-P_F^{B0})+P_F^A(1-P_F^{B1})]p_0 p_{Y_C|H_0}(\xi_0|H_0) \\ &\quad +[(1-P_D^A)(1-P_D^{B0})+P_D^A(1-P_D^{B1})]p_1 p_{Y_C|H_1}(\xi_0|H_1)\end{aligned} \tag{3.32}$$

$$\begin{aligned}\frac{\partial P_\epsilon^{3-Tand}}{\partial \xi_1} &= -[(1-P_F^A)P_F^{B0}+P_F^A P_F^{B1}]p_0 p_{Y_C|H_0}(\xi_1|H_0) \\ &\quad +[(1-P_D^A)P_D^{B0}+P_D^A P_D^{B1}]p_1 p_{Y_C|H_1}(\xi_1|H_1)\end{aligned} \tag{3.33}$$

Example 4: 4-Asym

For the 4-Asym network (Figure 2-14) we obtain the tree shown in Figure 3-9.

Event tuples in this tree are representable as $(H, u_A, (u_B, u_C), u_D)$, where again the inner parenthesis indicates simultaneous parallel events. Similarly, we may obtain directly from this tree the following operating point parameterization of the probability of error

$$\begin{aligned}P_\epsilon^{4-Asym} &= p_0[\,(1-P_F^A)\,((1-P_F^{B0})(1-P_F^C)P_F^{D(00)}+(1-P_F^{B0})P_F^C P_F^{D(01)} \\ &\quad +P_F^{B0}(1-P_F^C)P_F^{D(10)}+P_F^{B0}P_F^C P_F^{D(11)}\,) \\ &\quad +P_F^A\,(\,(1-P_F^{B1})(1-P_F^C)P_F^{D(00)}+(1-P_F^{B1})P_F^C P_F^{D(01)} \\ &\quad +P_F^{B1}(1-P_F^C)P_F^{D(10)}+P_F^{B1}P_F^C P_F^{D(11)})\,] \\ &\quad +p_1[\,(1-P_D^A)\,((1-P_D^{B0})(1-P_D^C)(1-P_D^{D(00)})+(1-P_D^{B0})P_D^C(1-P_D^{D(01)}) \\ &\quad +P_D^{B0}(1-P_D^C)(1-P_D^{D(10)})+P_D^{B0}P_D^C(1-P_D^{D(11)})) \\ &\quad +P_D^A\,((1-P_D^{B1})(1-P_D^C)(1-P_D^{D(00)})+(1-P_D^{B1})P_D^C(1-P_D^{D(01)}) \\ &\quad +P_D^{B1}(1-P_D^C)(1-P_D^{D(10)})+P_D^{B1}P_D^C(1-P_D^{D(11)}))\,]\end{aligned} \tag{3.34}$$

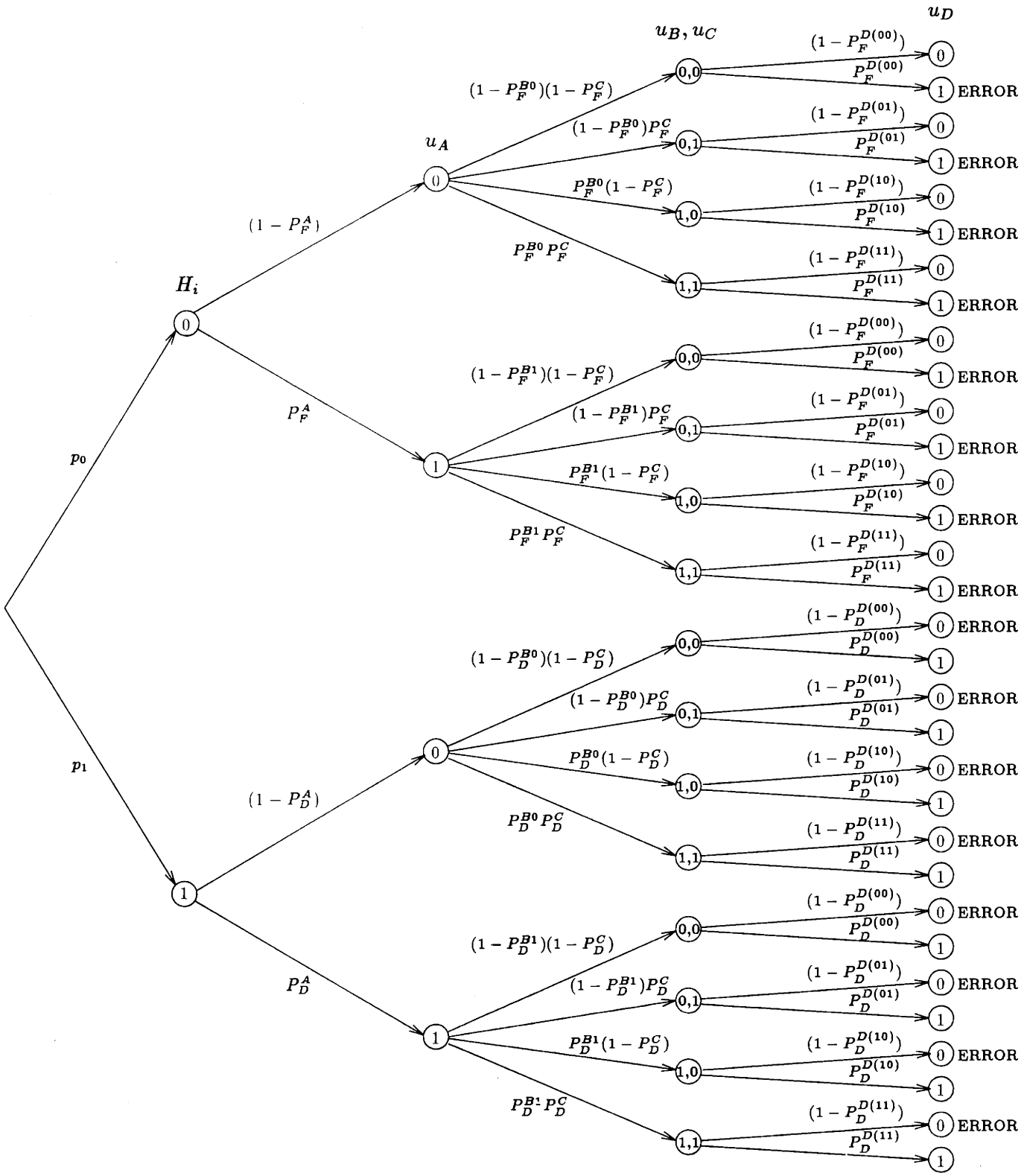


Figure 3-9: Sample Space for 4-Asym

from which the team probabilities of false alarm and miss are given by

$$\begin{aligned}
P_F^{4-asym} &= [\,(1-P_F^A)((1-P_F^{B0})(1-P_F^C)P_F^{D(00)} + (1-P_F^{B0})P_F^C P_F^{D(01)} \\
&\quad +P_F^{B0}(1-P_F^C)P_F^{D(10)} + P_F^{B0}P_F^C P_F^{D(11)}) \\
&\quad +P_F^A((1-P_F^{B1})(1-P_F^C)P_F^{D(00)} + (1-P_F^{B1})P_F^C P_F^{D(01)} \\
&\quad +P_F^{B1}(1-P_F^C)P_F^{D(10)} + P_F^{B1}P_F^C P_F^{D(11)})\,] \\
P_M^{4-Asym} &= [\,(1-P_D^A)((1-P_D^{B0})(1-P_D^C)(1-P_D^{D(00)}) + (1-P_D^{B0})P_D^C(1-P_D^{D(01)}) \\
&\quad +P_D^{B0}(1-P_D^C)(1-P_D^{D(10)}) + P_D^{B0}P_D^C(1-P_D^{D(11)})) \\
&\quad +P_D^A((1-P_D^{B1})(1-P_D^C)(1-P_D^{D(00)}) + (1-P_D^{B1})P_D^C(1-P_D^{D(01)}) \\
&\quad +P_D^{B1}(1-P_D^C)(1-P_D^{D(10)}) + P_D^{B1}P_D^C(1-P_D^{D(11)}))\,] \qquad (3.35)
\end{aligned}$$

The partial derivatives for the linear threshold parameterization are readily computed from (3.34) as

$$\begin{aligned}
\frac{\partial P_\epsilon^{4-Asym}}{\partial \alpha} &= -[\,-((1-P_F^{B0})(1-P_F^C)P_F^{D(00)} + (1-P_F^{B0})P_F^C P_F^{D(01)} \\
&\quad +P_F^{B0}(1-P_F^C)P_F^{D(10)} + P_F^{B0}P_F^C P_F^{D(11)}) \\
&\quad +((1-P_F^{B1})(1-P_F^C)P_F^{D(00)} + (1-P_F^{B1})P_F^C P_F^{D(01)} \\
&\quad +P_F^{B1}(1-P_F^C)P_F^{D(10)} + P_F^{B1}P_F^C P_F^{D(11)})\,] \\
&\quad p_0 p_{Y_A|H_0}(\alpha|H_0) \\
&\quad +[\,((1-P_D^{B0})(1-P_D^C)(1-P_D^{D(00)}) + (1-P_D^{B0})P_D^C(1-P_D^{D(01)}) \\
&\quad +P_D^{B0}(1-P_D^C)(1-P_D^{D(10)}) + P_D^{B0}P_D^C(1-P_D^{D(11)})) \\
&\quad -((1-P_D^{B1})(1-P_D^C)(1-P_D^{D(00)}) + (1-P_D^{B1})P_D^C(1-P_D^{D(01)}) \\
&\quad +P_D^{B1}(1-P_D^C)(1-P_D^{D(10)}) + P_D^{B1}P_D^C(1-P_D^{D(11)}))\,] \\
&\quad p_1 p_{Y_A|H_1}(\alpha|H_1) \qquad (3.36)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{4-Asym}}{\partial \beta_0} &= -[\,(1-P_F^A)(-(1-P_F^C)P_F^{D(00)} - P_F^C P_F^{D(01)} \\
&\quad +(1-P_F^C)P_F^{D(10)} + P_F^C P_F^{D(11)})\,]p_0 p_{Y_B|H_0}(\beta_0|H_0)
\end{aligned}$$

$$
\begin{aligned}
&+[\,(1-P_D^A)((1-P_D^C)(1-P_D^{D(00)})+P_D^C(1-P_D^{D(01)}) \\
&-(1-P_D^C)(1-P_D^{D(10)})-P_D^C(1-P_D^{D(11)}))\,]p_1 p_{Y_B|H_1}(\beta_0|H_1) \quad (3.37)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial P_\epsilon^{4-Asym}}{\partial \beta_1} &= -[\,P_F^A(-(1-P_F^C)P_F^{D(00)}-P_F^C P_F^{D(01)} \\
&+(1-P_F^C)P_F^{D(10)}+P_F^C P_F^{D(11)})\,]p_0 p_{Y_B|H_0}(\beta_1|H_0) \\
&+[\,P_D^A((1-P_D^C)(1-P_D^{D(00)})+P_D^C(1-P_D^{D(01)}) \\
&-(1-P_D^C)(1-P_D^{D(10)})-P_D^C(1-P_D^{D(11)}))\,]p_1 p_{Y_B|H_1}(\beta_1|H_1) \quad (3.38)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial P_\epsilon^{4-Asym}}{\partial \xi} &= -[\,(1-P_F^A)(-(1-P_F^{B0})P_F^{D(00)}+(1-P_F^{B0})P_F^{D(01)} \\
&-P_F^{B0}P_F^{D(10)}+P_F^{B0}P_F^{D(11)}) \\
&+P_F^A(-(1-P_F^{B1})P_F^{D(00)}+(1-P_F^{B1})P_F^{D(01)} \\
&-P_F^{B1}P_F^{D(10)}+P_F^{B1}P_F^{D(11)})\,]p_0 p_{Y_C|H_0}(\xi|H_0) \\
&+[\,(1-P_D^A)((1-P_D^{B0})(1-P_D^{D(00)})-(1-P_D^{B0})(1-P_D^{D(01)}) \\
&+P_D^{B0}(1-P_D^{D(10)})-P_D^{B0}(1-P_D^{D(11)})) \\
&+P_D^A((1-P_D^{B1})(1-P_D^{D(00)})-(1-P_D^{B1})(1-P_D^{D(01)}) \\
&+P_D^{B1}(1-P_D^{D(10)})-P_D^{B1}(1-P_D^{D(11)}))\,]p_1 p_{Y_C|H_1}(\xi|H_1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial P_\epsilon^{4-Asym}}{\partial \zeta_{00}} &= -[\,(1-P_F^A)(1-P_F^{B0})(1-P_F^C) \\
&+P_F^A(1-P_F^{B1})(1-P_F^C)\,]p_0 p_{Y_D|H_0}(\zeta_{00}|H_0) \\
&[\,(1-P_D^A)(1-P_D^{B0})(1-P_D^C) \\
&+P_D^A(1-P_D^{B1})(1-P_D^C)\,]p_1 p_{Y_D|H_1}(\zeta_{00}|H_1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial P_\epsilon^{4-Asym}}{\partial \zeta_{01}} &= -[\,(1-P_F^A)(1-P_F^{B0})P_F^C \\
&+P_F^A(1-P_F^{B1})P_F^C\,]p_0 p_{Y_D|H_0}(\zeta_{01}|H_0) \\
&[\,(1-P_D^A)(1-P_D^{B0})P_D^C \\
&+P_D^A(1-P_D^{B1})P_D^C\,]p_1 p_{Y_D|H_1}(\zeta_{01}|H_1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial P_{\epsilon}^{4-Asym}}{\partial \zeta_{10}} &= -[\,(1-P_F^A)P_F^{B0}(1-P_F^C) \\
&\quad +P_F^A P_F^{B1}(1-P_F^C)\,]p_0 p_{Y_D|H_0}(\zeta_{10}|H_0) \\
&\quad [\,(1-P_D^A)P_D^{B0}(1-P_D^C) \\
&\quad +P_D^A P_D^{B1}(1-P_D^C)\,]p_1 p_{Y_D|H_1}(\zeta_{10}|H_1)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial P_{\epsilon}^{4-Asym}}{\partial \zeta_{11}} &= -[\,(1-P_F^A)P_F^{B0}P_F^C \\
&\quad +P_F^A P_F^{B1}P_F^C\,]p_0 p_{Y_D|H_0}(\zeta_{11}|H_0) \\
&\quad [\,(1-P_D^A)P_D^{B0}P_D^C \\
&\quad +P_D^A P_D^{B1}P_D^C\,]p_1 p_{Y_D|H_1}(\zeta_{11}|H_1)
\end{aligned} \tag{3.39}
$$

3.2.3 Deterministic Optimal Control Formulation

The optimization of the decision rules may be formulated as a deterministic, finite-horizon, nonlinear, discrete time optimal control problem. This fact was, to our knowledge, first suggested by Ekchian in [19] en route to developing the "sweep algorithm" for numerically computing the optimal decision rules. However, it was also independently derived in an alternate form by Tang in [59], and is discussed at length in [61], [62].

In this formulation, the discrete time index corresponds to the stage variable, with the final stage corresponding to the last DM in the information pathway, i.e., the primary DM. The problem becomes a spatial optimal control problem, with terminal cost on the performance of this DM. It is interesting that what initially appeared to be a *stochastic* optimal control problem is in fact reducible to an equivalent *deterministic* problem by selecting the appropriate notion of state, in particular the aggregate probabilities of false alarm and detection of each DM. This choice of state is sufficient to capture the relevant statistical properties of the output stream of each DM.

The value of this formulation, from the point of view of this report, is twofold. In the first place, it demonstrates how to take advantage of the structure of the problem to make it amenable to solution by algorithms which are well-suited to solving optimal control problems. These algorithms may be more efficient, particularly in terms of communication requirements in distributed optimization algorithms. Along these lines, we will show in Chapter 5 how the optimal control formulation described here may be exploited to give a "back propagation" version of a gradient descent approach for the network optimization problem. Application of optimal control ideas will permit each DM to compute the derivative of the team probability of error with respect to its local thresholds with only communication from its immediate predecessors and successors. This proves useful, particularly in large problems since, as was made evident by the development in the previous sections, this derivative generally depends on *all* of the other operating points in the network. A second value of the optimal control formulation is that, by viewing the problem in this way, quantities such as cost-to-go are made available which allow interesting interpretations to be

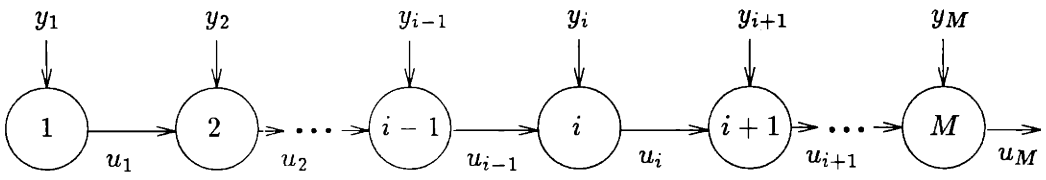


Figure 3-10: General M DM Tandem Network

made.

Our development here follows the approach in [59], [61] with minor modification to use the observation thresholds as the control variables, rather than the conditional operating points. The material we present here will be used later in the development of a back-propagation training algorithm. It is sufficient for our purposes to summarize the formulation for tandem networks only. For this case the formulation is also the most transparent. The generalization to arbitrary tree-structured networks is very similar, and is discussed in detail in [59], [62].

Tandem topologies such as 2-Tand and 3-Tand have the general form shown in Figure 3-10, where we have used numerical rather than letter labels in order to more easily express recursion.

The primary DM is DM M. Recall that for tandem configurations, DM 1 uses a single threshold, which we denote θ_1, while the other DMs $i = 2, \ldots, M$ each use two, which we denote $\theta_{i/0}, \theta_{i/1}$, respectively. We use the notation $\underline{\theta}$ to denote the vector of all network threshold parameters. We denote the aggregate probability of false alarm of DM i as P_F^i, and the two conditional probabilities of false alarm as $P_F^{i/0}, P_F^{i/1}$.

We define the main components of the corresponding optimal control formulation as follows.

cost: The cost to be optimized is the terminal cost given by the performance of the primary DM M

$$P_\epsilon(\underline{\theta}) = p_0 P_F^M(\underline{\theta}) + p_1(1 - P_D^M(\underline{\theta})) \tag{3.40}$$

controls: We take the control variables of the problem to be the observation thresholds, so that there is a single control variable θ_1 to be chosen at DM 1, while there are two, $\theta_{i/0}, \theta_{i/1}$ to be chosen at every other. Choice of the control vari-

ables is unconstrained, i.e., $\theta_1, \theta_{i/0}, \theta_{i/1} \in \Re$, $\forall i$.

states: There are two state variables corresponding to the aggregate probabilities of false alarm and detection at each node, which evolve forward in stage (toward the primary DM) according to

$$\begin{aligned} P_F^i &= \sum_{u_{i-1}=0,1} \Pr(U_i = 1 | U_{i-1} = u_{i-1}, H_0) \Pr(U_{i-1} = u_{i-1} | H_0) \\ &= P_F^{i/0}(1 - P_F^{i-1}) + P_F^{i/1} P_F^{i-1} \end{aligned} \tag{3.41}$$

and

$$\begin{aligned} P_D^i &= \sum_{u_{i-1}=0,1} \Pr(U_i = 1 | U_{i-1} = u_{i-1}, H_1) \Pr(U_{i-1} = u_{i-1} | H_1) \\ &= P_D^{i/0}(1 - P_D^{i-1}) + P_D^{i/1} P_D^{i-1} \end{aligned} \tag{3.42}$$

where the fixed initial conditions are established by introducing a dummy node 0, with $P_F^0 = 0, P_D^0 = 0$, which gives

$$P_F^1 = P_F^{1/0}, \qquad P_D^1 = P_D^{1/0} \tag{3.43}$$

We may then view the network as a two-dimensional dynamical system with the states evolving according to the above state dynamics. Dependence on the control is masked by this notation, but the probabilities of false alarm and detection depend directly on the threshold parameters. Making this dependence explicit, we could write

$$\begin{aligned} P_F^i &= P_F^{i/0}(\theta_{i/0})(1 - P_F^{i-1}(\theta_{(i-1)/0}, \theta_{(i-1)/1})) \\ &\quad + P_F^{i/1}(\theta_{i/1}) P_F^{i-1}(\theta_{(i-1)/0}, \theta_{(i-1)/1}) \end{aligned} \tag{3.44}$$

and

$$P_D^i = P_D^{i/0}(\theta_{i/0})(1 - P_D^{i-1}(\theta_{(i-1)/0}, \theta_{(i-1)/1}))$$

$$+P_D^{i/1}(\theta_{i/1})P_D^{i-1}(\theta_{(i-1)/0},\theta_{(i-1)/1}) \tag{3.45}$$

costates: The costates represent the derivatives of the cost with respect to the states, i.e., the aggregate probabilities of false alarm and detection at each node. The costates at node i are denoted by

$$\mu_F^i = \frac{\partial P_\epsilon}{\partial P_F^i}(\underline{\theta}), \qquad \mu_D^i = \frac{\partial P_\epsilon}{\partial P_D^i}(\underline{\theta}) \tag{3.46}$$

and evolve backward in stage (toward DM 1) according to the dynamics

$$\begin{aligned} \mu_F^i &= [P_F^{(i+1)/1} - P_F^{(i+1)/0}]\mu_F^{(i+1)} \\ \mu_D^i &= [P_D^{(i+1)/1} - P_D^{(i+1)/0}]\mu_D^{(i+1)} \end{aligned} \tag{3.47}$$

with initial conditions

$$\mu_F^M = p_0, \qquad \mu_D^M = p_1 \tag{3.48}$$

Again, the effect of the controls on the evolution of the costate is not readily apparent in this notation, but is buried in the conditional probabilities.

An interesting observation pointed out in [61] is that since

$$\begin{aligned} [P_F^{(i+1)/1} - P_F^{(i+1)/0}] &\leq 1 \\ [P_D^{(i+1)/1} - P_D^{(i+1)/0}] &\leq 1 \end{aligned} \tag{3.49}$$

the sensitivity of the cost with respect to the state of a node i is monotonically nonincreasing as the distance from node i to the root node M increases.

Using the above relations, the partial derivatives of the cost with respect to the controls may be computed as

$$\begin{aligned} \frac{\partial P_\epsilon}{\partial \theta_{i/0}}(\underline{\theta}) &= (1-P_F^{i-1})\mu_F^i \frac{dP_F^{i/0}}{\partial \theta_{i/0}} + (1-P_D^{i-1})\mu_D^i \frac{dP_D^{i/0}}{\partial \theta_{i/0}} \\ &= (1-P_F^{i-1})\mu_F^i(-p_{Y_i|H_0}(\theta_{i/0}|H_0)) \\ &\quad +(1-P_D^{i-1})\mu_D^i(-p_{Y_i|H_1}(\theta_{i/0}|H_1)) \end{aligned}$$

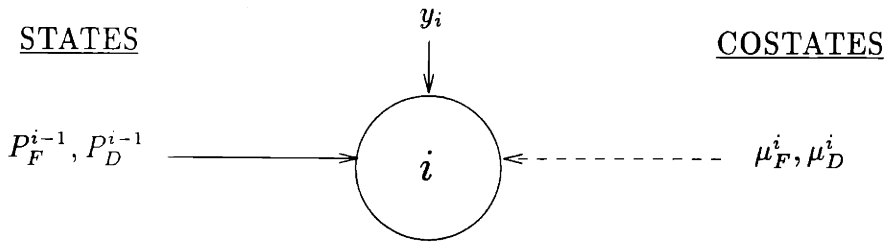


Figure 3-11: Information required for DM i to compute gradient

$$\begin{aligned}
\frac{\partial P_\epsilon}{\partial \theta_{i/1}}(\underline{\theta}) &= P_F^{i-1}\mu_F^i \frac{dP_F^{i/1}}{\partial \theta_{i/1}} + P_D^{i-1}\mu_D^i \frac{dP_D^{i/1}}{\partial \theta_{i/1}} \\
&= P_F^{i-1}\mu_F^i(-p_{Y_i|H_0}(\theta_{i/1}|H_0)) + P_D^{i-1}\mu_D^i(-p_{Y_i|H_1}(\theta_{i/1}|H_1)) \\
\frac{\partial P_\epsilon}{\partial \theta_1}(\underline{\theta}) &= \mu_F^1 \frac{dP_F^1}{\partial \theta_1} + \mu_D^1 \frac{dP_D^1}{\partial \theta_1} \\
&= \mu_F^1(-p_{Y_1|H_0}(\theta_1|H_0)) + \mu_D^1(-p_{Y_1|H_1}(\theta_1|H_1))
\end{aligned} \tag{3.50}$$

This technique is well known from the calculus of variations approach to solving optimal control problems, and is sometimes referred to as the adjoint method for computing gradients. The value of the formulation is clear from these expressions. The coefficients in each partial derivative at DM i have been expressed in terms of the states of the immediate predecessor node $i-1$, and the costates at node i which are functions only of information at the successor node $i+1$. This information dependence is illustrated in Figure 3-11, which indicates that it is sufficient for each DM to communicate only with its neighbors when computing its partial derivative. This method forms the basis of the back propagation technique to be presented in Chapter 5.

We now illustrate that the above formalism is sufficient to compute the partial derivatives by examining the two tandem teams we introduced in Chapter 2.

Example: 2-Tand

Letting $M = 2$, and associating DM A with DM 1, DM B with DM 2, $\theta_1 = \alpha$, $\theta_{2/0} = \beta_0$, and $\theta_{2/1} = \beta_1$, we refer to the above equations to write the following.

The adjoint equations are given by

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_{2/0}} &= (1-P_F^1)\mu_F^2 \frac{dP_F^{2/0}}{d\theta_{2/0}} + (1-P_D^1)\mu_D^2 \frac{dP_D^{2/0}}{d\theta_{2/0}} \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_{2/1}} &= P_F^1\mu_F^2 \frac{dP_F^{2/1}}{d\theta_{2/1}} + P_D^1\mu_D^2 \frac{dP_D^{2/1}}{d\theta_{2/1}} \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_1} &= \mu_F^1 \frac{dP_F^1}{d\theta_1} + \mu_D^1 \frac{dP_D^1}{d\theta_1}
\end{aligned} \tag{3.51}$$

The states propagate forward as

$$P_F^1 = P_F^1, \qquad P_D^1 = P_D^1 \tag{3.52}$$

$$P_F^2 = P_F^{2/0}(1-P_F^1) + P_F^{2/1}P_F^1, \quad P_D^2 = P_D^{2/0}(1-P_D^1) + P_D^{2/1}P_D^1 \tag{3.53}$$

and the costates propagate backwards as

$$\mu_F^2 = p_0, \qquad \mu_D^2 = -p_1 \tag{3.54}$$

$$\mu_F^1 = [P_F^{2/1} - P_F^{2/0}]\mu_F^2 = p_0[P_F^{2/1} - P_F^{2/0}] \tag{3.55}$$

$$\mu_D^1 = [P_D^{2/1} - P_D^{2/0}]\mu_D^2 = -p_1[P_D^{2/1} - P_D^{2/0}] \tag{3.56}$$

Then the partial derivatives are given by

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_{2/0}} &= p_0(1-P_F^1)\frac{dP_F^{2/0}}{d\theta_{2/0}} - p_1(1-P_D^1)\frac{dP_D^{2/0}}{d\theta_{2/0}} \\
&= -[1-P_F^1]p_0 p_{Y_2|H_0}(\theta_{2/0}|H_0) + [1-P_D^1]p_1 p_{Y_2|H_1}(\theta_{2/0}|H_1)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_{2/1}} &= p_0 P_F^1 \frac{dP_F^{2/1}}{d\theta_{2/1}} - p_1 P_D^1 \frac{dP_D^{2/1}}{d\theta_{2/1}} \\
&= -P_F^1 p_0 p_{Y_2|H_0}(\theta_{2/1}|H_0) + P_D^1 p_1 p_{Y_2|H_1}(\theta_{2/1}|H_1)
\end{aligned}$$

$$\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_1} = p_0[P_F^{2/1} - P_F^{2/0}]\frac{dP_F^1}{d\theta_1} - p_1[P_D^{2/1} - P_D^{2/0}]\frac{dP_D^1}{d\theta_1}$$

$$= -[P_F^{2/1} - P_F^{2/0}]p_0 p_{Y_1|H_0}(\theta_1|H_0) + [P_D^{2/1} - P_D^{2/0}]p_1 p_{Y_1|H_1}(\theta_1|H_1) \quad (3.57)$$

which agree with (3.14)-(3.16).

Example: 3-Tand

Letting $M = 3$, $\lambda_0 = \lambda_1 = 1$, and associating DM A with DM 1, DM B with DM 2, DM C with DM 3, $\theta_1 = \alpha$, $\theta_{2/0} = \beta_0$, $\theta_{2/1} = \beta_1$, $\theta_{3/0} = \xi_0$, and $\theta_{3/1} = \xi_1$, can write the following relations.

The adjoint equations are given by

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{3/0}} &= (1 - P_F^2)\mu_F^3 \frac{dP_F^{3/0}}{d\theta_{3/0}} + (1 - P_D^2)\mu_D^3 \frac{dP_D^{3/0}}{d\theta_{3/0}} \\
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{3/1}} &= P_F^2 \mu_F^3 \frac{dP_F^{3/1}}{d\theta_{3/1}} + P_D^2 \mu_D^3 \frac{dP_D^{3/1}}{d\theta_{3/1}} \\
\\
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{2/0}} &= (1 - P_F^1)\mu_F^2 \frac{dP_F^{2/0}}{d\theta_{2/0}} + (1 - P_D^1)\mu_D^2 \frac{dP_D^{2/0}}{d\theta_{2/0}} \\
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{2/1}} &= P_F^1 \mu_F^2 \frac{dP_F^{2/1}}{d\theta_{2/1}} + P_D^1 \mu_D^2 \frac{dP_D^{2/1}}{d\theta_{2/1}} \\
\\
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_1} &= \mu_F^1 \frac{dP_F^1}{d\theta_1} + \mu_D^1 \frac{dP_D^1}{d\theta_1}
\end{aligned} \quad (3.58)$$

The states propagate forward as

$$P_F^1 = P_F^1, \quad P_D^1 = P_D^1 \quad (3.59)$$

$$P_F^2 = P_F^{2/0}(1 - P_F^1) + P_F^{2/1} P_F^1, \quad P_D^2 = P_D^{2/0}(1 - P_D^1) + P_D^{2/1} P_D^1 \quad (3.60)$$

$$P_F^3 = P_F^{3/0}(1 - P_F^2) + P_F^{3/1} P_F^2, \quad P_D^3 = P_D^{3/0}(1 - P_D^2) + P_D^{3/1} P_D^2 \quad (3.61)$$

and the costates propagate backwards as

$$\mu_F^3 = p_0, \quad \mu_D^3 = -p_1 \quad (3.62)$$

$$\mu_F^2 = [P_F^{3/1} - P_F^{3/0}]\mu_F^3 = p_0[P_F^{3/1} - P_F^{3/0}] \quad (3.63)$$

$$\mu_D^2 = [P_D^{3/1} - P_D^{3/0}]\mu_D^3 = -p_1[P_D^{3/1} - P_D^{3/0}] \tag{3.64}$$

$$\mu_F^1 = [P_F^{2/1} - P_F^{2/0}]\mu_F^2 = p_0[P_F^{2/1} - P_F^{2/0}][P_F^{3/1} - P_F^{3/0}] \tag{3.65}$$

$$\mu_D^1 = [P_D^{2/1} - P_D^{2/0}]\mu_D^2 = -p_1[P_D^{2/1} - P_D^{2/0}][P_D^{3/1} - P_D^{3/0}] \tag{3.66}$$

Then the partial derivatives are given by

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{3/0}} &= p_0(1 - P_F^2)\frac{dP_F^{3/0}}{d\theta_{3/0}} - p_1(1 - P_D^2)\frac{dP_D^{3/0}}{d\theta_{3/0}} \\
&= -[(1 - P_F^1)(1 - P_F^{2/0}) + P_F^1(1 - P_F^{2/1})]p_0 p_{Y_3|H_0}(\theta_{3/0}|H_0) \\
&\quad +[(1 - P_D^1)(1 - P_D^{2/0}) + P_D^1(1 - P_D^{2/1})]p_1 p_{Y_3|H_1}(\theta_{3/0}|H_1)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{3/1}} &= p_0 P_F^2 \frac{dP_F^{3/1}}{d\theta_{3/1}} - p_1 P_D^2 \frac{dP_D^{3/1}}{d\theta_{3/1}} \\
&= -[(1 - P_F^1)P_F^{2/0} + P_F^1 P_F^{2/1}]p_0 p_{Y_3|H_0}(\theta_{3/1}|H_0) \\
&\quad +[(1 - P_D^1)P_D^{2/0} + P_D^1 P_D^{2/1}]p_1 p_{Y_3|H_1}(\theta_{3/1}|H_1)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{2/0}} &= p_0(P_F^{3/1} - P_F^{3/0})(1 - P_F^1)\frac{dP_F^{2/0}}{d\theta_{2/0}} - p_1(P_D^{3/1} - P_D^{3/0})(1 - P_D^1)\frac{dP_D^{2/0}}{d\theta_{2/0}} \\
&= -[(1 - P_F^1)(P_F^{3/1} - P_F^{3/0})]p_0 p_{Y_2|H_0}(\theta_{2/0}|H_0) \\
&\quad +[(1 - P_D^1)(P_D^{3/1} - P_D^{3/0})]p_1 p_{Y_2|H_1}(\theta_{2/0}|H_1)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_{2/1}} &= p_0(P_F^{3/1} - P_F^{3/0})P_F^1\frac{dP_F^{2/1}}{d\theta_{2/1}} - p_1(P_D^{3/1} - P_D^{3/0})P_D^1\frac{dP_D^{2/1}}{d\theta_{2/1}} \\
&= -[P_F^1(P_F^{3/1} - P_F^{3/0})]p_0 p_{Y_2|H_0}(\theta_{2/1}|H_0) \\
&\quad +[P_D^1(P_D^{3/1} - P_D^{3/0})]p_1 p_{Y_2|H_1}(\theta_{2/1}|H_1)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \theta_1} &= p_0(P_F^{2/1} - P_F^{2/0})(P_F^{3/1} - P_F^{3/0})\frac{dP_F^1}{d\theta_1} - p_1(P_D^{2/1} - P_D^{2/0})(P_D^{3/1} - P_D^{3/0})\frac{dP_D^1}{d\theta_1} \\
&= -[(P_F^{2/1} - P_F^{2/0})(P_F^{3/1} - P_F^{3/0})]p_0 p_{Y_1|H_0}(\theta_1|H_0) \\
&\quad +[(P_D^{2/1} - P_D^{2/0})(P_D^{3/1} - P_D^{3/0})]p_1 p_{Y_1|H_1}(\theta_1|H_1)
\end{aligned} \tag{3.67}$$

which agree with (3.29)-(3.33).

3.3 Properties of the Probability of Error Criterion for the Linear Threshold Parameterization

In this section we investigate the properties of the criterion function which result from the parameterization suggested in Section 3.1. We are particularly concerned with the conditions which must be satisfied by the conditional densities of the hypothesis test in order that the resulting cost can be optimized using gradient-based techniques. At the most fundamental level, the cost function must be shown to be differentiable as a function of the observation thresholds in order to even consider applying gradient-based optimization techniques. Characterization of the stationary points is also critical for knowing what to expect from such methods, i.e., in order to determine whether the algorithm can be expected to converge to a globally optimal solution or at best to a locally optimal one. Additional smoothness properties, such as Lipschitz continuity of the gradient for example, may required to specify the range of allowable step sizes for constant stepsize methods or to invoke the descent lemma (Appendix B) when using the descent approach to prove convergence as we do in Chapters 6 and 7. The development of this section is laborious, but is necessary for establishing the validity of the training algorithms we present in Chapters 4 and 5.

For the single DM problem, we discuss both the Bayes risk $J_B(\theta)$ as well as the probability of error $P_\epsilon(\theta)$ since the Bayes risk problem arises as a subproblem in conjunction with the team problem. For the team problem, we consider only $P_\epsilon(\underline{\theta})$, although similar results to those we provide could certainly be provided for the case $J_B(\underline{\theta})$.

3.3.1 The Single DM Case

We consider the single DM case not only because it is an obvious natural starting point for the network problem, but also because we find that the team problem has identical structure to the single DM problem when considered in a person-by-person manner.

Properties of $J_B(\theta), P_\epsilon(\theta)$

Assume that the DM processes a scalar-valued observation as discussed in Section 2.2. For a linear threshold rule of the form (2.21) the probability of error as a function of the threshold is given by

$$P_\epsilon(\theta) = p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{3.68}$$

In order to get some feel for the error surface, we can plot $P_\epsilon(\theta)$ as a function of θ for the Gaussian detection problem as shown in Figure 3-12. While the plot indicates that the function is clearly not convex, it is still bowl-like, i.e., unimodal with a single global minimum. We will give conditions in the following which ensure that this property holds for general distributions. Also note that in the limit as the threshold θ becomes small or large, the surface flattens out toward the value 0.5.

For the unequal cost formulation, the criterion function under linear threshold parameterization is given by

$$J_B(\theta) = \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{3.69}$$

where λ_0 and λ_1 are positive bounded constants. It is not necessary that the costs be normalized so that $\lambda_0 + \lambda_1 = 1$, although from the point of view of optimization, normalization minimizes the effect that extreme choices of cost may have on the error surface. A typical J_B surface is shown in Figure 3-13.

It is possible to make some further limited qualitative statements about how the shape of the $P_\epsilon(\theta)$ and $J_B(\theta)$ cost surfaces are altered by variations in the parameters

of the underlying hypothesis test by resorting to numerical experiments. The following figures are parametric studies of the Gaussian detection problem

$$Y = \begin{cases} \mu_0 + W & \text{if } H = H_0 \\ \mu_1 + W & \text{if } H = H_1 \end{cases} \tag{3.70}$$

where $W \sim N(0, \sigma^2)$, as the parameters of the test are varied one at a time.

Figure 3-14 illustrates the P_ϵ surface as the variance σ^2 is changed by factors of two and the prior probabilities are held at $p_0 = p_1 = 0.5$. Notice that changes in the variance act symmetrically on the surface, and do not affect the minimizing value of θ, although lower values of variance obviously reduce the minimum attainable probability of error. The sides of the bowl become steeper with decreasing variance, while the flat edges are extended toward the minimum. Conversely, increasing the variance decreases the slope of the bowl, and extends it toward the edges. Notice that as the variance is increased, the size of the level sets containing the minimizing value of θ

$$\{\theta | P_\epsilon(\theta) - P_\epsilon^* \leq C,\ C > 0\} \tag{3.71}$$

where P_ϵ^* is the minimum value and C is some scalar, increase also. Thus, there are more points within a fixed percentage of the optimum in the case of higher variance. Note also that the curves cross over one another, so that lower variance does not guarantee uniformly better performance, but only in the vicinity of the optimum.

Figure 3-15 illustrates the surface as the prior probabilities p_0 and $p_1 = 1 - p_0$ are made asymmetric. In contrast to changing the variance, changing the priors warps the entire cost surface by pulling the edges of the bowl up and down. As θ approaches $+\infty$, $P_\epsilon(\theta)$ approaches p_1 as the errors become exclusively misses, and it approaches p_0 as θ goes to $-\infty$ and the errors become exclusively false alarms. Notice that this has resulted in the bowl having walls with significantly different slopes. Most importantly, however, breaking the symmetry of the priors shifts the minimizing value of the threshold left or right as the underlying hypothesis test is biased. In particular, as p_0 is made to exceed p_1, the minimizing value of θ increases, which makes sense

because if H_0 is known to be more likely, the region for which H_0 is the correct decision should be enlarged. Conversely, as p_1 exceeds p_0 the minimizing value of θ is reduced.

Figure 3-16 illustrates the effect of changing the variance when the prior probabilities are no longer balanced. Note that in this case, changing the variance does shift the optimal value, as is evident in equation (2.22).

Figure 3-17 illustrates the effect of symmetric changes in the spread of the means μ_0 and μ_1. As the distance between the means increases, the P_ϵ surface is shifted uniformly lower, meaning that lower values of error are attained for any value of the threshold than for the same problem with the means closer. Thus, spreading the means has resulted in an "easier" hypothesis testing problem. The minimizing value of θ remains the same under these symmetric changes, although the optimal attainable cost is reduced. In contrast to changing the variances, changing the means has had no visible impact on the size of the level sets around the minimum point.

The impact of varying costs on the J_B cost surface resembles the effect of varying the prior probabilities, without the normalization. Thus, in Figure 3-18 the same general changes in the surface are evident, but as the difference in the costs increases, the maximum derivative in the cost increases, and the mimimum becomes less pronounced.

We now indicate properties of the conditional densities which must hold to ensure certain essential properties of the cost. Since many of the later results of this report rely on these properties holding for the unequal cost problem, we formulate the propositions for $J_B(\theta)$, with the corresponding properties for $P_\epsilon(\theta)$ evident by evaluating the conclusions for $\lambda_0 = \lambda_1 = 1$.

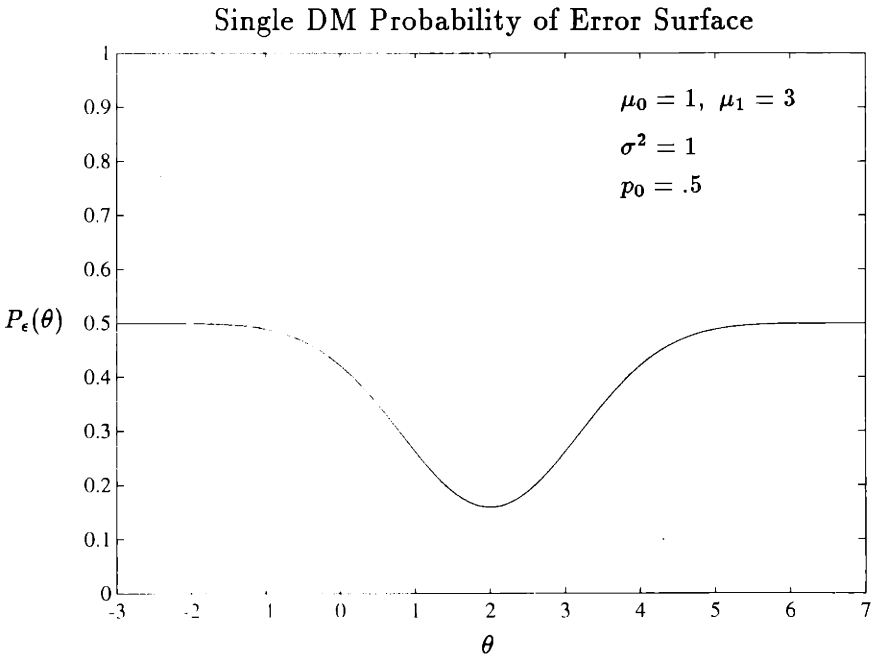


Figure 3-12: Single DM Probability of Error Surface; Gaussian detection

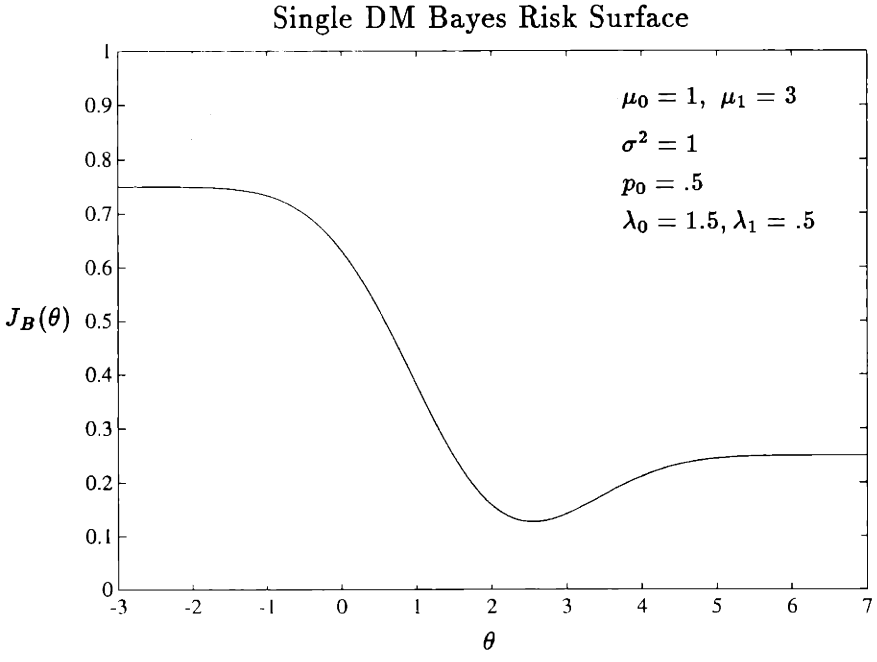


Figure 3-13: Single DM Bayes Risk (unequal cost) Surface; Gaussian detection

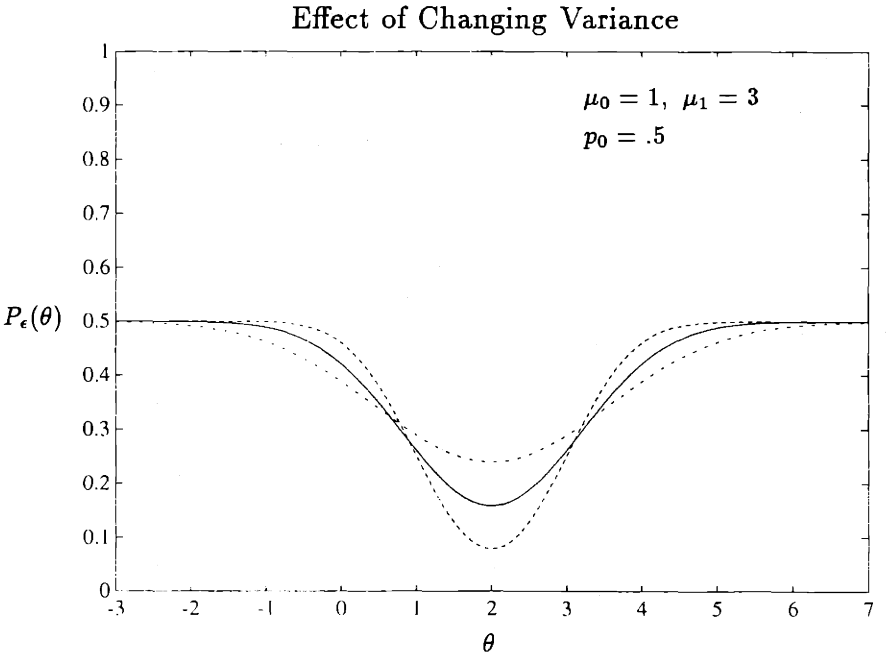


Figure 3-14: Single DM $P_\epsilon(\theta)$: $\sigma^2 = 1$ (solid), $\sigma^2 = .5$ (dashed), $\sigma^2 = 2$ (dotted and dashed); Gaussian detection

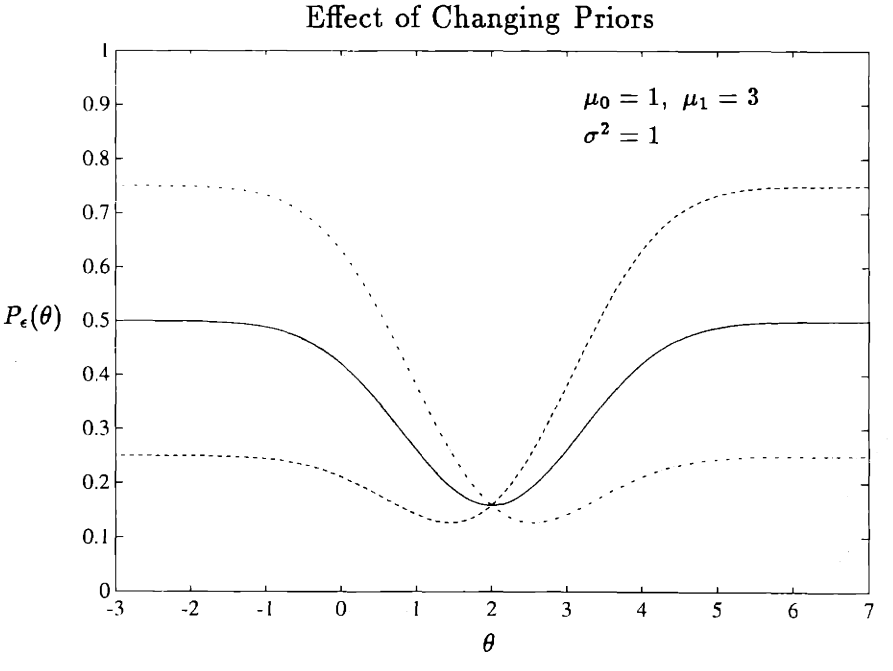


Figure 3-15: Single DM $P_\epsilon(\theta)$: $p_0 = .5$ (solid), $p_0 = .25$ (dashed), $p_0 = .75$ (dotted and dashed); Gaussian detection

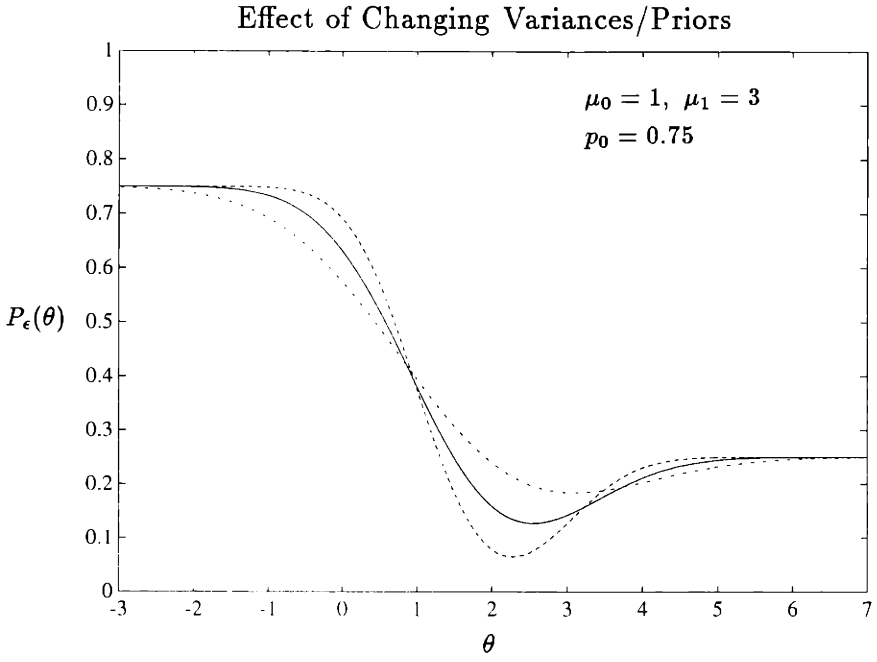


Figure 3-16: Single DM $P_\epsilon(\theta)$: $\sigma^2 = 1$ (solid), $\sigma^2 = .25$ (dashed), $\sigma^2 = 2$ (dotted and dashed); Gaussian detection

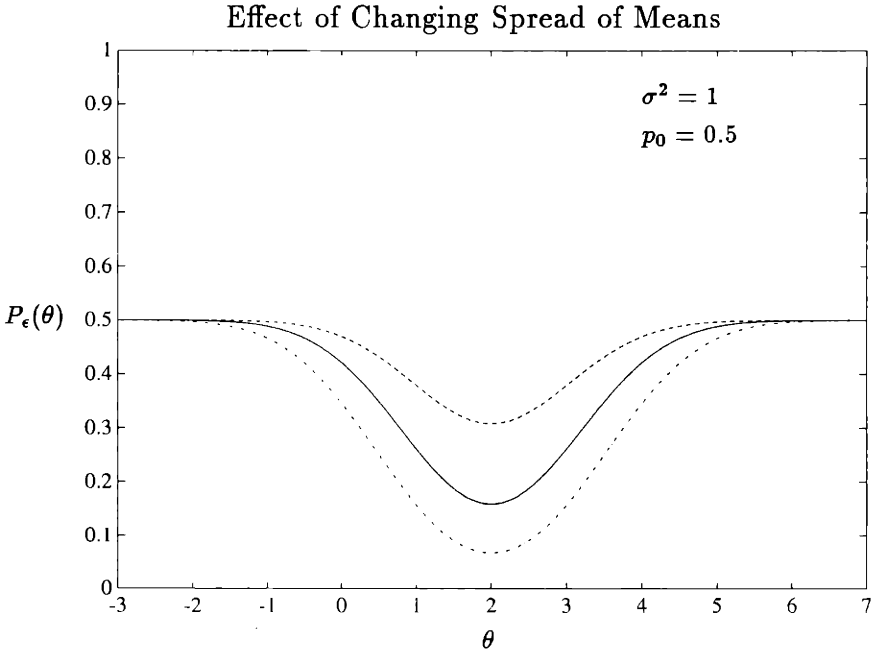


Figure 3-17: Single DM $P_\epsilon(\theta)$: $\mu_0 = 1$, $\mu_1 = 3$ (solid); $\mu_0 = 1.5$, $\mu_1 = 2.5$ (dashed); $\mu_0 = .5$, $\mu_1 = 3.5$ (dotted and dashed); Gaussian detection

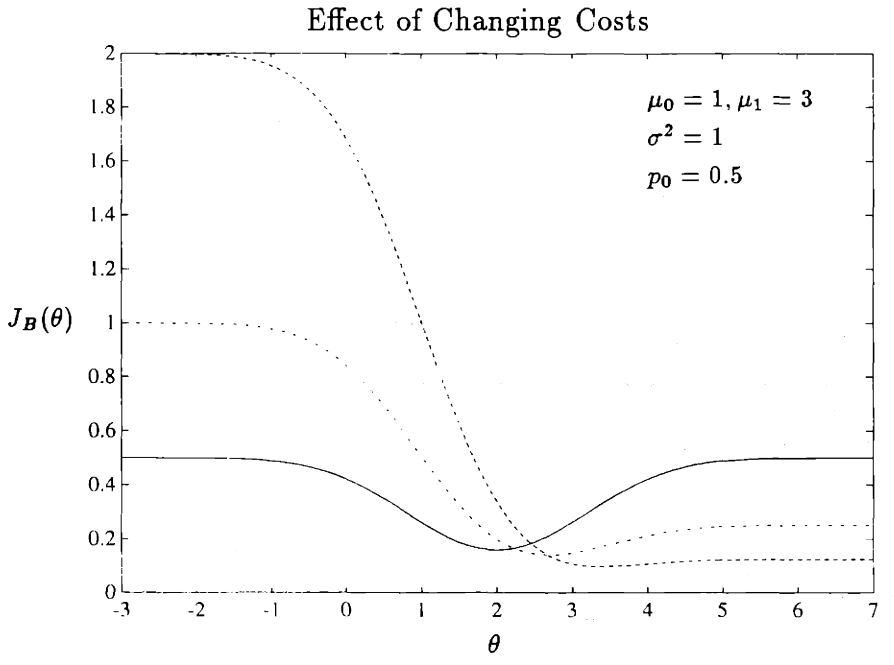


Figure 3-18: Single DM $J_B(\theta)$: $\lambda_0 = 1$, $\lambda_1 = 1$ (solid); $\lambda_0 = 4$, $\lambda_1 = .25$ (dashed); $\lambda_0 = 2$, $\lambda_1 = .5$ (dotted and dashed); Gaussian detection

Proposition 3.2 (Properties of $J_B(\theta)$)

For

$$J_B(\theta) = \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{3.72}$$

with λ_0 and λ_1 bounded positive constants,

(a) *(Boundedness) There holds $0 \leq J_B(\theta) \leq (\lambda_0 p_0 + \lambda_1 p_1)$*

(b) *(Differentiability) If the conditional densities $p_{Y|H_j}(y|H_j), j = 0, 1$ are continuous for all $y \in \Re$, then $J_B(\theta)$ is a continuously differentiable function of θ.*

Proof.

(a) The lower bound follows from the nonnegativity of the integrands $p_{Y|H_0}(y|H_0)$ and $p_{Y|H_1}(y|H_1)$. The upper bound is easily obtained as

$$J_B(\theta) \;=\; \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy$$

$$\leq \lambda_0 p_0 \int_{-\infty}^{\infty} p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^{\infty} p_{Y|H_1}(y|H_1)\,dy$$
$$\leq \lambda_0 p_0 + \lambda_1 p_1 \tag{3.73}$$

using the fact that the functions $p_{Y|H}$ are probability density functions.
(b) The functions $F_0(\theta) = \int_{\theta}^{\infty} p_{Y|H_0}(y|H_0)\,dy$ and $F_1(\theta) = \int_{-\infty}^{\theta} p_{Y|H_1}(y|H_1)\,dy$ are continuous functions of θ if the conditional densities are continuous for all $y \in \Re$ [63]. Since $J_B(\theta) = \lambda_0 p_0 F_0(\theta) + \lambda_1 p_1 F_1(\theta)$, it is also a continuous function of θ.

Since the conditional densities are continuous, F_0 and F_1 are differentiable functions of θ, with derivatives given by

$$\frac{d}{d\theta} F_0(\theta) = \frac{d}{d\theta}\left(\int_{\theta}^{\infty} p_{Y|H_0}(y|H_0)\,dy\right) = -p_{Y|H_0}(\theta|H_0) \tag{3.74}$$

and

$$\frac{d}{d\theta} F_1(\theta) = \frac{d}{d\theta}\left(\int_{-\infty}^{\theta} p_{Y|H_1}(y|H_1)\,dy\right) = p_{Y|H_1}(\theta|H_1) \tag{3.75}$$

so that

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 p_{Y|H_0}(\theta|H_0) + \lambda_1 p_1 p_{Y|H_1}(\theta|H_1) \tag{3.76}$$

which is also a continuous function of θ.

■

Thus, the Bayes risk J_B is easily shown to be bounded, and is guaranteed to be a differentiable function of θ under a continuity assumption on the conditional densities. Without such an assumption, gradient-based methods cannot be directly applied in general.

Properties of the Derivatives

Under some additional reasonable assumptions on the conditional densities and their derivatives, some nice smoothness properties on the first and second derivatives of $J_B(\theta)$ can be established. These conditions on the cost will be required when we prove convergence of the training algorithms in Chapters 6 and 7.

Proposition 3.3 (Properties of the Derivatives of $J_B(\theta)$)

Assume that $p_{Y|H_0}, p_{Y|H_1}$ are continuous and twice differentiable, and that there exist bounded positive constants B_0, B_1, B_2 and B_3 such that

$$p_{Y|H_0}(y|H_0) \leq B_0,\ p_{Y|H_1}(y|H_1) \leq B_1,\ \ \forall y \in \Re \tag{3.77}$$

$$\left|\frac{d}{dy}(p_{Y|H_0}(y|H_0))\right| \leq B_2,\ \left|\frac{d}{dy}(p_{Y|H_1}(y|H_1))\right| \leq B_3,\ \ \forall y \in \Re \tag{3.78}$$

Then, for

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 p_{Y|H_0}(\theta|H_0) + \lambda_1 p_1 p_{Y|H_1}(\theta|H_1) \tag{3.79}$$

$$\frac{d^2J_B}{d\theta^2}(\theta) = -\lambda_0 p_0 \frac{d}{dy}(p_{Y|H_0}(\theta|H_0)) + \lambda_1 p_1 \frac{d}{dy}(p_{Y|H_1}(\theta|H_1)) \tag{3.80}$$

with λ_0 and λ_1 bounded positive constants it holds that

(a) *(Boundedness of the First Derivative) For $K_1 = \lambda_0 p_0 B_0 + \lambda_1 p_1 B_1$,*

$$\left|\frac{dJ_B}{d\theta}(\theta)\right| \leq K_1,\ \ \forall \theta \in \Re \tag{3.81}$$

(b) *(Boundedness of the Second Derivative, Lipschitz Continuity of the First Derivative)*

(i) For $K_2 = \lambda_0 p_0 B_2 + \lambda_1 p_1 B_3$,

$$\left|\frac{d^2J_B}{d\theta^2}(\theta)\right| \leq K_2,\ \ \forall \theta \in \Re \tag{3.82}$$

(ii) For $L = \lambda_0 p_0 B_2 + \lambda_1 p_1 B_3$,

$$\left|\frac{dJ_B}{d\theta}(\theta_1) - \frac{dJ_B}{d\theta}(\theta_2)\right| \leq L|\theta_1 - \theta_2|,\ \ \forall \theta_1, \theta_2 \in \Re \tag{3.83}$$

Proof.

(a)

$$\begin{aligned}\left|\frac{dJ_B}{d\theta}(\theta)\right| &= |-\lambda_0 p_0 p_{Y|H_0}(\theta|H_0) + \lambda_1 p_1 p_{Y|H_1}(\theta|H_1)| \\ &\leq |-\lambda_0 p_0 p_{Y|H_0}(\theta|H_0)| + |\lambda_1 p_1 p_{Y|H_1}(\theta|H_1)| \\ &\leq \lambda_0 p_0 |p_{Y|H_0}(\theta|H_0)| + \lambda_1 p_1 |p_{Y|H_1}(\theta|H_1)| \\ &\leq \lambda_0 p_0 B_0 + \lambda_1 p_1 B_1 \end{aligned} \tag{3.84}$$

so that it suffices to choose $K_1 = \lambda_0 p_0 B_0 + \lambda_1 p_1 B_1$.

(b) (i) We write

$$\begin{aligned}\left|\frac{d^2 J_B}{d\theta^2}(\theta)\right| &= \left|-\lambda_0 p_0 \frac{d}{dy}(p_{Y|H_0}(\theta|H_0)) + \lambda_1 p_1 \frac{d}{dy}(p_{Y|H_1}(\theta|H_1))\right| \\ &\leq \lambda_0 p_0 |\frac{d}{dy}(p_{Y|H_0}(\theta|H_0))| + \lambda_1 p_1 |\frac{d}{dy}(p_{Y|H_1}(\theta|H_1))| \\ &\leq \lambda_0 p_0 B_2 + \lambda_1 p_1 B_3 \end{aligned} \tag{3.85}$$

so that it suffices to choose $K_2 = \lambda_0 p_0 B_2 + \lambda_1 p_1 B_3$.

(ii) Invoking the Mean Value Inequality (Appendix B), we take L to be an upper bound on the quantity

$$\sup_{\theta \in \Re} \left|\frac{d^2 J_B}{d\theta^2}(\theta)\right| \tag{3.86}$$

which from part (i) is given by $L = K_2$.

■

The Gaussian Detection Problem

It is possible to analytically compute the bounds of Proposition 3.3 for the Gaussian detection problem, and this is worth doing because it clarifies the relationship between the underlying properties of the hypothesis test and the desirable properties of the derivatives. We also wish to demonstrate that the assumptions we make later do in fact hold for the primary case of interest.

For this case, the conditional densities are of the form

$$p_{Y|H_0}(y|H_0) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-\mu_0)^2}{2\sigma^2}}, \quad p_{Y|H_1}(y|H_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-\mu_1)^2}{2\sigma^2}} \tag{3.87}$$

which are both clearly continuous and infinitely differentiable. Then $J_B(\theta)$ takes the form

$$J_B(\theta) = \lambda_0 p_0 \int_\theta^\infty \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-\mu_0)^2}{2\sigma^2}}\, dy + \lambda_1 p_1 \int_{-\infty}^\theta \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-\mu_1)^2}{2\sigma^2}}\, dy \tag{3.88}$$

with first derivative

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}} + \lambda_1 p_1 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}} \tag{3.89}$$

and with second derivative

$$\frac{d^2J_B}{d\theta^2}(\theta) = \lambda_0 p_0 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}}(\theta-\mu_0) - \lambda_1 p_1 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}}(\theta-\mu_1) \tag{3.90}$$

Examples of these derivatives are displayed in Figures 3-19 and 3-20 for the case $\lambda_0 = \lambda_1 = 1$.

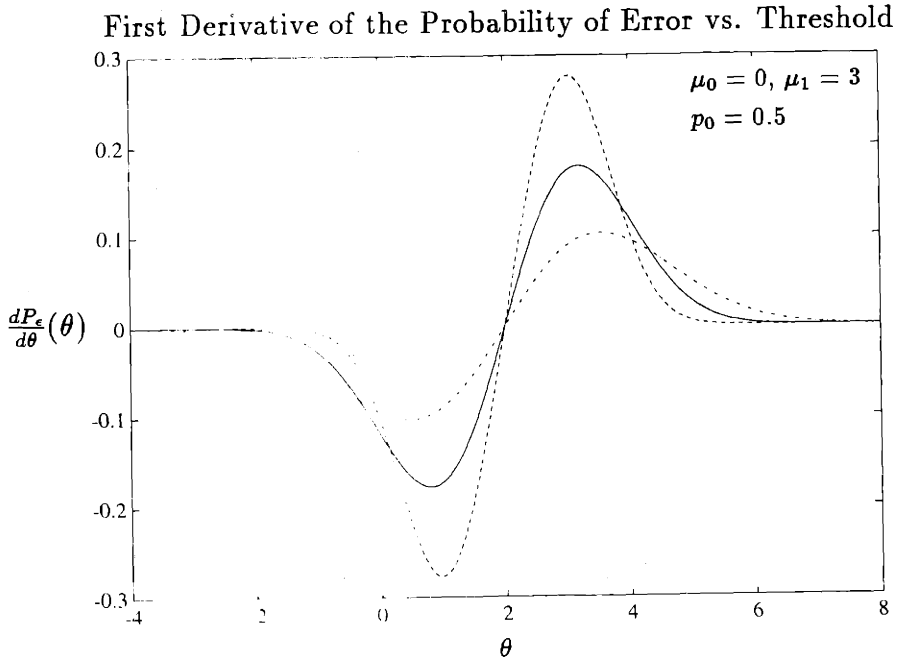


Figure 3-19: Single DM First Derivative of the Probability of Error: $\sigma^2 = 1$ (solid), $\sigma^2 = 0.5$ (dashed), $\sigma^2 = 2$ (dotted and dashed); Gaussian case

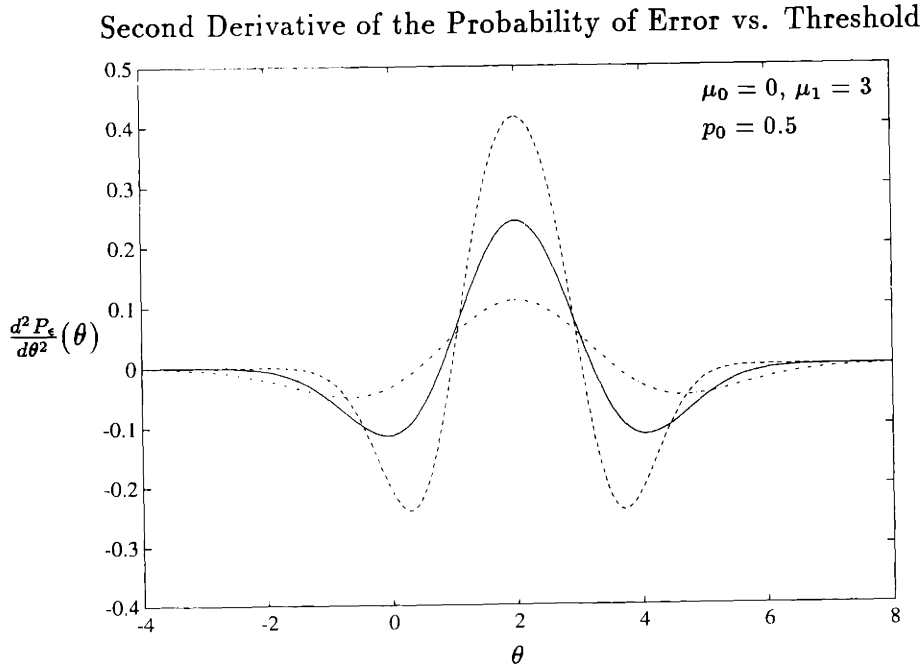


Figure 3-20: Single DM Second Derivative of the Probability of Error: $\sigma^2 = 1$ (solid), $\sigma^2 = 0.5$ (dashed), $\sigma^2 = 2$ (dotted and dashed); Gaussian case

Proposition 3.4 (Single DM Gaussian Detection Problem)

Let $J_B(\theta)$ be defined as in (3.88) with λ_0 and λ_1 bounded positive constants. Then

(a) *(Differentiability) $J_B(\theta)$ is infinitely continuously differentiable*

(b) *(First Derivative)*

(i) (Bounded)

$$\left|\frac{dJ}{d\theta}(\theta)\right| \leq K_1, \quad \forall \theta \in \Re \tag{3.91}$$

for

$$K_1 = (\lambda_0 p_0 + \lambda_1 p_1)\frac{1}{\sigma\sqrt{2\pi}} \tag{3.92}$$

(ii) (Lipschitz)

$$\left|\frac{dJ_B}{d\theta}(\theta_1) - \frac{dJ_B}{d\theta}(\theta_2)\right| \leq L_1|\theta_1 - \theta_2|, \quad \forall \theta_1, \theta_2 \in \Re \tag{3.93}$$

for

$$L_1 = \frac{\lambda_0 p_0 + \lambda_1 p_1}{\sigma^2\sqrt{2\pi}} e^{-\frac{1}{2}} \tag{3.94}$$

(c) *(Second Derivative)*

(i) (Bounded)

$$\left|\frac{d^2J}{d\theta^2}(\theta)\right| \leq K_2, \quad \forall \theta \in \Re \tag{3.95}$$

for

$$K_2 = L_1 = \frac{\lambda_0 p_0 + \lambda_1 p_1}{\sigma^2\sqrt{2\pi}} e^{-\frac{1}{2}} \tag{3.96}$$

(ii) (Lipschitz)

$$\left|\frac{d^2J_B}{d\theta^2}(\theta_1) - \frac{d^2J_B}{d\theta^2}(\theta_2)\right| \leq L_2|\theta_1 - \theta_2|, \quad \forall \theta_1, \theta_2 \in \Re \tag{3.97}$$

for

$$L_2 = \frac{\lambda_0 p_0 + \lambda_1 p_1}{\sigma^3\sqrt{2\pi}}(1 + 2e^{-1}) \tag{3.98}$$

Proof.

(a) Follows from well-known properties of exponential [54].

(b)(i) It holds that

$$\begin{aligned}\left|\frac{dJ_B}{d\theta}(\theta)\right| &= \left|-\lambda_0 p_0 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}} + \lambda_1 p_1 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}}\right| \\ &\leq \lambda_0 p_0 \frac{1}{\sigma\sqrt{2\pi}} \left|e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}}\right| + \lambda_1 p_1 \frac{1}{\sigma\sqrt{2\pi}} \left|e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}}\right| \\ &\leq (\lambda_0 p_0 + \lambda_1 p_1)\frac{1}{\sigma\sqrt{2\pi}} \qquad (3.99)\end{aligned}$$

so that it suffices to choose $K_1 = (\lambda_0 p_0 + \lambda_1 p_1)\frac{1}{\sigma\sqrt{2\pi}}$.

(ii) The bound follows directly from the Mean Value Inequality (Appendix B). L_1 is taken to be an upper bound of the quantity

$$\sup_{\theta\in\Re}\left|\frac{d^2 J_B}{d\theta^2}(\theta)\right| \qquad (3.100)$$

the existence of which we now demonstrate. The following relationships hold $\forall\theta$:

$$\begin{aligned}\left|\frac{d^2 J_B}{d\theta^2}(\theta)\right| &= \left|\lambda_0 p_0 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}}(\theta-\mu_0) - \lambda_1 p_1 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}}(\theta-\mu_1)\right| \\ &\leq \left|\lambda_0 p_0 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}}(\theta-\mu_0)\right| + \left|\lambda_1 p_1 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}}(\theta-\mu_1)\right| \\ &= \frac{\lambda_0 p_0}{\sigma^3\sqrt{2\pi}} \left|e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}}(\theta-\mu_0)\right| + \frac{\lambda_1 p_1}{\sigma^3\sqrt{2\pi}} \left|e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}}(\theta-\mu_1)\right| \qquad (3.101)\end{aligned}$$

where we have used the triangle inequality and the nonnegativity of the costs λ_0 and λ_1. To finish the bound, we need the following result concerning the function $e^{\frac{-u^2}{2\sigma^2}}u$, which is plotted in Figure 3-21.

Lemma 3.1 (Maximum of $e^{\frac{-u^2}{2\sigma^2}}u$)

There holds

$$\left|e^{\frac{-u^2}{2\sigma^2}}u\right| \leq \sigma e^{-\frac{1}{2}} \qquad (3.102)$$

for all $u \in \Re$

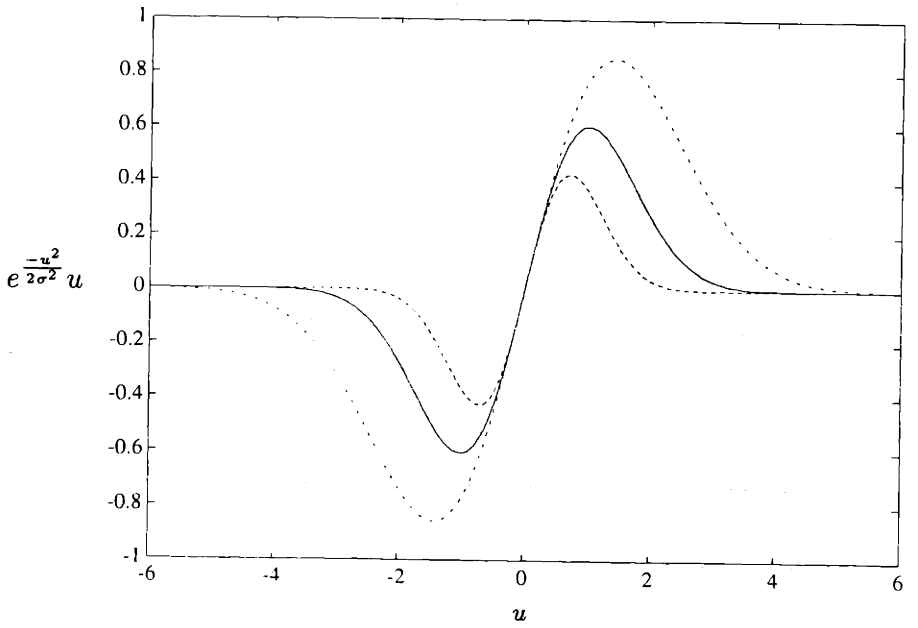


Figure 3-21: $\sigma^2 = 1$ (solid), $\sigma^2 = .5$ (dashed), $\sigma^2 = 2$ (dotted and dashed)

Proof. The stationary points of

$$f(u) = e^{\frac{-u^2}{2\sigma^2}}u \tag{3.103}$$

are solutions to the equation

$$\frac{df}{du}(u) = e^{-\frac{u^2}{2\sigma^2}} - (1/\sigma^2)u^2 e^{-\frac{u^2}{2\sigma^2}} = 0 \tag{3.104}$$

which implies

$$(1/\sigma^2)u^2 = 1 \tag{3.105}$$

or

$$u = \pm\sigma \tag{3.106}$$

This gives

$$\min_u f(u) = -\sigma e^{-\frac{1}{2}}, \quad \max_u f(u) = \sigma e^{-\frac{1}{2}} \tag{3.107}$$

which imply

$$\left| e^{\frac{-u^2}{2\sigma^2}} u \right| \leq \sigma e^{-\frac{1}{2}} \tag{3.108}$$

and the lemma is proved. □

The lemma implies that

$$\begin{aligned} \frac{\lambda_0 p_0}{\sigma^3\sqrt{2\pi}} \left| e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}} (\theta-\mu_0) \right| &+ \frac{\lambda_1 p_1}{\sigma^3\sqrt{2\pi}} \left| e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}} (\theta-\mu_1) \right| \\ &\leq \frac{\lambda_0 p_0}{\sigma^3\sqrt{2\pi}} (\sigma e^{-\frac{1}{2}}) + \frac{\lambda_1 p_1}{\sigma^3\sqrt{2\pi}} (\sigma e^{-\frac{1}{2}}) \\ &= \frac{\lambda_0 p_0 + \lambda_1 p_1}{\sigma^2\sqrt{2\pi}} e^{-\frac{1}{2}} \end{aligned} \tag{3.109}$$

and it is sufficient to choose

$$L_1 = \frac{\lambda_0 p_0 + \lambda_1 p_1}{\sigma^2\sqrt{2\pi}} e^{-\frac{1}{2}} \tag{3.110}$$

(c)(i) Follows immediately from b(ii) that $K_2 = L_1$.

(ii) Derivation of the Lipschitz constant for the second derivative proceeds by obtaining a bound on the third derivative. The development is sufficiently similar to part b(i) that we do not repeat it here. The main difference is that the bound

$$\left| e^{-\frac{u^2}{2\sigma^2}} u^2 \right| \leq 2\sigma^2 e^{-1} \tag{3.111}$$

must be obtained.

■

Note that a Lipschitz bound on the second derivative was also obtained. We will need such a condition to argue convergence of two-sided finite difference techniques.

It should be emphasized that these are rather weak choices of L_1, L_2 in many cases. The reason for this is that any information about the signal to noise ratio, i.e., the spread of the means compared to the noise variance, is ignored when obtaining the bound. However, the bound is sufficient to indicate that the derivatives become less Lipschitz as the costs are increased or as the variance is reduced, results which

are intuitive when compared with Figures 3-18 and 3-14, respectively.

3.3.2 Team Case

In this section we investigate properties of the probability of error criterion which result from the linear threshold parameterization for the team problem. We assume that the parameter vector of thresholds $\underline{\theta} \in \Re^N$, so that $P_\epsilon(\underline{\theta}) : \Re^N \mapsto \Re$. We first error surfaces of the type which arise for the teams introduced in Chapter 2 using numerical experiments. We then provide some analytic results concerning the properties of the cost for general teams.

Characteristics of the Surface for Team of DMs In this section we illustrate some typical error surfaces for the team problem as we did previously for the single DM problem. This is useful for getting a qualitative feel for the surface, as well as for anticipating difficulties which might arise when applying gradient based (hill-climbing) optimization techniques to these surfaces.

We will focus exclusively on the 2-Tand topology, as this is sufficient for our purposes here. Because 2-Tand is parameterized by the three thresholds α, β_0, and β_1, we must display contour and mesh plots as a function of two parameters at a time, with the third held at a fixed value. We will fix the third parameter at its optimal value throughout. Of course, this approach provides an incomplete picture of the cost surface, but it is sufficient to determine some general properties of the landscape. We present contour and mesh plots for the 2-Tand Gaussian detection cases previously described in Figures 2-9(a), 2-10(b), and 2-11(c). These cases cover changes in variance between the DMs as well as a priori bias.

Figures 3-22 - 3-24 illustrate the P_ϵ^{2-Tand} cost surfaces corresponding to Figure 2-9(a). In Figure 3-22, the cost is shown as a function of β_0 and β_1 with α held fixed at $\alpha = \alpha^* = 5.0$. The most immediately obvious property of this surface is that it is bowl-like, with a single global minimum at $\beta_0^* = 13.0697$ and $\beta_1^* = -3.0697$ as indicated on the contour plot below. The surface is symmetric because there is no prior bias. The fact that there is no functional dependence between the

thresholds β_0 and β_1 is evident in the fact that slices as a function of β_0 have identically the same shape for any fixed value of β_1 and vice versa. Notice that each of these one-dimensional sections has a shape typical of the unequal cost Gaussian problem illustrated previously in Figure 3-18. This property will be evident throughout all of the plots which follow. Within the flat region, the cost appears relatively insensitive to the exact values of the thresholds. In this region the thresholds are in the correct relative orientation to one another, i.e., $\beta_0 > \beta_1$, and at reasonable locations. On the contour plot, this "good" region corresponds to the upper left corner. As the relative positions of the thresholds switch, the cost rapidly rises.

Figure 3-23 depicts the cost for the same case, as a function of α and β_1, with $\beta_0 = \beta_0^* = 13.0697$. Again, the unique minimum of this surface is $\alpha^* = 5.0$ and $\beta_1^* = -3.0697$ as indicated on the contour plot. On this surface we can observe the following phenomenon: as α approaches $-\infty$, threshold β_1 is always selected downstream at DM B, so that the cost becomes quite sensitive to the value of β_1. Indeed, the curvature is very pronounced on this side of the surface, and the error can clearly be seen to approach the value 0.5 on either side as threshold β_1 is essentially used in isolation, and the problem decouples. As α approaches $+\infty$, threshold β_0 is always selected, so the reverse effect occurs. On this side of the contour, the surface becomes very flat, indicating insensitivity to the value of β_1.

Figure 3-24 depicts the cost for the same case, as a function of α and β_0, with $\beta_1 = \beta_1^* = -3.0697$. Now, the mirror effect of that just described may be observed. As α becomes large, threshold β_0 is always selected downstream, the problem again decouples, and the cost along a slice approaches that of a single DM operating in isolation.

Figures 3-25 - 3-27 illustrate the cost surfaces for the case of Figure 2-10(b), for which the variances at the two DMs are different, and for which prior bias is also present. Specifically, for this case $\sigma_A^2 = 50$, $\sigma_B^2 = 100$, and $p_0 = 0.7$. The most obvious change evident in Figure 3-25, which illustrates the cost as a function of β_0 and β_1, with $\alpha = \alpha^* = 7.5825$, when compared with Figure 3-22, is that the symmetry has been destroyed. This is due to the addition of a priori bias. Since

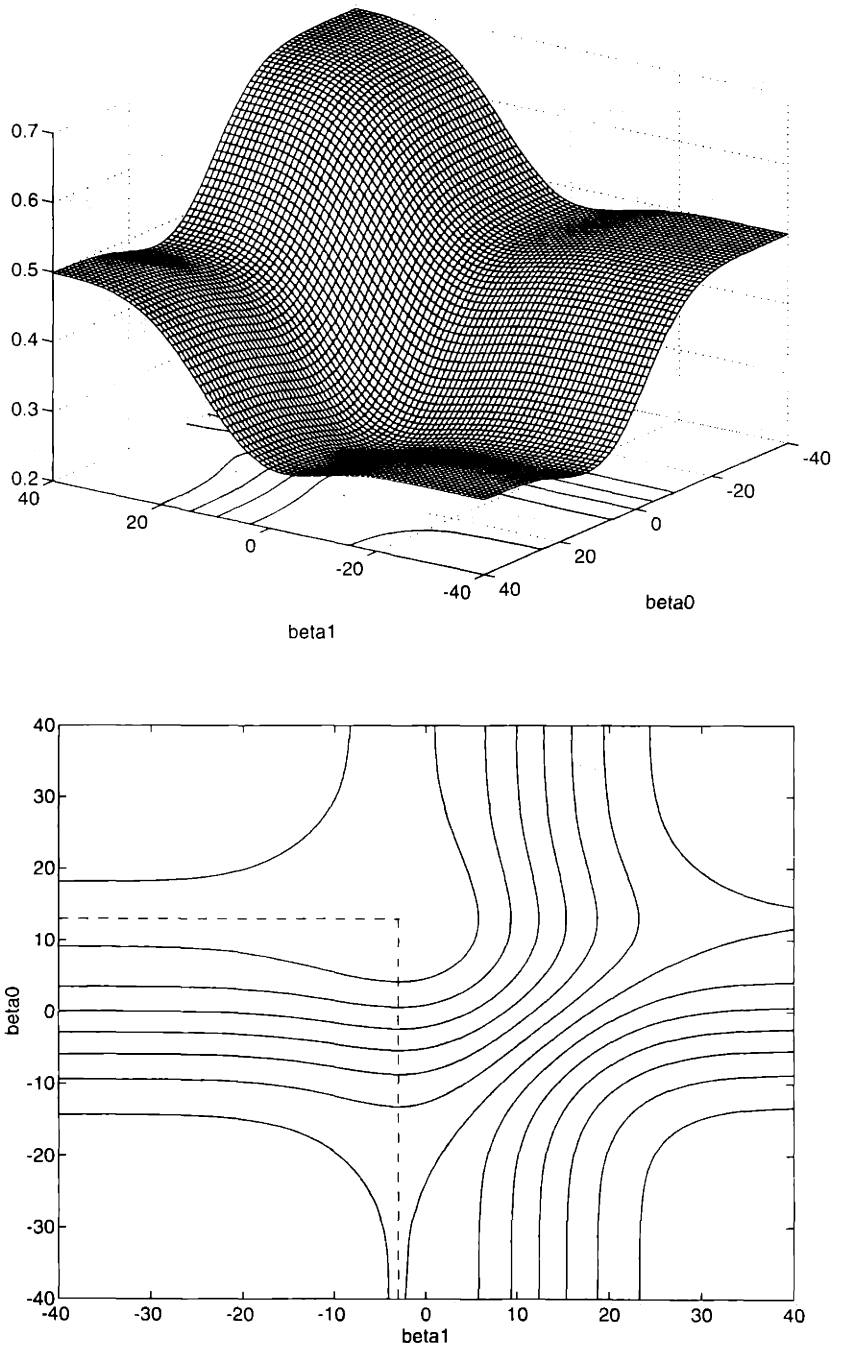


Figure 3-22: P_ϵ^{2-Tand} as a function of β_0 and β_1 with $\alpha = \alpha^* = 5.0$. Optimal values are $\beta_0^* = 13.0697$ and $\beta_1^* = -3.0697$ indicated on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = \sigma_B^2 = 100$, $p_0 = 0.5$.

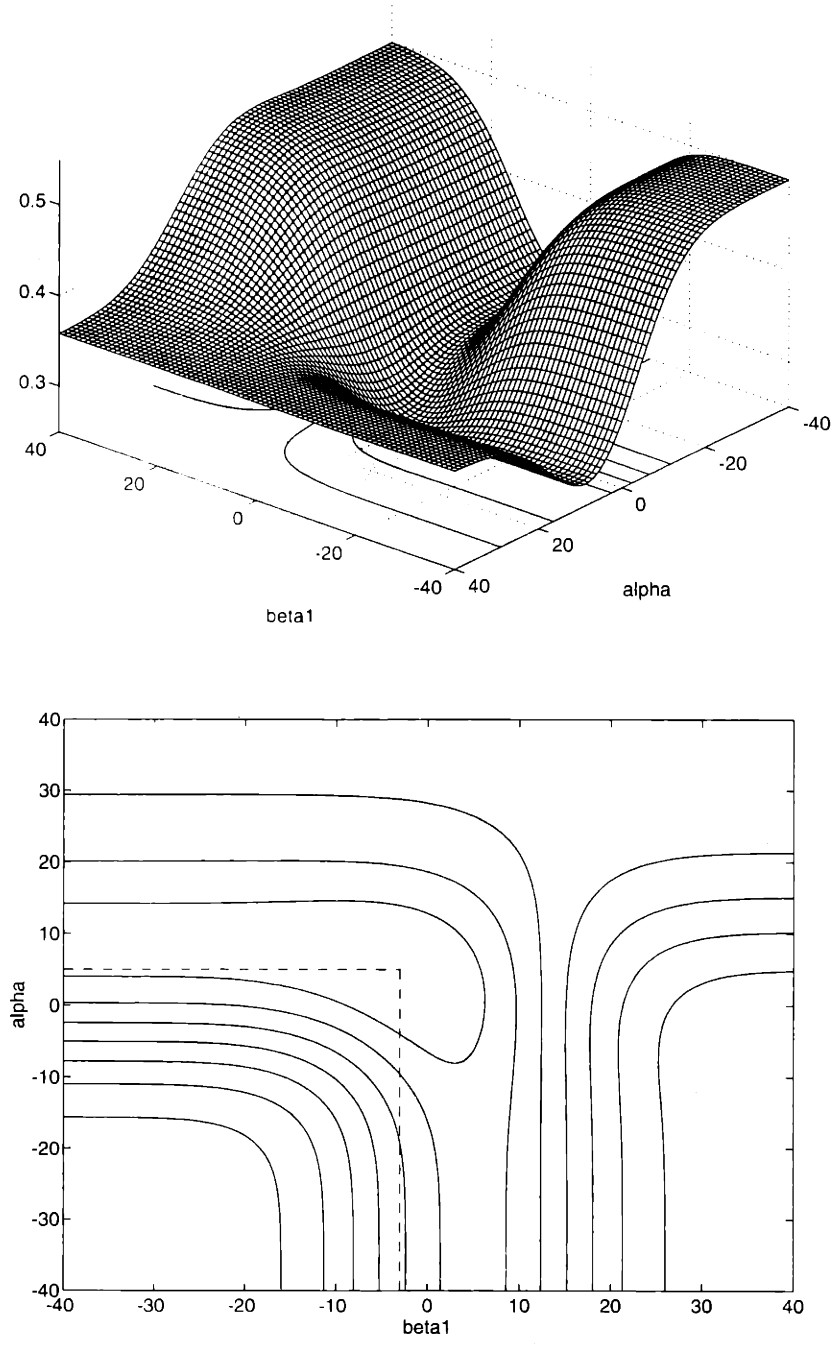


Figure 3-23: P_ϵ^{2-Tand} as a function of α and β_1 with $\beta_0 = \beta_0^* = 13.0697$. Optimal values are $\alpha^* = 5.0$ and $\beta_1 = -3.0697$ indicated on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = \sigma_B^2 = 100$, $p_0 = 0.5$.

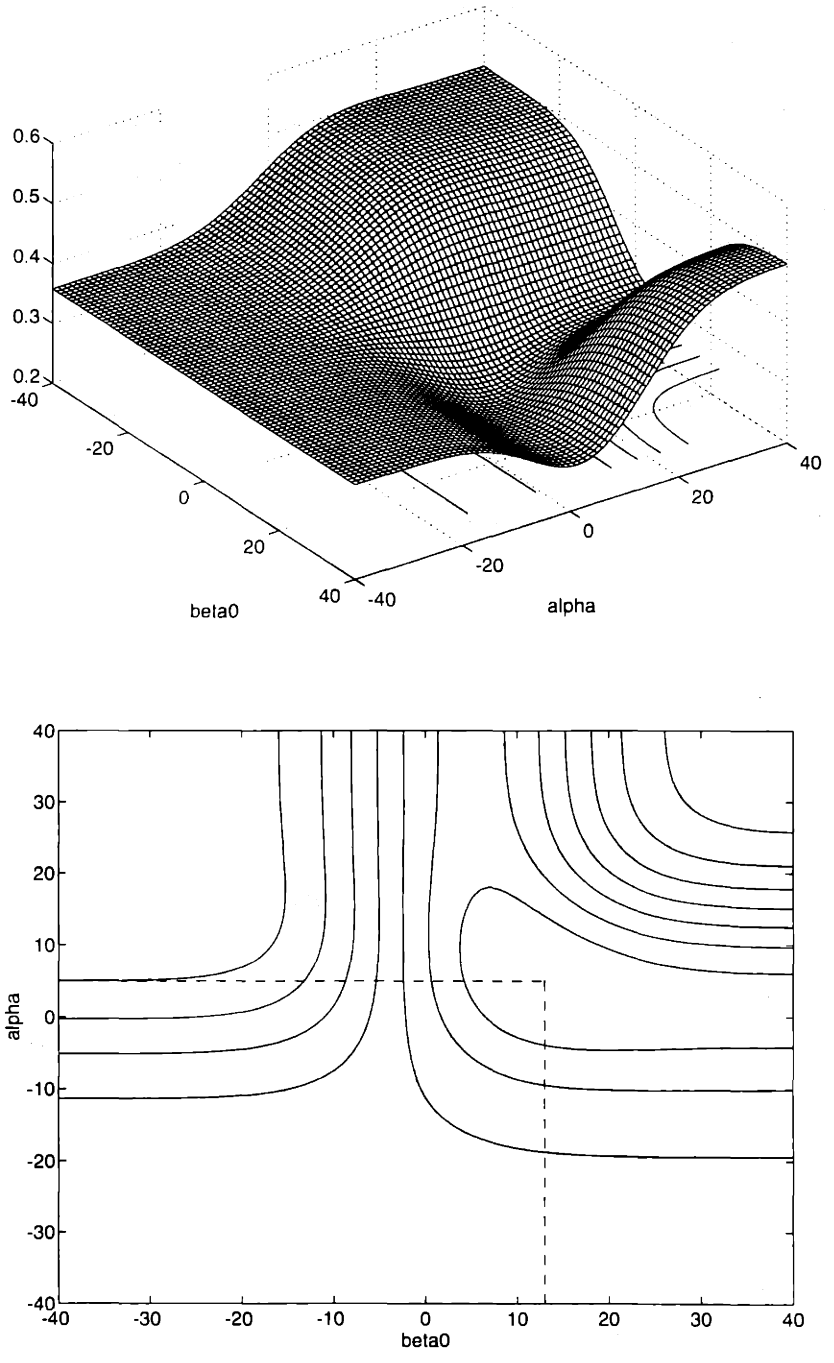


Figure 3-24: P_ϵ^{2-Tand} as a function of α and β_0 with $\beta_1 = \beta_1^* = -3.0697$. Optimal values are $\alpha^* = 5.0$ and $\beta_0^* = 13.0697$ shown on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = \sigma_B^2 = 100$, $p_0 = 0.5$.

$p_0 = 0.7$ and α is fixed at its optimal value, the probability of selecting threshold β_0 downstream is higher, and the sensitivity of the cost with respect to β_0 is expected to also be higher. This effect is clearly evident on the surface, since for a given fixed value of β_0, variation of β_1 produces little change in the team cost. Notice that the functional independence of β_0 and β_1 continues to be evident in the cross sections. Also notice that the higher variance at DM B has resulted in an even flatter "flat region", indicating that the sensitivity of the cost with respect to both parameters of B has been reduced.

In Figure 3-26 the cost as a function of α and $beta_1$, with $\beta_0 = \beta_0^* = 21.9892$ is shown. Similar comments to those made for Figure 3-23 apply here. In particular, as alpha becomes small, threshold β_1 is always selected, the problem decouples, and the section of the team cost approaches that of DM B operating in isolation with the single threshold β_1. In contrast to Figure 3-23, the cost now approaches the value $p_1 = 0.3$ as β_1 becomes large, and $p_0 = 0.7$ as β_1 becomes small. Also notice that the lower variance of DM A is evident on the surface, as a narrower bowl is apparent for slices as a function of α than in Figure 3-23.

In Figure 3-27, the cost as a function of α and β_0, with $\beta_1 = \beta_1^* = -1.5009$ is shown. Again, this surface mirrors the previous one, and illustrates the effect of decreased variance at DM A when compared with Figure 3-24.

Finally, Figures 3-28 - 3-30 correspond to the cost of the case of Figure 2-11(c) for which $\sigma_A^2 = 100$, $\sigma_B^2 = 50$, and $p_0 = 0.3$. In Figure 3-28, the cost as a function of β_0 and β_1, with $\alpha = \alpha^* = 2.9813$ is shown. The presence of the opposite prior probability has switched the sensitivities, so that now the cost is much more sensitive to parameter β_1. However, it is clear that the reduced variance at DM B has resulted in increased sensitivity of the cost with respect to the parameters at B. The flat region is must less flat, and now a distinct "pit" is visible.

In Figure 3-29, the cost as a function of α and β_1, with $\beta_0 = \beta_0^* = 5.4576$ is shown. Switching the priors has reflected the bump in the cost from its location in Figure 3-26.

In Figure 3-30, the cost as a function of α and β_0, with $\beta_1 = \beta_1^* = -2.6564$ is

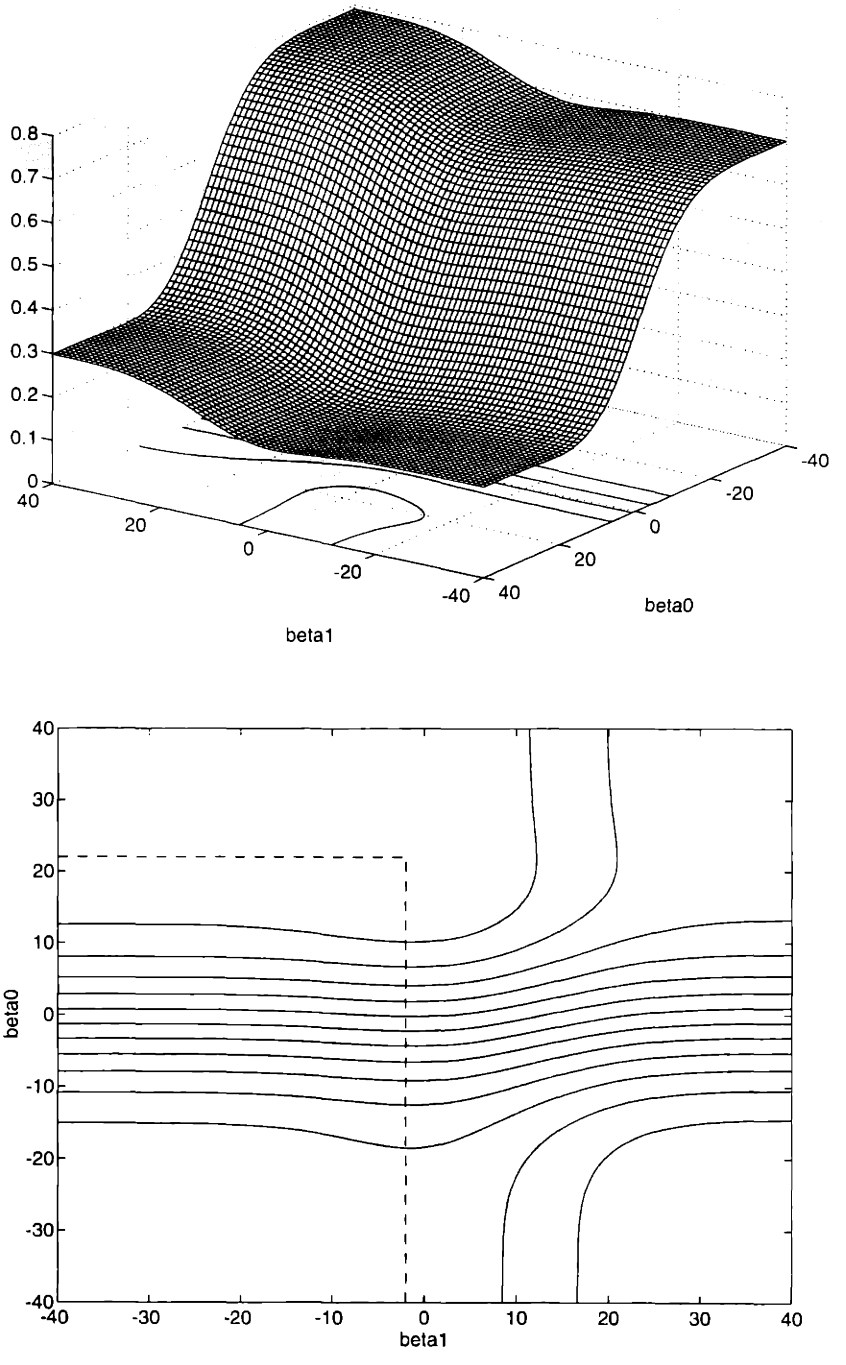


Figure 3-25: P_{ϵ}^{2-Tand} as a function of β_0 and β_1 with $\alpha = \alpha^* = 7.5825$. Optimal values are $\beta_0^* = 21.9892$ and $\beta_1^* = -1.5009$ indicated on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = 50$, $\sigma_B^2 = 100$, $p_0 = 0.7$.

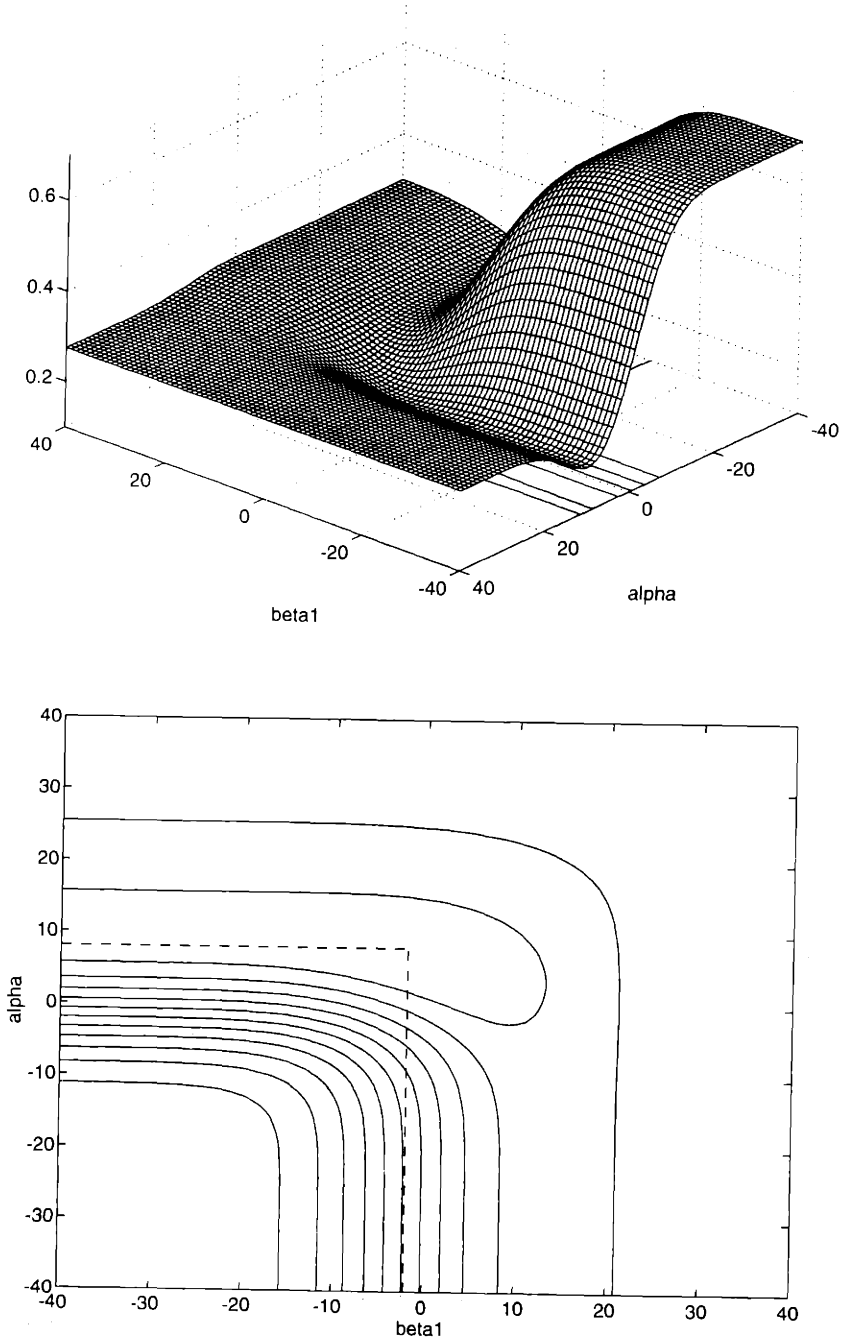


Figure 3-26: P_{ϵ}^{2-Tand} as a function of α and β_1 with $\beta_0 = \beta_0^* = 21.9892$. Optimal values are $\alpha^* = 7.5825$ and $\beta_1 = -1.5009$ indicated on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = 50$, $\sigma_B^2 = 100$, $p_0 = 0.7$.

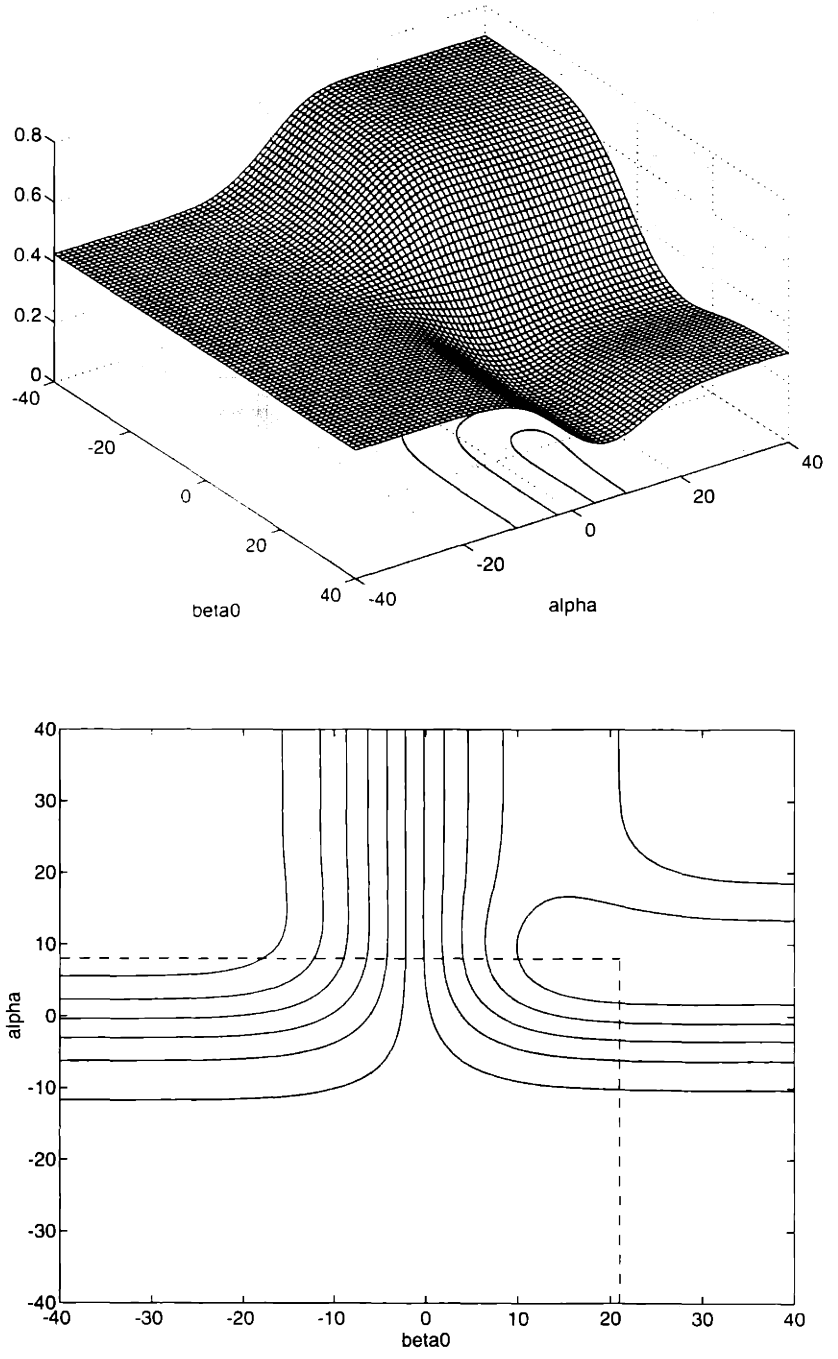


Figure 3-27: P_ϵ^{2-Tand} as a function of α and β_0 with $\beta_1 = \beta_1^* = -1.5009$. Optimal values are $\alpha^* = 7.5825$ and $\beta_0^* = 21.9892$ shown on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = 50$, $\sigma_B^2 = 100$, $p_0 = 0.7$.

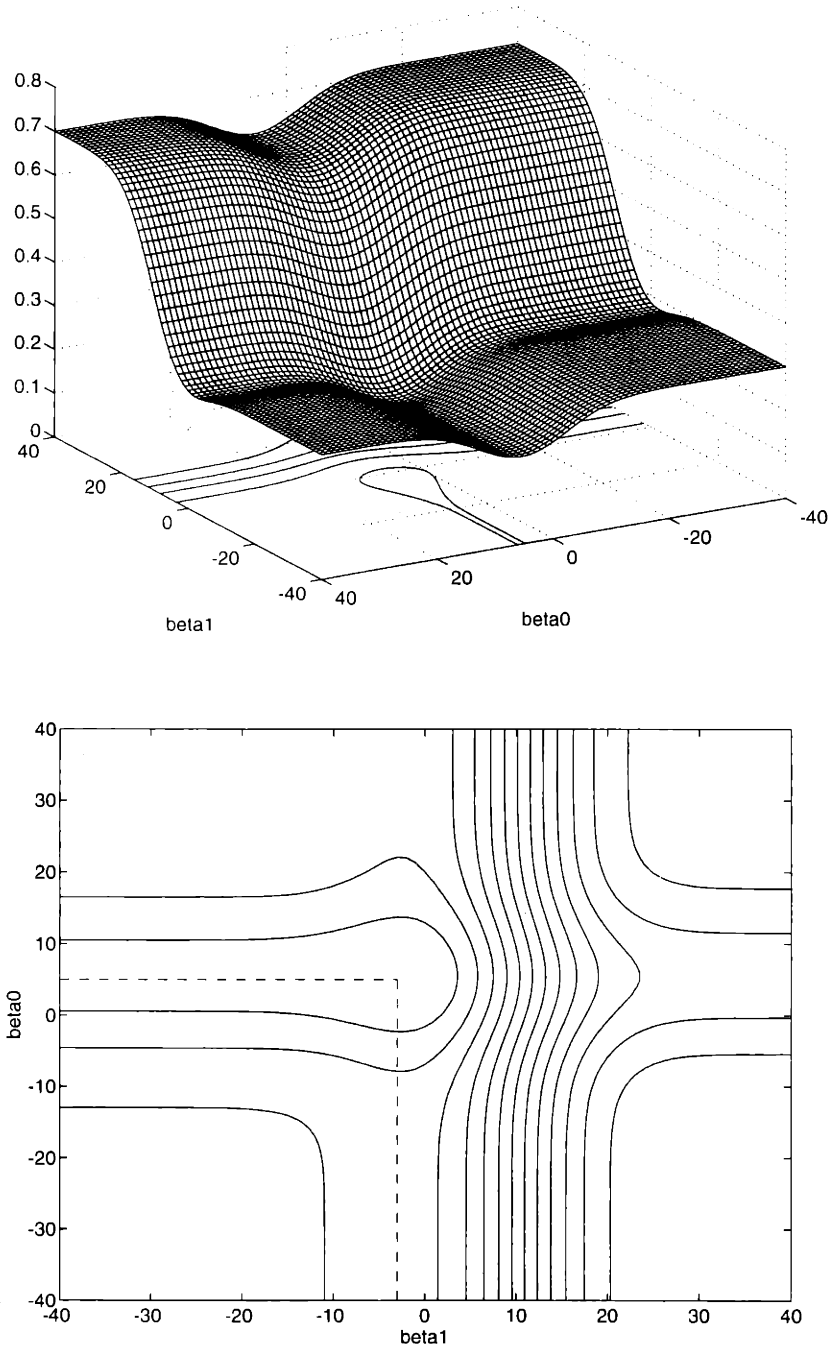


Figure 3-28: P_{ϵ}^{2-Tand} as a function of β_0 and β_1 with $\alpha = \alpha^* = 2.9813$. Optimal values are $\beta_0^* = 5.4576$ and $\beta_1^* = -2.6564$ indicated on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = 100$, $\sigma_B^2 = 50$, $p_0 = 0.3$.

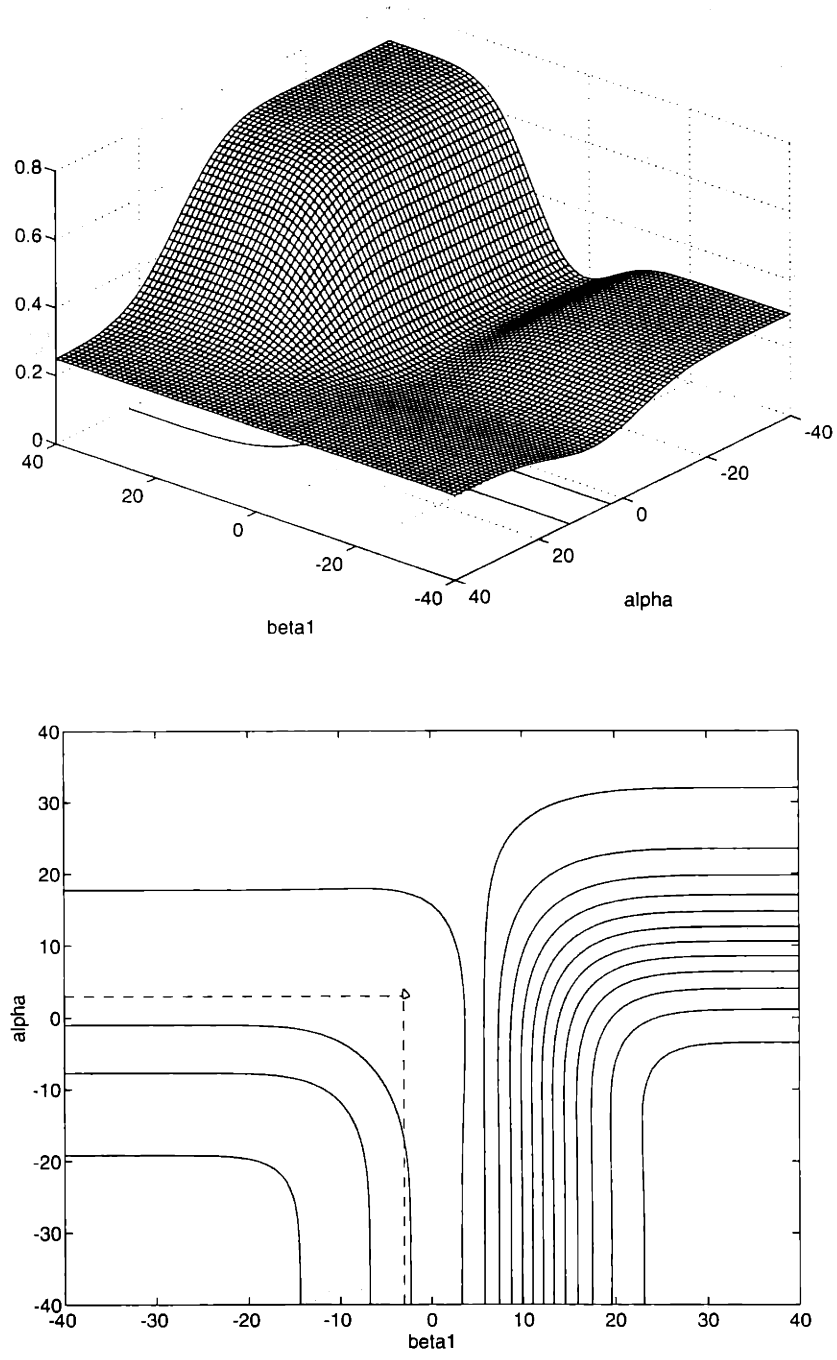


Figure 3-29: P_ϵ^{2-Tand} as a function of α and β_1 with $\beta_0 = \beta_0^* = 5.4576$. Optimal values are $\alpha^* = 2.9813$ and $\beta_1 = -2.6564$ indicated on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = 100$, $\sigma_B^2 = 50$, $p_0 = 0.3$.

shown. Again notice the reflected shape of the cost.

In addition to providing significant insight into the optimization of the thresholds for 2-Tand, the form of the surfaces suggest the following reasonable conclusions, which we further substantiate in the remainder of this report:

- Cross sections of the team cost indicate that the one-dimensional subproblems faced by each DM to update a given threshold parameter given that the other thresholds are held fixed, is of the same form as the unequal cost general Bayes problem

- Reduced sensitivity of the cost with respect to a given parameter flattens the cost as a function of that parameter, and may create difficulty in distinguishing the optimal value. Reduced sensitivity may arise because the probability of selecting a downstream parameter is low, or because of increased variance.

- Reducing the variance of a given DM narrows that bowl as a function of its parameter(s), and increases the sensitivity of the cost with respect to its parameter(s) in the vicinity of the optimum value

Properties of $P_\epsilon(\underline{\theta})$

In this section we identify some of the structure of the $P_\epsilon(\underline{\theta})$ function, and relate its properties to the properties of the underlying conditional density functions.

Assume a team of topology $Team$ containing M DMs, parameterized by N thresholds, and with the assumptions of Section 2.4.1 and Section 3.1 in place. In particular, assume $Team$ is a tree-type topology, with conditionally independent observations, and that the decision rules are parameterized by linear threshold tests. We argued in Section 3.2.2, that for tree-type topologies with conditionally independent observations, an operating point formulation of the team error probability could *always* be determined by a systematic sample space enumeration. The approach was demonstrated on the four example teams of Chapter 2, where it was shown that the cost and its derivatives possess certain structure. We now argue that this structure is

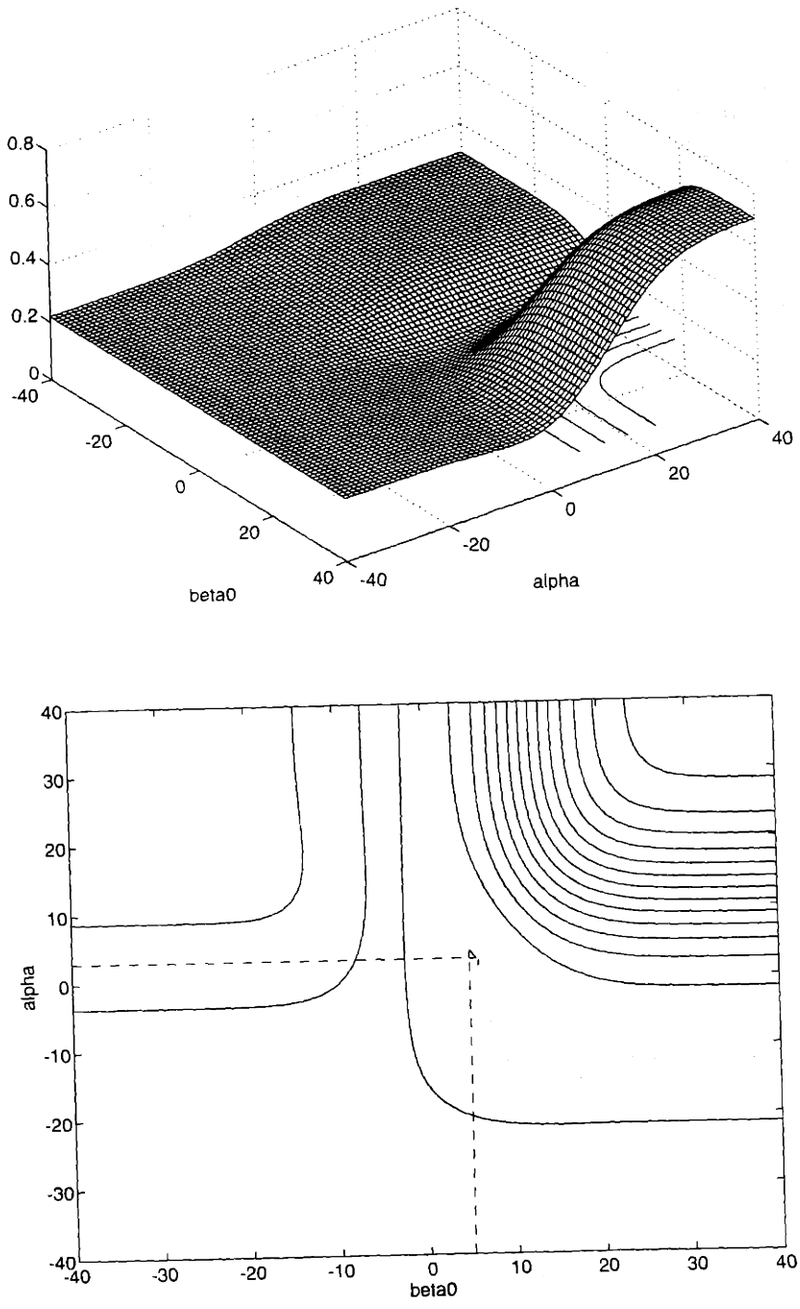


Figure 3-30: P_{ϵ}^{2-Tand} as a function of α and β_0 with $\beta_1 = \beta_1^* = -2.6564$. Optimal values are $\alpha^* = 2.9813$ and $\beta_0^* = 5.4576$ shown on contour plot below. Gaussian detection, $\mu_0 = 0$, $\mu_1 = 10$, $\sigma_A^2 = 100$, $\sigma_B^2 = 50$, $p_0 = 0.3$.

present in general. The notational scheme we adopt to make the following arguments is precise only to the degree required to make the arguments.

Construction of a parameterization of $P_\epsilon^{Team}(\underline{\theta})$ using the method of Section 3.2.2 involved tracing paths to errors in the enumeration tree, where a path was defined as an event tuple of the form $(H, u_1, \ldots, u_M)$. The team error probability resulted from summing the probability of taking these paths, and yielded a sum-of-products expression of the form, where DM M is the primary DM,

$$\begin{aligned}
P_\epsilon^{Team} &= \sum_{\{\text{paths terminating in team error}\}} \Pr(\text{following the path}) \\
&= p_0 \left[\sum_{\{\text{paths } p | U_{Team} = 1, H_0\}} \Pr(\text{follow path } p | H_0) \right] \\
&\quad + p_1 \left[\sum_{\{\text{paths } q | U_{Team} = 0, H_1\}} \Pr(\text{follow path } q | H_1) \right] \\
&= p_0 \left[\sum_{u_1,\ldots,u_{M-1}} \Pr(p = (0, u_1, \ldots, u_{M-1}, 1)\ ) \right] \\
&\quad + p_1 \left[\sum_{u_1,\ldots,u_{M-1}} \Pr(q = (1, u_1, \ldots, u_{M-1}, 0)\ ) \right] \\
&= p_0 \left[\sum_{u_1,\ldots,u_{M-1}} \prod_{i=1}^{M} F_0^{pi} \right] + p_1 \left[\sum_{u_1,\ldots,u_{M-1}} \prod_{i=1}^{M} F_1^{qi} \right]
\end{aligned} \tag{3.112}$$

where F_0^{pi} is a probability of false alarm or its complement

$$F_0^{pi} = \begin{cases} 1 - \int_{\theta_{pi}}^{\infty} p_{Y_i|H_0}(y_i|H_0)\, dy_i & \text{if } u_i = 0 \\ \int_{\theta_{pi}}^{\infty} p_{Y_i|H_0}(y_i|H_0)\, dy_i & \text{if } u_i = 1 \end{cases} \tag{3.113}$$

where θ_{pi} is the threshold employed by DM i required when following path p, and where F_1^{qi} is a probability of detection or its complement

$$F_1^{qi} = \begin{cases} 1 - \int_{\theta_{qi}}^{\infty} p_{Y_i|H_1}(y_i|H_1)\, dy_i & \text{if } u_i = 0 \\ \int_{\theta_{qi}}^{\infty} p_{Y_i|H_1}(y_i|H_1)\, dy_i & \text{if } u_i = 1 \end{cases} \tag{3.114}$$

where θ_{qi} is the threshold employed by DM i required when path q is followed. Equations (3.10), (3.18), (3.27), and (3.34) substantiate this claim. For the integral form of (3.10) given in (3.12) the claim is perhaps most transparent.

The number of product terms is determined by the number of DMs, since there is a product term on the left side of the sum (p_0 term) for each possible choice of tuple $(0, u_1, \ldots, u_{M-1}, 1)$. Each of these product terms is symmetrically represented on the right side of the sum (p_1 term) by a tuple of the form $(1, u_1, \ldots, u_{M-1}, 0)$. Thus, there are $2(2^{M-1}) = 2^M$ possible incorrect paths.

Several important properties of this parameterization are clear from the nature of the construction:

(1) Each component conditional probability (or its complement) in each product term necessarily corresponds to a different DM

(2) A term corresponding to a given threshold θ_l arises *at most once* in each product term

With the above in mind, we can now identify several properties of the team cost and its derivatives.

Proposition 3.5 (Properties of $P_\epsilon(\underline{\theta})$)

Let Team be a network of M DMs with tree-type topology and conditionally independent observations. Then, for the team cost under linear threshold parameterization by N thresholds

(a) *(Boundedness)* $0 \leq P_\epsilon^{Team}(\underline{\theta}) \leq 1$

(b) *(Differentiability) If the conditional densities* $p_{Y_i|H_j}(y_i|H_j)$ *are continuous for all* $i = 1, \ldots, M$ *and* $j = 0, 1$ *then* $P_\epsilon^{Team}(\underline{\theta})$ *is a continuously differentiable function of every threshold parameter* θ_l, $l = 1, \ldots, N$.

Proof. (a) Clear.

(b) Let θ_l correspond to DM i. Then, viewing $P_\epsilon^{Team}(\underline{\theta})$ as a function of only parameter θ_l, with the other parameters held fixed, and using the fact that a term corresponding to θ_l arises at most once in each product term, we see that the cost as

a function of θ_l can be expressed in the general form

$$P_\epsilon^{Team}(\theta_l|\theta_j, j \neq l) = A_l^i \int_{\theta_l}^{\infty} p_{Y_i|H_0}(y_i|H_0)\, dy_i + B_l^i \int_{\theta_l}^{\infty} p_{Y_i|H_1}(y_i|H_1)\, dy_i + C_l^i \quad (3.115)$$

where A_l^i, B_l^i, and C_l^i are constants when viewed as functions of θ_l. Actually, a form we will find to be more convenient for our purposes utilizes the probability of miss rather than detection, and pulls out the prior probabilities, to express this function in the alternative form

$$P_\epsilon^{Team}(\theta_l|\theta_j, j \neq l) = \lambda_0^{il} p_0 \int_{\theta_l}^{\infty} p_{Y_i|H_0}(y_i|H_0)\, dy_i + \lambda_1^{il} p_1 \int_{-\infty}^{\theta_l} p_{Y_i|H_1}(y_i|H_1)\, dy_i + D_l^i \quad (3.116)$$

where $\lambda_0^{il} = A_l^i/p_0$, $\lambda_1^{il} = -B_l^i/p_1$, and D_l^i is another constant.

In either case, dependence of the cost on parameter θ_l is through an integral bound, and is well-known to be continuous under the assumption that the densities $p_{Y_i|H_j}(y_i|H_j)$, $j = 0, 1$ are continuous [63]. Furthermore, the cost is also differentiable as a function of θ_l, with partial derivative given by

$$\frac{\partial P_\epsilon^{Team}}{\partial \theta_l}(\underline{\theta}) = -\lambda_0^{il} p_0 p_{Y_i|H_0}(\theta_l|H_0) + \lambda_1^{il} p_1 p_{Y_i|H_1}(\theta_l|H_1) \quad (3.117)$$

which is also continuous if the conditional densities are continuous.

■

In the process of proving Proposition 3.5(b), it was shown that the team cost, when viewed as a function of one parameter at a time, has the same form as the unequal cost Bayes formulation, modulo a possible constant. As an example, consider the cost for 2-Tand which was derived earlier in the chapter and found to be of the form

$$\begin{aligned} P_\epsilon^{2-Tand}(\alpha, \beta_0, \beta_1) \;=\; & p_0 \Big[\Big(1 - \textstyle\int_\alpha^{+\infty} p_{Y_A|H_0}(y_A|H_0) dy_A\Big) \int_{\beta_0}^{+\infty} p_{Y_B|H_0}(y_B|H_0) dy_B \\ & + \textstyle\int_\alpha^{+\infty} p_{Y_A|H_0}(y_A|H_0) dy_A \int_{\beta_1}^{+\infty} p_{Y_B|H_0}(y_B|H_0) dy_B \Big] \\ & + p_1 \Big[\Big(1 - \textstyle\int_\alpha^{+\infty} p_{Y_A|H_1}(y_A|H_1) dy_A\Big) \Big(1 - \textstyle\int_{\beta_0}^{+\infty} p_{Y_B|H_1}(y_B|H_1) dy_B\Big) \\ & + \textstyle\int_\alpha^{+\infty} p_{Y_A|H_1}(y_A|H_1) dy_A \Big(1 - \textstyle\int_{\beta_1}^{+\infty} p_{Y_B|H_1}(y_B|H_1) dy_B\Big) \Big] \end{aligned} \quad (3.118)$$

We can express

$$\begin{aligned} P_\epsilon^{2-Tand}(\alpha|\beta_0,\beta_1) &= [P_F^{B1} - P_F^{B0}]p_0 \int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A \\ &\quad -[P_D^{B1} - P_D^{B0}]p_1 \int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A \\ &\quad +[p_0 P_F^{B0} + p_1(1 - P_D^{B0})] \end{aligned} \tag{3.119}$$

in the form

$$\begin{aligned} P_\epsilon^{2-Tand}(\alpha|\beta_0,\beta_1) &= A^A \int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A \\ &\quad +B^A \int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A \\ &\quad +C^A \end{aligned} \tag{3.120}$$

where we can make the identifications

$$\begin{aligned} A^A &= [P_F^{B1} - P_F^{B0}]p_0 \\ B^A &= -[P_D^{B1} - P_D^{B0}]p_1 \\ C^A &= [p_0 P_F^{B0} + p_1(1 - P_D^{B0})] \end{aligned} \tag{3.121}$$

In the alternative form suggested above, the equation becomes

$$\begin{aligned} P_\epsilon^{2-Tand}(\alpha|\beta_0,\beta_1) &= \lambda_0^A p_0 \int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A \\ &\quad +\lambda_1^A p_1 \int_{-\infty}^\alpha p_{Y_A|H_1}(y_A|H_1)\,dy_A \\ &\quad +D^A \end{aligned} \tag{3.122}$$

where

$$\begin{aligned} \lambda_0^A &= [P_F^{B1} - P_F^{B0}] \\ \lambda_1^A &= [P_D^{B1} - P_D^{B0}] \\ D^A &= [p_1 + p_0 P_F^{B0} - p_1 P_D^{B1}] \end{aligned} \tag{3.123}$$

Similarly for the other components we may write

$$\begin{aligned} P_\epsilon^{2-Tand}(\beta_0|\alpha,\beta_1) &= \lambda_0^{B0} p_0 \int_{\beta_0}^{\infty} p_{Y_B|H_0}(y_B|H_0)\, dy_B \\ &\quad +\lambda_1^{B0} p_1 \int_{-\infty}^{\beta_0} p_{Y_B|H_1}(y_B|H_1)\, dy_B \\ &\quad +D^{B0} \end{aligned} \tag{3.124}$$

where we identify

$$\begin{aligned} \lambda_0^{B0} &= [1 - P_F^A] \\ \lambda_1^{B0} &= [1 - P_D^A] \\ D^{B0} &= [p_0 P_F^A P_F^{B1} + p_1(1 - P_D^A) + p_1 P_D^A (1 - P_D^{B1})] \end{aligned} \tag{3.125}$$

Finally, for component β_1 we can write

$$\begin{aligned} P_\epsilon^{2-Tand}(\beta_1|\alpha,\beta_0) &= \lambda_0^{B1} p_0 \int_{\beta_1}^{\infty} p_{Y_B|H_0}(y_B|H_0)\, dy_B \\ &\quad +\lambda_1^{B1} p_1 \int_{-\infty}^{\beta_1} p_{Y_B|H_1}(y_B|H_1)\, dy_B \\ &\quad +D^{B1} \end{aligned} \tag{3.126}$$

where we identify

$$\begin{aligned} \lambda_0^{B1} &= P_F^A \\ \lambda_1^{B1} &= P_D^A \\ D^{B1} &= [p_0(1 - P_F^A) P_F^{B0} + p_1(1 - P_D^A)(1 - P_D^{B0})] \end{aligned} \tag{3.127}$$

Properties of $\nabla P_\epsilon(\underline{\theta})$

The results of this section are critical to the development in the remainder of this report. We examine in detail the form of the gradient for the team problem, and find that, under the conditions of tree-type topologies and conditionally independent observations, we may identify a common structure in each partial derivative.

First we summarize a result which arose in the course of proving Proposition 3.5(b).

Proposition 3.6 (Structure of $\partial P_\epsilon^{Team}(\underline{\theta})/\partial\theta_l$)
Let Team be a network of M DMs with tree-type topology and conditionally independent observations and let $P_\epsilon^{Team}(\underline{\theta})$ denote a linear threshold parameterization of the team probability of error containing N thresholds. Then, if θ_l is a threshold parameter corresponding to DM i,

$$\frac{\partial P_\epsilon^{Team}}{\partial\theta_l}(\underline{\theta}) = -[\lambda_0^{il}(\underline{\theta})]p_0 p_{Y_i|H_0}(\theta_l|H_0) + [\lambda_1^{il}(\underline{\theta})]p_1 p_{Y_i|H_1}(\theta_l|H_1) \tag{3.128}$$

where $\lambda_0^{il}(\underline{\theta})$ and $\lambda_1^{il}(\underline{\theta})$ are coupling costs with analytic form determined by the topology Team and with functional dependence on the operating points corresponding to components $\theta_j, j \neq l$.

That the partial derivatives take this form is also clear when it is considered that a value of θ satisfying the necessary conditions for optimality satisfies the corresponding LRT with equality, so that these forms were already evident in the LRTs of Chapter 2. Thus, the constructive argument going through sequential sample space enumeration, has provided an alternative method of proof of the fact that for tree-type topologies with conditionally independent observations, and team decision made by a primary DM to minimize the team probability of error, the necessary conditions for optimality may be expressed as a coupled set of LRTs.

As an example, equations (3.14)-(3.16) giving the partial derivatives of P_ϵ^{2-Tand} with respect to the threshold parameters are of the form

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial\alpha} &= -\lambda_0^A p_0 p_{Y_A|H_0}(\alpha|H_0) + \lambda_1^A p_1 p_{Y_A|H_1}(\alpha|H_1) \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial\beta_0} &= -\lambda_0^{B0} p_0 p_{Y_B|H_0}(\beta_0|H_0) + \lambda_1^{B0} p_1 p_{Y_B|H_1}(\beta_0|H_1)
\end{aligned}$$

$$\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_1} = -\lambda_0^{B1} p_0 p_{Y_B|H_0}(\beta_1|H_0) + \lambda_1^{B1} p_1 p_{Y_B|H_1}(\beta_1|H_1) \qquad (3.129)$$

where

$$\begin{aligned}
\lambda_0^A(\beta_0,\beta_1) &= [P_F^{B1} - P_F^{B0}] \\
&= \left[\int_{\beta_1}^{\infty} p_{Y_B|H_0}(y_B|H_0)\,dy_B - \int_{\beta_0}^{\infty} p_{Y_B|H_0}(y_B|H_0)\,dy_B\right] \\
\lambda_1^A(\beta_0,\beta_1) &= [P_D^{B1} - P_D^{B0}] \\
&= \left[\int_{\beta_1}^{\infty} p_{Y_B|H_1}(y_B|H_1)\,dy_B - \int_{\beta_0}^{\infty} p_{Y_B|H_1}(y_B|H_1)\,dy_B\right]
\end{aligned}$$

$$\begin{aligned}
\lambda_0^{B0}(\alpha) &= [1 - P_F^A] \\
&= \left[1 - \int_{\alpha}^{\infty} p_{Y_A|H_0}(y_A|H_0)\,dy_A\right] \\
\lambda_1^{B0}(\alpha) &= [1 - P_D^A] \\
&= \left[1 - \int_{\alpha}^{\infty} p_{Y_A|H_1}(y_A|H_1)\,dy_A\right]
\end{aligned}$$

$$\begin{aligned}
\lambda_0^{B1}(\alpha) &= P_F^A \\
&= \int_{\alpha}^{\infty} p_{Y_A|H_0}(y_A|H_0)\,dy_A \\
\lambda_1^{B1}(\alpha) &= P_D^A \\
&= \int_{\alpha}^{\infty} p_{Y_A|H_1}(y_A|H_1)\,dy_A
\end{aligned} \qquad (3.130)$$

Similarly, for 3-Vee the coupling costs in (3.20)-(3.25) are of the form

$$\lambda_0^A(\beta,\xi_{00},\xi_{01},\xi_{10},\xi_{11}) = [(1-P_F^B)(P_F^{C(10)} - P_F^{C(00)}) + P_F^B(P_F^{C(11)} - P_F^{C(01)})] \qquad (3.131)$$

$$\begin{aligned}
&= \left[\left(1 - \int_{\beta}^{\infty} p_{Y_B|H_0}(y_B|H_0)\,dy_B\right)\left(\int_{\xi_{10}}^{\infty} p_{Y_C|H_0}(y_C|H_0)\,dy_C - \int_{\xi_{00}}^{\infty} p_{Y_C|H_0}(y_C|H_0)\,dy_C\right)\right. \\
&\quad \left. + \int_{\beta}^{\infty} p_{Y_B|H_0}(y_B|H_0)\,dy_B\left(\int_{\xi_{11}}^{\infty} p_{Y_C|H_0}(y_C|H_0)\,dy_C - \int_{\xi_{01}}^{\infty} p_{Y_C|H_0}(y_C|H_0)\,dy_C\right)\right]
\end{aligned}$$

$$\lambda_1^A(\beta,\xi_{00},\xi_{01},\xi_{10},\xi_{11}) = [(1-P_D^B)(P_D^{C(10)} - P_D^{C(00)}) + P_D^B(P_D^{C(11)} - P_D^{C(01)})] \qquad (3.132)$$

$$\begin{aligned}
&= \left[\left(1 - \int_{\beta}^{\infty} p_{Y_B|H_1}(y_B|H_1)\,dy_B\right)\left(\int_{\xi_{10}}^{\infty} p_{Y_C|H_1}(y_C|H_1)\,dy_C - \int_{\xi_{00}}^{\infty} p_{Y_C|H_1}(y_C|H_1)\,dy_C\right)\right. \\
&\quad \left. + \int_{\beta}^{\infty} p_{Y_B|H_1}(y_B|H_1)\,dy_B\left(\int_{\xi_{11}}^{\infty} p_{Y_C|H_1}(y_C|H_1)\,dy_C - \int_{\xi_{01}}^{\infty} p_{Y_C|H_1}(y_C|H_1)\,dy_C\right)\right]
\end{aligned}$$

$$\lambda_0^B(\alpha,\xi_{00},\xi_{01},\xi_{10},\xi_{11}) = [(1-P_F^A)(P_F^{C(01)}-P_F^{C(00)})+P_F^A(P_F^{C(11)}-P_F^{C(10)})] \quad (3.133)$$

$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A\right)\left(\int_{\xi_{01}}^\infty p_{Y_C|H_0}(y_C|H_0)\,dy_C-\int_{\xi_{00}}^\infty p_{Y_C|H_0}(y_C|H_0)\,dy_C\right)\right.$$
$$\left.+\int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A\left(\int_{\xi_{11}}^\infty p_{Y_C|H_0}(y_C|H_0)\,dy_C-\int_{\xi_{10}}^\infty p_{Y_C|H_0}(y_C|H_0)\,dy_C\right)\right]$$

$$\lambda_1^B(\alpha,\xi_{00},\xi_{01},\xi_{10},\xi_{11}) = [(1-P_D^A)(P_D^{C(01)}-P_D^{C(00)})+P_D^A(P_D^{C(11)}-P_D^{C(10)})] \quad (3.134)$$

$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_1}(y_B|H_1)\,dy_A\right)\left(\int_{\xi_{01}}^\infty p_{Y_C|H_1}(y_C|H_1)\,dy_C-\int_{\xi_{00}}^\infty p_{Y_C|H_1}(y_C|H_1)\,dy_C\right)\right.$$
$$\left.+\int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A\left(\int_{\xi_{11}}^\infty p_{Y_C|H_1}(y_C|H_1)\,dy_C-\int_{\xi_{10}}^\infty p_{Y_C|H_1}(y_C|H_1)\,dy_C\right)\right]$$

$$\lambda_0^{C00}(\alpha,\beta) = [(1-P_F^A)(1-P_F^B)]$$
$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A\right)\left(1-\int_\beta^\infty p_{Y_B}|H_0(y_B|H_0)\,dy_B\right)\right]$$
$$\lambda_1^{C00}(\alpha,\beta) = [(1-P_D^A)(1-P_D^B)]$$
$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A\right)\left(1-\int_\beta^\infty p_{Y_B}|H_1(y_B|H_1)\,dy_B\right)\right]$$

$$\lambda_0^{C01}(\alpha,\beta) = [(1-P_F^A)P_F^B]$$
$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A\right)\int_\beta^\infty p_{Y_B}|H_0(y_B|H_0)\,dy_B\right]$$
$$\lambda_1^{C01}(\alpha,\beta) = [(1-P_D^A)P_D^B]$$
$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A\right)\int_\beta^\infty p_{Y_B}|H_1(y_B|H_1)\,dy_B\right]$$

$$\lambda_0^{C10}(\alpha,\beta) = [P_F^A(1-P_F^B)]$$
$$= \left[\int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A\left(1-\int_\beta^\infty p_{Y_B}|H_0(y_B|H_0)\,dy_B\right)\right]$$
$$\lambda_1^{C10}(\alpha,\beta) = [P_D^A(1-P_D^B)]$$
$$= \left[\int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A\left(1-\int_\beta^\infty p_{Y_B}|H_1(y_B|H_1)\,dy_B\right)\right]$$

$$\lambda_0^{C11}(\alpha,\beta) = [P_F^A P_F^B]$$
$$= \left[\int_\alpha^\infty p_{Y_A|H_0}(y_A|H_0)\,dy_A\int_\beta^\infty p_{Y_B}|H_0(y_B|H_0)\,dy_B\right]$$
$$\lambda_1^{C11}(\alpha,\beta) = [(1-P_D^A)(1-P_D^B)]$$
$$= \left[\left(1-\int_\alpha^\infty p_{Y_A|H_1}(y_A|H_1)\,dy_A\right)\left(1-\int_\beta^\infty p_{Y_B}|H_1(y_B|H_1)\,dy_B\right)\right] \quad (3.135)$$

We refer to the coefficients $\lambda_0^{il},\lambda_1^{il}$ as coupling costs since these coefficients play

the role of the unequal costs on each type of error in the general Bayes risk problem. The following properties of the coupling costs are immediate consequences of our construction of $P_\epsilon(\underline{\theta})$.

Proposition 3.7 (Properties of the Coupling Costs)

Let $\lambda_0^{il}(\underline{\theta})$, $\lambda_1^{il}(\underline{\theta})$ be the coupling costs associated with parameter θ_l corresponding to DM i, and let F_0^{pi}, F_1^{qi} be defined as in (3.113) and (3.114), respectively. Then

(a) *(Sum of Products)*

$$\lambda_0^{il}(\underline{\theta}) = \sum_{\{p|H=H_0, U_{Team}=1, \theta_l=\theta_{pi}\}} (-1)^{s_p} \prod_{\substack{1\le j\le M \\ j\neq i}} F_0^{pi} \tag{3.136}$$

$$\lambda_1^{il}(\underline{\theta}) = -\sum_{\{q|H=H_1, U_{Team}=0, \theta_l=\theta_{qi}\}} (-1)^{s_q} \prod_{\substack{1\le j\le M \\ j\neq i}} F_1^{qi} \tag{3.137}$$

where the sign variable s_p is defined by

$$s_p = \begin{cases} 0 & \text{if } \theta_l \text{ enters as } \int_{\theta_l}^{\infty} p_{Y_i|H_0}(y_i|H_0)\, dy_i \\ 1 & \text{if } \theta_l \text{ enters as } \left(1 - \int_{\theta_l}^{\infty} p_{Y_i|H_0}(y_i|H_0)\, dy_i\right) \end{cases} \tag{3.138}$$

and s_q is defined analogously.

(b) *(Bounded) There exist constants $K_0^{il} > 0$, $K_1^{il} > 0$ $\forall i, l$ such that*

$$\begin{aligned} |\lambda_0^{il}(\underline{\theta})| &\le K_0^{il} \quad \forall \underline{\theta} \in \Re^N \\ |\lambda_1^{il}(\underline{\theta})| &\le K_1^{il} \quad \forall \underline{\theta} \in \Re^N \end{aligned} \tag{3.139}$$

Proof. Part (a) follows directly from differentiation of (3.116) while (b) follows from the fact that each sum has at most 2^{M-1} terms, each of which is a product of conditional probabilities which are bounded between zero and one.

■

The proposition indicates that each cost is expressible as a sum-of-products with

the following properties:

(1) Each component conditional probability (or its complement) in each product term corresponds to a different DM

(2) The costs $\lambda_0^{il}, \lambda_1^{il}$ have no dependence on θ_l or any other thresholds corresponding to DM i

These properties will prove critical in the development of one class of network training algorithm which obtains local estimates of these costs. Since each component of each product term corresponds to a different DM, we will be able to show that combining locally generated operating point estimates still results in unbiased estimates of the local coupling costs, a key property in proving convergence of this class of network training algorithm.

We summarize here the coupling costs for each of the four networks of Chapter 2, as we will need them at a later point.

2-Tand:

$$\lambda_0^A = [P_F^{B1} - P_F^{B0}], \quad \lambda_1^A = [P_D^{B1} - P_D^{B0}] \tag{3.140}$$

$$\lambda_0^{B0} = [1 - P_F^A], \quad \lambda_1^{B0} = [1 - P_D^A] \tag{3.141}$$

$$\lambda_0^{B1} = P_F^A, \quad \lambda_1^{B1} = P_D^A \tag{3.142}$$

3-Vee:

$$\begin{aligned} \lambda_0^A &= [(1 - P_F^B)(P_F^{C(10)} - P_F^{C(00)}) \\ &\quad + P_F^B(P_F^{C(11)} - P_F^{C(01)})] \\ \lambda_1^A &= [(1 - P_D^B)(P_D^{C(10)} - P_D^{C(00)}) \\ &\quad + P_D^B(P_D^{C(11)} - P_D^{C(01)})] \end{aligned} \tag{3.143}$$

$$\begin{aligned} \lambda_0^B &= [(1 - P_F^A)(P_F^{C(01)} - P_F^{C(00)}) \\ &\quad + P_F^A(P_F^{C(11)} - P_F^{C(10)})] \\ \lambda_1^B &= [(1 - P_D^A)(P_D^{C(01)} - P_D^{C(00)}) \end{aligned}$$

$$+P_D^A(P_D^{C(11)} - P_D^{C(10)})] \tag{3.144}$$

$$\lambda_0^{C(00)} = [(1-P_F^A)(1-P_F^B)], \;\; \lambda_1^{C(00)} = [(1-P_D^A)(1-P_D^B)] \tag{3.145}$$

$$\lambda_0^{C(01)} = [(1-P_F^A)P_F^B], \;\; \lambda_1^{C(01)} = [(1-P_D^A)P_D^B] \tag{3.146}$$

$$\lambda_0^{C(10)} = [P_F^A(1-P_F^B)], \;\; \lambda_1^{C(10)} = [P_D^A(1-P_D^B)] \tag{3.147}$$

$$\lambda_0^{C(11)} = [P_F^A P_F^B], \;\; \lambda_1^{C(11)} = [P_D^A P_D^B] \tag{3.148}$$

3-Tand:

$$\begin{aligned} \lambda_0^A &= [(P_F^{B1} - P_F^{B0})(P_F^{C1} - P_F^{C0})] \\ \lambda_1^A &= [(P_D^{B1} - P_D^{B0})(P_D^{C1} - P_D^{C0})] \end{aligned} \tag{3.149}$$

$$\begin{aligned} \lambda_0^{B0} &= [(1-P_F^A)(P_F^{C1} - P_F^{C0})] \\ \lambda_1^{B0} &= [(1-P_D^A)(P_D^{C1} - P_D^{C0})] \end{aligned} \tag{3.150}$$

$$\begin{aligned} \lambda_0^{B1} &= [P_F^A(P_F^{C1} - P_F^{C0})] \\ \lambda_1^{B1} &= [P_D^A(P_D^{C1} - P_D^{C0})] \end{aligned} \tag{3.151}$$

$$\begin{aligned} \lambda_0^{C0} &= [(1-P_F^A)(1-P_F^{B0}) + P_F^A(1-P_F^{B1})] \\ \lambda_1^{C0} &= [(1-P_D^A)(1-P_D^{B0}) + P_D^A(1-P_D^{B1})] \end{aligned} \tag{3.152}$$

$$\begin{aligned} \lambda_0^{C1} &= [(1-P_F^A)P_F^{B0} + P_F^A P_F^{B1}] \\ \lambda_1^{C1} &= [(1-P_D^A)P_D^{B0} + P_D^A P_D^{B1}] \end{aligned} \tag{3.153}$$

4-Asym:

$$
\begin{aligned}
\lambda_0^A &= [\,-((1-P_F^{B0})(1-P_F^C)P_F^{D(00)} + (1-P_F^{B0})P_F^C P_F^{D(01)} \\
&\quad + P_F^{B0}(1-P_F^C)P_F^{D(10)} + P_F^{B0}P_F^C P_F^{D(11)}) \\
&\quad + ((1-P_F^{B1})(1-P_F^C)P_F^{D(00)} + (1-P_F^{B1})P_F^C P_F^{D(01)} \\
&\quad + P_F^{B1}(1-P_F^C)P_F^{D(10)} + P_F^{B1}P_F^C P_F^{D(11)})\,] \\
\lambda_1^A &= [\,((1-P_D^{B0})(1-P_D^C)(1-P_D^{D(00)}) + (1-P_D^{B0})P_D^C(1-P_D^{D(01)}) \\
&\quad + P_D^{B0}(1-P_D^C)(1-P_D^{D(10)}) + P_D^{B0}P_D^C(1-P_D^{D(11)})) \\
&\quad - ((1-P_D^{B1})(1-P_D^C)(1-P_D^{D(00)}) + (1-P_D^{B1})P_D^C(1-P_D^{D(01)}) \\
&\quad + P_D^{B1}(1-P_D^C)(1-P_D^{D(10)}) + P_D^{B1}P_D^C(1-P_D^{D(11)}))\,]
\end{aligned}
\tag{3.154}
$$

$$
\begin{aligned}
\lambda_0^{B0} &= [\,(1-P_F^A)\,(-(1-P_F^C)P_F^{D(00)} - P_F^C P_F^{D(01)} \\
&\quad + (1-P_F^C)P_F^{D(10)} + P_F^C P_F^{D(11)})\,] \\
\lambda_1^{B0} &= [\,(1-P_D^A)\,((1-P_D^C)(1-P_D^{D(00)}) + P_D^C(1-P_D^{D(01)}) \\
&\quad - (1-P_D^C)(1-P_D^{D(10)}) - P_D^C(1-P_D^{D(11)}))\,]
\end{aligned}
\tag{3.155}
$$

$$
\begin{aligned}
\lambda_0^{B1} &= [\,P_F^A\,(-(1-P_F^C)P_F^{D(00)} - P_F^C P_F^{D(01)} \\
&\quad + (1-P_F^C)P_F^{D(10)} + P_F^C P_F^{D(11)})\,] \\
\lambda_1^{B1} &= [\,P_D^A\,((1-P_D^C)(1-P_D^{D(00)}) + P_D^C(1-P_D^{D(01)}) \\
&\quad - (1-P_D^C)(1-P_D^{D(10)}) - P_D^C(1-P_D^{D(11)}))\,]
\end{aligned}
\tag{3.156}
$$

$$
\begin{aligned}
\lambda_0^C &= [\,(1-P_F^A)\,(-(1-P_F^{B0})P_F^{D(00)} + (1-P_F^{B0})P_F^{D(01)} \\
&\quad - P_F^{B0}P_F^{D(10)} + P_F^{B0}P_F^{D(11)}) \\
&\quad + P_F^A\,(-(1-P_F^{B1})P_F^{D(00)} + (1-P_F^{B1})P_F^{D(01)} \\
&\quad - P_F^{B1}P_F^{D(10)} + P_F^{B1}P_F^{D(11)})\,] \\
\lambda_1^C &= [\,(1-P_D^A)\,((1-P_D^{B0})(1-P_D^{D(00)}) - (1-P_D^{B0})(1-P_D^{D(01)})
\end{aligned}
$$

$$
\begin{aligned}
&+P_D^{B0}(1-P_D^{D(10)}) - P_D^{B0}(1-P_D^{D(11)}))\\
&+P_D^{A}\left((1-P_D^{B1})(1-P_D^{D(00)}) - (1-P_D^{B1})(1-P_D^{D(01)})\right.\\
&\left.+P_D^{B1}(1-P_D^{D(10)}) - P_D^{B1}(1-P_D^{D(11)})\right]
\end{aligned}
\tag{3.157}
$$

$$
\begin{aligned}
\lambda_0^{D(00)} &= [\,(1-P_F^A)(1-P_F^{B0})(1-P_F^C) + P_F^A(1-P_F^{B1})(1-P_F^C)\,]\\
\lambda_1^{D(00)} &= [\,(1-P_D^A)(1-P_D^{B0})(1-P_D^C) + P_D^A(1-P_D^{B1})(1-P_D^C)\,]
\end{aligned}
\tag{3.158}
$$

$$
\begin{aligned}
\lambda_0^{D(01)} &= [\,(1-P_F^A)(1-P_F^{B0})P_F^C + P_F^A(1-P_F^{B1})P_F^C\,]\\
\lambda_1^{D(01)} &= [\,(1-P_D^A)(1-P_D^{B0})P_D^C + P_D^A(1-P_D^{B1})P_D^C\,]
\end{aligned}
\tag{3.159}
$$

$$
\begin{aligned}
\lambda_0^{D(10)} &= [\,(1-P_F^A)P_F^{B0}(1-P_F^C) + P_F^A P_F^{B1}(1-P_F^C)\,]\\
\lambda_1^{D(10)} &= [\,(1-P_D^A)P_D^{B0}(1-P_D^C) + P_D^A P_D^{B1}(1-P_D^C)\,]
\end{aligned}
\tag{3.160}
$$

$$
\begin{aligned}
\lambda_0^{D(11)} &= [\,(1-P_F^A)P_F^{B0}P_F^C + P_F^A P_F^{B1}P_F^C\,]\\
\lambda_1^{D(11)} &= [\,(1-P_D^A)P_D^{B0}P_D^C + P_D^A P_D^{B1}P_D^C\,]
\end{aligned}
\tag{3.161}
$$

Proposition 3.8 (Properties of the Gradient, Hessian of $P_\epsilon(\underline{\theta})$)

Let Team be a network of M DMs parameterized by N thresholds. Assume that $p_{Y_i|H_0}, p_{Y_i|H_1}$ are continuous and twice differentiable for all $i = 1, \ldots, M$, and that there exist bounded positive constants B_0^i, B_1^i, B_2^i and B_3^i such that

$$p_{Y_i|H_0}(y_i|H_0) \leq B_0^i, \; p_{Y_i|H_1}(y_i|H_1) \leq B_1^i, \;\; \forall y_i \in \Re \tag{3.162}$$

$$\left|\frac{d}{dy_i}(p_{Y_i|H_0}(y_i|H_0))\right| \leq B_2^i, \; \left|\frac{d}{dy_i}(p_{Y_i|H_1}(y_i|H_1))\right| \leq B_3^i, \;\; \forall y_i \in \Re \tag{3.163}$$

Then,

(a) *(Boundedness of the Gradient) There exists $K_1 > 0$ such that*

$$\left\|\nabla P_\epsilon^{Team}(\underline{\theta})\right\| \leq K_1, \;\; \forall \underline{\theta} \in \Re^N \tag{3.164}$$

(b) *(Boundedness of the Hessian, Lipschitz Continuity of the Gradient)*

(i) There exists $K_2 > 0$ such that

$$\left\|\nabla^2 P_\epsilon^{Team}(\underline{\theta})\right\| \leq K_2 \tag{3.165}$$

where the matrix norm denotes the maximum singular value.

(ii) There exists a constant L such that

$$\left\|\nabla P_\epsilon^{Team}(\underline{\theta}) - \nabla P_\epsilon^{Team}(\underline{\theta}')\right\| \leq L\|\underline{\theta} - \underline{\theta}'\|, \quad \forall \underline{\theta}, \underline{\theta}' \in \Re^N \tag{3.166}$$

Proof.

(a) We wish to bound the quantity

$$\|\nabla P_\epsilon(\underline{\theta})\| = \sqrt{\left(\frac{\partial P_\epsilon}{\partial \theta_1}(\underline{\theta})\right)^2 + \cdots + \left(\frac{\partial P_\epsilon}{\partial \theta_N}(\underline{\theta})\right)^2} \tag{3.167}$$

Let i_l denote the DM corresponding to component l. By Propositions 3.6 and 3.7(b),

$$\begin{aligned}\left|\frac{\partial P_\epsilon}{\partial \theta_l}(\underline{\theta})\right| &= \left|-\lambda_0^{i_l l} p_0 p_{Y_{i_l}|H_0}(\theta_l|H_0) + \lambda_1^{i_l l} p_1 p_{Y_{i_l}|H_1}(\theta_l|H_1)\right| \\ &\leq \sup_{\underline{\theta}} |\lambda_0^{i_l l}(\underline{\theta})| p_0 |p_{Y_{i_l}|H_0}(\theta_l|H_0)| + \sup_{\underline{\theta}} |\lambda_1^{i_l l}(\underline{\theta})| p_1 |p_{Y_{i_l}|H_1}(\theta_l|H_1)| \\ &\leq \lambda_0^{i_l l} p_0 B_0^{i_l} + \lambda_1^{i_l l} p_1 B_1^{i_l} \end{aligned} \tag{3.168}$$

where the $\lambda_j^{i_l l}$ denote upper bounds on the magnitude of the coupling costs. If we then denote

$$K_1' = \max_{l=1,\ldots,N} \{\lambda_0^{i_l l} p_0 B_0^{i_l} + \lambda_1^{i_l l} p_1 B_1^{i_l}\} \tag{3.169}$$

then

$$\|\nabla P_\epsilon(\underline{\theta})\| \leq \sqrt{N} K_1' \tag{3.170}$$

so that we may take $K_1 = K_1'$.

(b)(i) The maximum singular value of the Hessian is bounded if and only if each element is bounded. There are two cases to consider.

Case $l = j$: Then

$$\begin{aligned}\left|\frac{\partial^2 P_\epsilon}{\partial \theta_l^2}(\underline{\theta})\right| &= \left|-\lambda_0^{i_l l}(\underline{\theta}) p_0 \frac{d}{dy_{i_l}}(p_{Y_{i_l}|H_0}(\theta_l|H_0)) + \lambda_1^{i_l l}(\underline{\theta}) p_1 \frac{d}{dy_{i_l}}(p_{Y_{i_l}|H_1}(\theta_l|H_1))\right| \\ &\leq \lambda_0^{i_l l} p_0 B_2^{i_l} + \lambda_1^{i_l l} p_1 B_3^{i_l} \\ &< \infty \end{aligned} \tag{3.171}$$

where the $\lambda_j^{i_l l}$ denote upper bounds on the magnitude of the coupling costs.

Case $l \neq j$: Then from Proposition 3.7(a)

$$\begin{aligned}\left|\frac{\partial^2 P_\epsilon}{\partial \theta_l \partial \theta_j}\right| &= \left|-\frac{\partial}{\partial \theta_j}\lambda_0^{i_l l}(\underline{\theta}) p_0 p_{Y_{i_l}|H_0}(\theta_l|H_0) + \frac{\partial}{\partial \theta_j}\lambda_1^{i_l l}(\underline{\theta}) p_1 p_{Y_{i_l}|H_1}(\theta_l|H_1)\right| \\ &\leq (A_0 B_0^j) p_0 B_0^{i_l} + (A_1 B_1^j) p_1 B_1^{i_l} \\ &< \infty \end{aligned} \tag{3.172}$$

where A_0 and A_1 are some positive constants.

(ii) We do not construct a bound for this case as it is messy, but rather refer to a

result from Bertsekas [5] which states that it follows directly from boundedness of the Hessian that the gradient obeys a Lipschitz condition.

■

Thus, the gradient and Hessian for the team cost were shown to possess the desired smoothness properties under reasonable boundedness and differentiability properties of the local conditional density functions. Because it is analytically difficult for the team problem, we do not attempt a detailed analysis of the Gaussian detection case.

3.4 Unconstrained Optimization of $P_\epsilon(\underline{\theta})$ using Gradient Descent

In this section we have two goals: to address some of the issues which arise in gradient descent optimization of the thresholds using complete knowledge of the statistics, and then to characterize the stationary points of both the single DM and team problems.

The most basic gradient method for unconstrained nonlinear optimization, known as steepest descent, is of the form

$$\underline{\theta}_{k+1} = \underline{\theta}_k - \rho_k \nabla J(\underline{\theta}_k),\ k = 1, 2, \ldots \tag{3.173}$$

where ρ_k is a possibly time-varying stepsize sequence. This iteration is the deterministic analog of the stochastic steepest descent algorithms we investigate for training in Chapters 4 and 5. Ideally, the stepsize ρ_k is chosen by line minimization, i.e., by choosing as the next iterate the value of $\underline{\theta}$ resulting in the minimum cost along direction $-\nabla J(\underline{\theta}_k)$ as in

$$\rho_k = \arg\min_{\rho} J(\underline{\theta}_k - \rho \nabla J(\underline{\theta}_k)) \tag{3.174}$$

However, a problem familiar to those readers who have worked with stochastic optimization or neural networks is that line minimization is often impractical or impossible. This is the case for our training algorithms as well. In the deterministic

problem, convergence of the steepest descent method to a stationary point using a *constant* stepsize can still be guaranteed for functions possessing a Lipschitz continuous gradient, provided that the stepsize is chosen small enough, where the maximum size is a function of the Lipschitz constant [5], [37]. When noise is present, as in our nonparametric training algorithms, a *decreasing* stepsize must be chosen to guarantee convergence. Constant and decreasing stepsizes generally result in a much slower rate of convergence than line minimization.

Loosely speaking, gradient descent methods using constant or decreasing stepsize operate as follows: the negative gradient represents a "downhill" direction of the cost, so that the parameters are continually updated, using small steps, in a downhill direction in an attempt to drive the cost to its minimum value. The method is not finitely convergent in general[5], and terminates only when the gradient becomes zero, i.e., at a stationary point. Thus, the most we can hope to guarantee is that the algorithms asymptotically determine a stationary point, or more specifically that every limit point is a stationary point. The method may become "stuck" at any stationary point(s), and cannot be assured to even find a local mimimum. If the cost is unimodal, with a single stationary point representing the global minimum, then the gradient algorithms can be guaranteed to asymptotically find the global minimum, subject to the previous discussion concerning stepsize.

Unfortunately, we will not be able to prove that the team cost is unimodal, even in the Gaussian case for a simple topology like 2-Tand. This discussion indicates that the strongest property we can hope to guarantee for the network gradient-based training algorithms we present in Chapter 5 are that they will converge, in some probabilistic sense, to the set of stationary points[6].

In addition, there are certain well-known difficulties associated with the steepest descent method [5], [37], [51]. A major problem is that the method is quite sensitive to scaling of the cost surface. In particular, elongated rather than round contours can

[5]It is usually terminated when a condition such as $||\nabla J(\underline{\theta}_k)|| \leq \epsilon$, for ϵ a small positive scalar, is met.

[6]In particular, we show that every limit point of the training algorithms is a stationary point with probability one.

cause difficulty, even when line minimization is used on unimodal problems, because the direction indicated by the gradient may not be in the direction of the minimum. In fact, it can even be close to perpendicular to the desired direction, so that the iterates travel along a zig-zagging path. In our application, elongated contours can arise, for example in the Gaussian case, when extreme choices of variance of the DMs in the team are made. These difficulties are normally addressed with the addition of positive definite scaling matrices R_k which modify (3.173) to the scaled steepest descent iteration

$$\underline{\theta}_{k+1} = \underline{\theta}_k - \rho_k R_k \nabla J(\underline{\theta}_k), \; k = 1, 2, \ldots \tag{3.175}$$

Properly chosen scaling effectively warps the surface, and converts elongated contours to rounded contours, substantially improving the rate of convergence of the algorithm. A well-known choice of scaling is the inverse Hessian

$$R_k = (\nabla^2 J(\underline{\theta}_k))^{-1} \tag{3.176}$$

which gives rise to the so-called Newton's method. We avoid consideration of these more sophisticated gradient methods in this report for simplicity and because our numerical experiments do not focus on badly scaled problem instances. We leave consideration of these methods for future work.

Another well known difficulty associated with steepest descent optimization is that it often exhibits a very slow rate of convergence, even on problems which are reasonably scaled. This problem is particularly evident in flat regions of the cost, for which the magnitude of the gradient is quite small, resulting in the magnitude of the update vector becoming quite small.

It is important to emphasize what we can expect from steepest descent employing constant or decreasing stepsize, even if the true gradient can be measured exactly. Since the situation in the training problem is certainly worse than that in the deterministic case, we can expect to encounter all of the difficulties of the deterministic case, in addition to additional complications caused by employing stochastic estimates of the gradient in the updates. We now characterize the stationary points for both

the single DM and team problems.

3.4.1 Single DM

We begin by characterizing the stationary points of $J_B(\theta)$.

Proposition 3.9 (Characterization of Stationary Points)
(a) *All values of the parameter θ^* such that*

$$\lambda_0 p_0 p_{Y|H_0}(\theta^*|H_0) = \lambda_1 p_1 p_{Y|H_1}(\theta^*|H_1) \tag{3.177}$$

satisfy the first-order necessary conditions for optimality of $J_B(\theta)$.
(b) *If the scaled conditional densities have a single point of intersection, (3.177) has a single solution, and that solution is the unique global minimizer of $J_B(\theta)$.*

Proof.
(a) Setting

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 p_{Y|H_0}(\theta|H_0) + \lambda_1 p_1 p_{Y|H_1}(\theta|H_1) \tag{3.178}$$

equal to zero, we see that point(s) θ^* which satisfy the equation

$$\lambda_0 p_0 p_{Y|H_0}(\theta^*|H_0) = \lambda_1 p_1 p_{Y|H_1}(\theta^*|H_1) \tag{3.179}$$

satisfy the first-order necessary condition for optimality.
(b) Rewrite

$$J_B(\theta) = \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{3.180}$$

in the form

$$J_B(\theta) = \lambda_1 p_1 + \left[\int_\theta^\infty (\,\lambda_0 p_0 p_{Y|H_0}(y|H_0) - \lambda_1 p_1 p_{Y|H_1}(y|H_1)\,)\,dy\right] \tag{3.181}$$

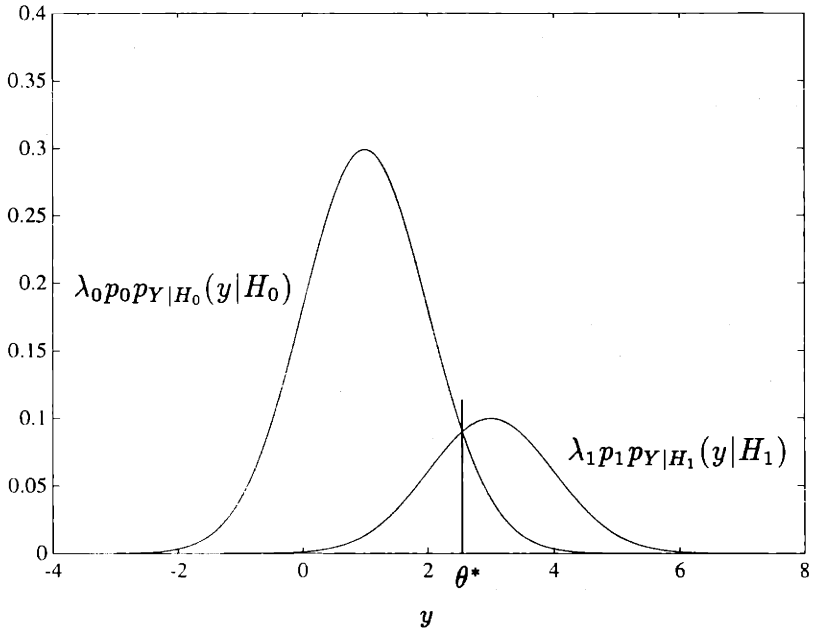


Figure 3-31: Typical Scaled Conditional Density Functions; Gaussian detection.

the value of which is clearly *minimized* if θ is chosen such that

$$\theta^* = \arg\min_{\theta \in \Re}[\int_{\theta}^{\infty} (\lambda_0 p_0 p_{Y|H_0}(y|H_0) - \lambda_1 p_1 p_{Y|H_1}(y|H_1))\, dy] \tag{3.182}$$

Since there is assumed to be a single point of intersection θ^* of the scaled conditional densities, it must hold that

$$\lambda_0 p_0 p_{Y|H_0}(y|H_0) \leq \lambda_1 p_1 p_{Y|H_1}(y|H_1), \quad \forall y > \theta^* \tag{3.183}$$

so that θ^* is the unique global minimizer of $J_B(\theta)$[7] .

■

The proposition says that a value of the observation which satisfies the LRT with equality is optimal, an intuitive result. The argument in the proof of part (b) of the proposition is depicted in Figure 3-31.

For the Gaussian detection problem, it is easily demonstrated that there is only

[7]If the opposite inequality holds, we may relabel the corresponding decision/hypothesis pair to obtain an equivalent statement.

a single point of intersection for *any choice* of positive priors, and *any choice* of positive bounded costs, since the single point satisfying the necessary condition may be determined in closed form. Viz.,

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}} + \lambda_1 p_1 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}} = 0 \tag{3.184}$$

may be solved for θ^* to yield the closed-form solution

$$\theta^* = \frac{\sigma^2}{\mu_1 - \mu_0} \ln \eta + \frac{\mu_1 + \mu_0}{2} \tag{3.185}$$

where $\eta = (\lambda_0 p_0)/(\lambda_1 p_1)$ which is identical to (2.22) defining the optimal threshold. By Proposition 3.9, there is no ambiguity that this single stationary point is the unique minimizer. That θ^* is the unique minimum is also easily verified by considering the second-order sufficient condition for optimality. The second derivative is given by

$$\frac{d^2 J_B}{d\theta^2}(\theta) = \lambda_0 p_0 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}} (\theta - \mu_0) - \lambda_1 p_1 \frac{1}{\sigma^3\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}} (\theta - \mu_1) \tag{3.186}$$

which we may rewrite as

$$\frac{d^2 J_B}{d\theta^2}(\theta) = (\lambda_0 p_0 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_0)^2}{2\sigma^2}})(\frac{1}{\sigma^2}(\theta - \mu_0)) - (\lambda_1 p_1 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta-\mu_1)^2}{2\sigma^2}})(\frac{1}{\sigma^2}(\theta - \mu_1)) \tag{3.187}$$

which, when evaluated at θ^*, yields

$$\frac{d^2 J_B}{d\theta^2}(\theta^*) = K \frac{1}{\sigma^2}(\mu_1 - \mu_0) > 0 \tag{3.188}$$

since

$$K = \lambda_0 p_0 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta^*-\mu_0)^2}{2\sigma^2}} = \lambda_1 p_1 \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\theta^*-\mu_1)^2}{2\sigma^2}} > 0 \tag{3.189}$$

and $\mu_1 > \mu_0$ by assumption if the problem is nontrivial.

We summarize the previous discussion in the form of the following additional corollary to Proposition 3.9.

Corollary 3.1 (Unimodality of $J_B(\theta)$ for the Gaussian Detection Problem) *For the Gaussian detection problem, with positive prior probabilities and positive bounded costs, the Bayes risk $J_B(\theta)$ is unimodal with a single global minimum.*

As we show in the next section, the local subproblem faced by a DM trying to determine a person-by-person optimal solution in the team problem is of the same form as the unequal cost single DM problem. This result suggests that, for the Gaussian problem at least, each person-by-person problem in the team case possesses desirable structure.

3.4.2 Team Problem

We now proceed to characterize the stationary points for the team problem.

Proposition 3.10 (Stationary Points for the Team problem) *Assume Team comprises M DMs, and is parameterized by N threshold parameters. Let $\lambda_0^{i_l l}, \lambda_1^{i_l l}$ denote the coupling costs corresponding to component θ_l controlled by DM i_l. Then, values of the parameter vector $\underline{\theta}^* \in \Re^N$ such that*

$$\begin{aligned}
\lambda_0^{i_1 1}(\underline{\theta}^*) p_0 p_{Y_{i_1}|H_0}(\theta_1^*|H_0) &= \lambda_1^{i_1 1}(\underline{\theta}^*) p_1 p_{Y_{i_1}|H_1}(\theta_1^*|H_1) \\
\lambda_0^{i_2 2}(\underline{\theta}^*) p_0 p_{Y_{i_2}|H_0}(\theta_2^*|H_0) &= \lambda_1^{i_2 1}(\underline{\theta}^*) p_1 p_{Y_{i_2}|H_1}(\theta_2^*|H_1) \\
&\vdots \\
\lambda_0^{i_N N}(\underline{\theta}^*) p_0 p_{Y_{i_N}|H_0}(\theta_N^*|H_0) &= \lambda_1^{i_N N}(\underline{\theta}^*) p_1 p_{Y_{i_N}|H_1}(\theta_N^*|H_1)
\end{aligned} \tag{3.190}$$

satisfy the first-order necessary conditions for optimality of $P_\epsilon^{Team}(\underline{\theta})$.

Proof. This statement is a direct consequence of Proposition 3.6. ■

It is difficult to characterize values of the parameter vector which solve this simultaneous system. For example, it is not clear what conditions should be imposed to guarantee a unique solution to this system.

However, if we proceed in a person-by-person fashion, considering the system one coordinate at a time, we can make following additional statements, which follow immediately from Proposition 3.6, and the results of the previous section.

Proposition 3.11 (Person-by-Person Optimal Solutions)

Assume Team comprises M DMs, and is parameterized by N threshold parameters. Assume parameter θ_l is controlled by DM i.

(a) *All values of the parameter θ_l^* such that*

$$\lambda_0^{il}(\underline{\theta})p_0 p_{Y_i|H_0}(\theta_l^*|H_0) = \lambda_1^{il}(\underline{\theta})p_1 p_{Y_i|H_1}(\theta_l^*|H_1) \tag{3.191}$$

satisfy the necessary conditions for optimality for parameter θ_l, given that θ_j, $j = 1,\ldots,N$, $j \neq l$ are held fixed.

(b) *If there is a single solution θ_l^* to equation (3.191), and the costs $\lambda_0^{il}(\underline{\theta})$ and $\lambda_1^{il}(\underline{\theta})$ are both positive and bounded, then θ_l^* is the unique global minimizer of $P_\epsilon^{Team}(\underline{\theta})$ along coordinate l given that θ_j, $j = 1,\ldots,N$, $j \neq l$ are held fixed.*

The following corollary then follows immediately from the properties of the conditional densities for the Gaussian detection problem.

Corollary 3.2 (Unimodality of $P_\epsilon^{Team}(\underline{\theta})$ along each Coordinate for the Gaussian Detection)

For the Team Gaussian detection problem, with positive prior probabilities, the probability of error $P_\epsilon^{Team}(\underline{\theta})$ is unimodal with a single global minimum along coordinate l, given fixed θ_j, $j = 1,\ldots,N$, $j \neq l$, if $\lambda_0^{il}(\underline{\theta}), \lambda_1^{il}(\underline{\theta}) > 0$ and bounded.

This proposition indicates that the person-by-person subproblems are well-posed in a certain sense. That is, a unique best local solution may be determined, given a fixed configuration of the decision rules of the other DMs, provided that the induced local costs are positive and bounded. The costs are bounded by Proposition 3.7. What we would like is to guarantee that these costs are always positive, under arbitrary choices of threshold by the other DMs. However, under independent choice of the threshold parameters this is clearly not the case.

Consider for example the costs for 2-Tand given by

$$\lambda_0^A = [P_F^{B1} - P_F^{B0}], \quad \lambda_1^A = [P_D^{B1} - P_D^{B0}] \tag{3.192}$$

$$\lambda_0^{B0} = [1 - P_F^A], \quad \lambda_1^{B0} = [1 - P_D^A] \tag{3.193}$$

$$\lambda_0^{B1} = P_F^A, \quad \lambda_1^{B1} = P_D^A \tag{3.194}$$

The costs for DM B are both positive, for arbitrary choice of the threshold parameter α. However, the sign of the costs for DM A depend on the relative positions of the threshold parameters β_0 and β_1. In particular, it can be shown, based on monotonicity of the ROC, that if $\beta_0 > \beta_1$ the costs are positive, if $\beta_0 < \beta_1$ the costs are negative, and clearly if $\beta_0 = \beta_1$ then the costs are zero.

It turns out that the nature of the construction of the cost function ensures that the costs for all parameters of the primary DM are always positive, for arbitrary configurations of the remaining network thresholds. However, the situation for the costs of the other DMs is not so clear. It appears that it may be possible, using properties of the ROC, to demonstrate the the costs always have the same sign, although the proof of this fact appears difficult. The implication would be that the intersection problems described above always have a solution because a point of intersection would always exist for each subproblem, for arbitrary choices of the other parameters. This is clearly not the case if the costs may take opposite signs, in which case no solution (stationary point along that coordinate) exists. Furthermore, in the case of both costs being negative, the local subproblem is still unimodal, but now

with a single global maximum.

The source of these difficulties is the *independent* choice of each network threshold, such as occurs with arbitrary initialization of the threshold parameters. If the thresholds are determined by iteration on the necessary conditions, for example, so that β_0 and β_1 are not independently chosen but rather computed based on a *common* value of (P_F^A, P_D^A), then it can be shown, under a strict concavity assumption on the ROC, that it always holds that $\beta_0 > \beta_1$, and the costs are always positive [60]. Similar conclusions may be reached in general using optimal control arguments as in [59].

The point of this discussion is to note that the local subproblems in person-by-person optimization of the team cost are not guaranteed to be well-posed under independent choice of the network observation thresholds. This suggests that caution should be exercised in the use of these methods. In particular, person-by-person schemes may be well-behaved only for initializations in certain regions of parameter space.

However, when well-posed, an interesting interpretation of the person-by-person optimization is suggested by these results. The subproblem faced by a given DM in the team in order to determine its person-by-person optimal threshold, is to determine the point of intersection of its scaled local conditional densities, where the scaling is determined by the current values of the other DMs' thresholds. As the thresholds of the other DMs change, the associated scalings change.

To visualize this, consider Figures 3-32, 3-33, 3-34 which depict the situations corresponding to Figures 2-9(a), 2-10(b), and 2-11(c), respectively. The figures illustrate that under optimal scaling, that is scaling corresponding to the optimal threshold settings, the optimal thresholds correspond to the points of intersection of the scaled local densities.

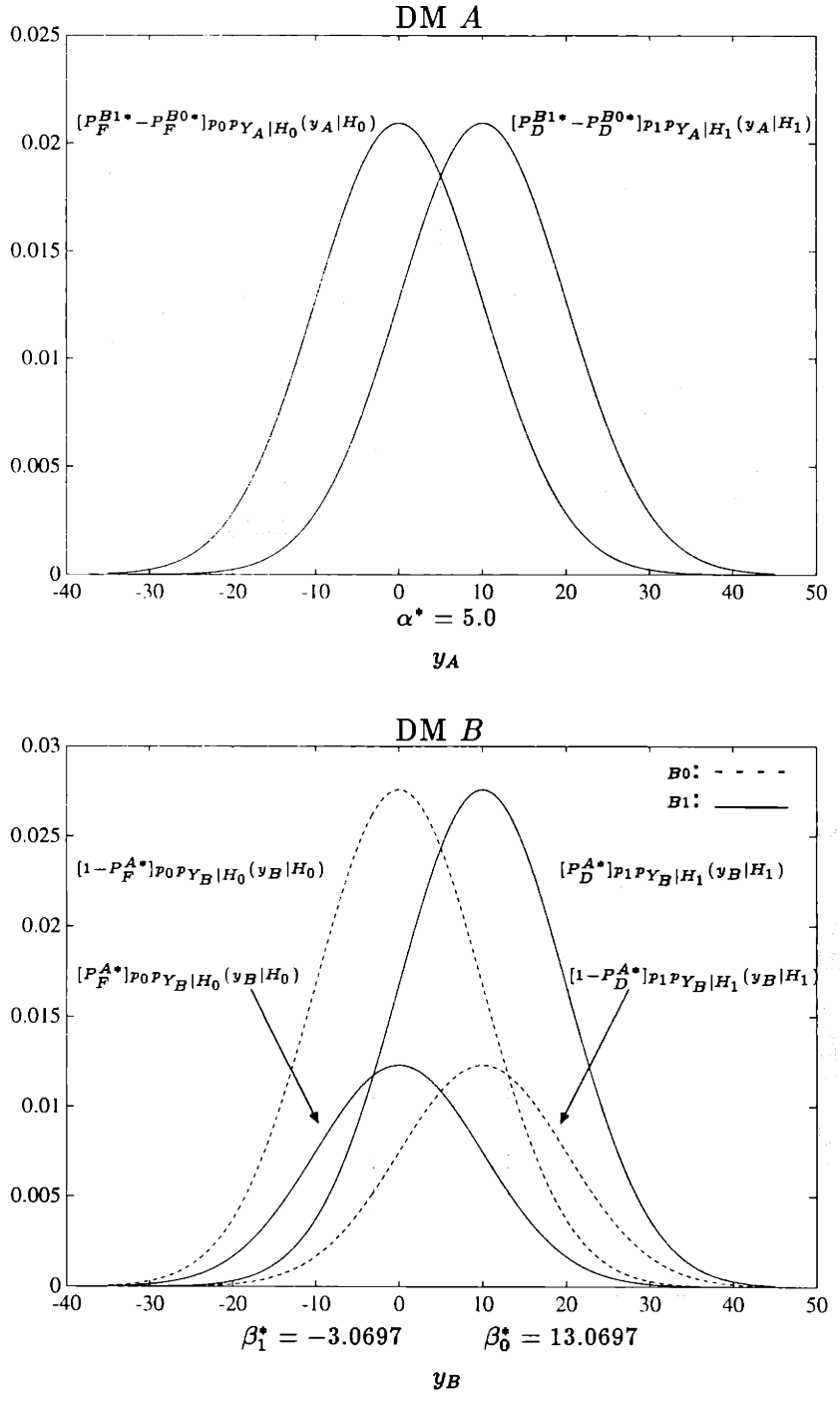


Figure 3-32: 2-Tand Optimally Scaled Conditional Densities: Equally Smart Case $\mu_0 = 0, \mu_1 = 10, \sigma_A^2 = \sigma_B^2 = 100, p_0 = 0.5$. Stars indicate operating points correspond to optimal threshold settings.

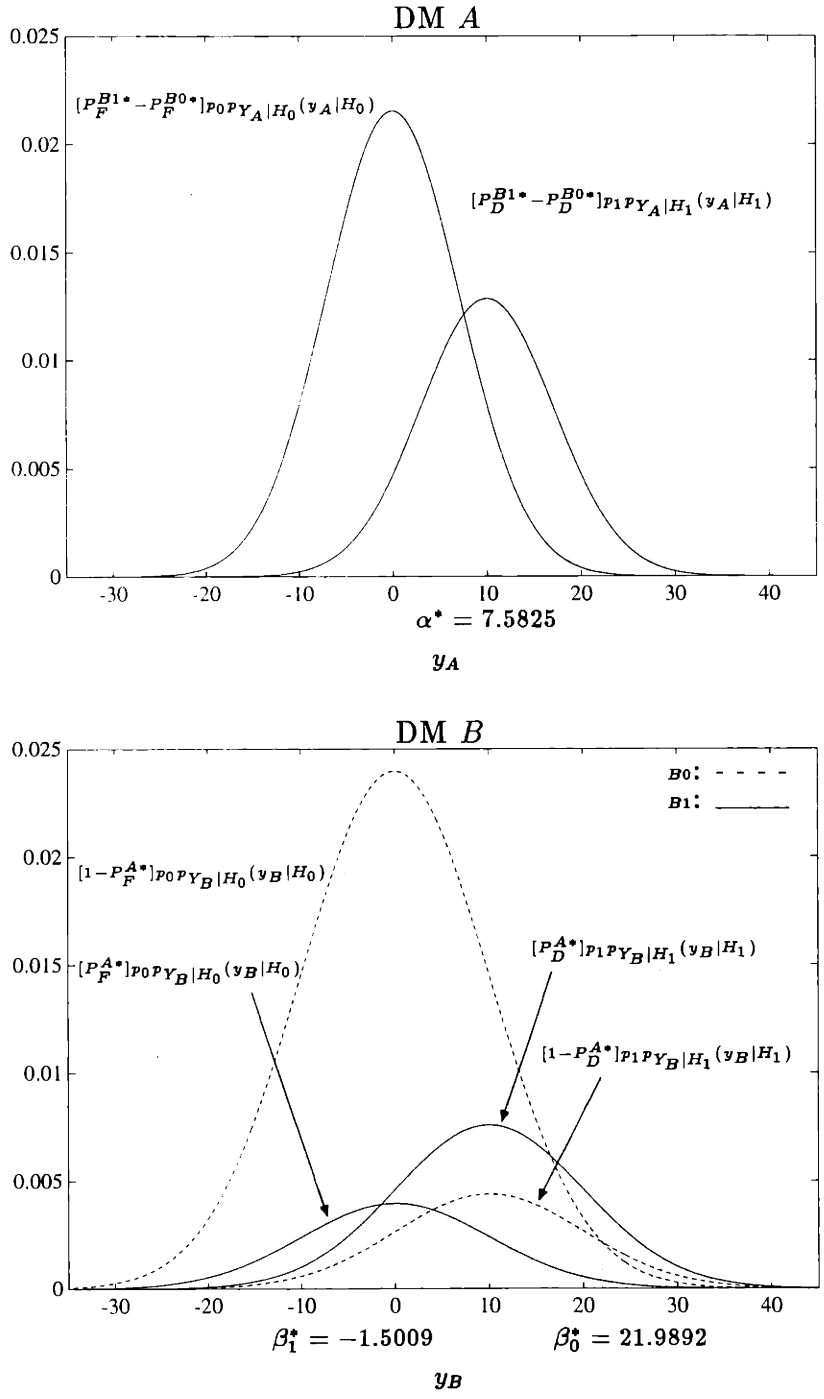


Figure 3-33: 2-Tand Optimally Scaled Conditional Densities: DM A Smart, DM B Dumb Case, $\mu_0 = 0, \mu_1 = 10$, $\sigma_A^2 = 50, \sigma_B^2 = 100$, $p_0 = 0.7$. Stars indicate the operating points correspond to optimal threshold settings.

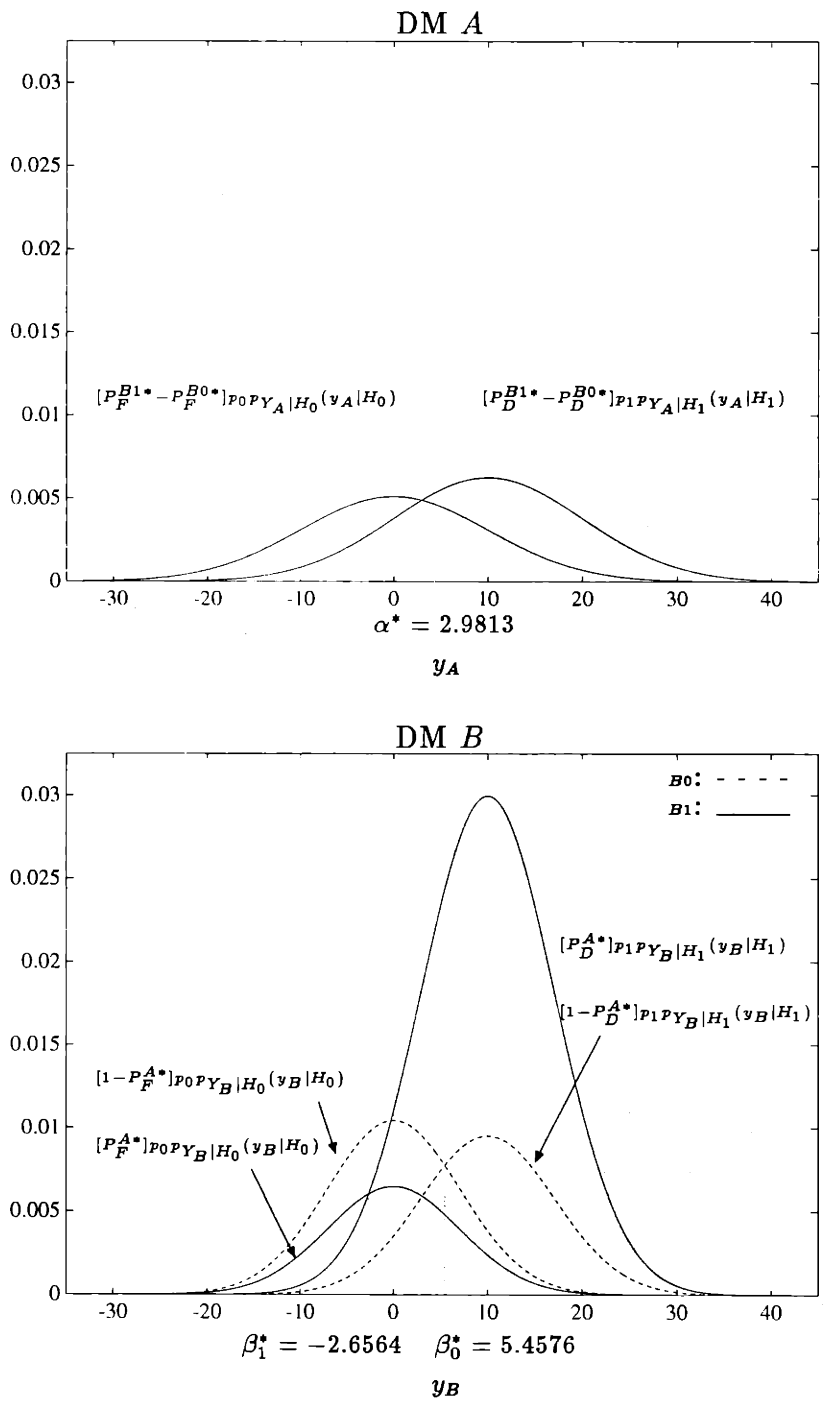


Figure 3-34: 2-Tand Optimally Scaled Conditional Densities: DM A dumb, DM B smart, $\mu_0 = 0, \mu_1 = 10$, $\sigma_A^2 = 100, \sigma_B^2 = 50$, $p_0 = 0.3$. Stars indicate the operating points correspond to optimal threshold settings.

3.5 Fixed Point Solutions

An alternative approach to obtaining the optimal thresholds is to try and directly solve the system of nonlinear equations which specify the necessary conditions for optimality. There are several potential difficulties with this approach. This problem is hard as it requires simultaneous solution of a system of nonlinear equations, and must be done numerically using an iterative technique as well. There may be multiple solutions to the system of equations, and solutions are only guaranteed to be stationary points of the cost. We comment on the approach because it will later be used to motivate one of the network training algorithms.

There are a multitude of techniques for solving systems of nonlinear equations [42]. We consider successive approximation as it has proved quite successful in the problems of interest to us. A successive approximation solution is generated as follows for 2-Tand. Recall the system of equations (2.71) which define the necessary conditions for optimality for 2-Tand for the Gaussian detection problem.

$$\begin{aligned}
\alpha &= \frac{\sigma_A^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_{\beta_0}(0)-\Phi_{\beta_1}(0)}{\Phi_{\beta_0}(1)-\Phi_{\beta_1}(1)}\right)+\frac{\sigma_A^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2} \\
\beta_0 &= \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\alpha(0)}{\Phi_\alpha(1)}\right)+\frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2} \qquad (3.195) \\
\beta_1 &= \frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{1-\Phi_\alpha(0)}{1-\Phi_\alpha(1)}\right)+\frac{\sigma_B^2}{\mu_1-\mu_0}\ln\left(\frac{p_0}{p_1}\right)+\frac{\mu_0+\mu_1}{2}
\end{aligned}$$

Suppose we define the parameter vector $\underline{\theta} = [\alpha, \beta_0, \beta_1]^T$ and view system (3.195) as a vector-valued equation of the form

$$\underline{\theta} = \underline{T}(\underline{\theta}) \qquad (3.196)$$

where $\underline{T} : \Re^3 \mapsto \Re^3$. Then we might attempt to determine a fixed point solution to equation (3.196) by successive approximation using the iterative algorithm

$$\underline{\theta}_{k+1} = \underline{T}(\underline{\theta}_k) \qquad (3.197)$$

where a fixed point is a vector $\underline{\theta^*}$ such that

$$\underline{\theta^*} = \underline{T}(\underline{\theta^*}) \tag{3.198}$$

If the mapping $\underline{T}$ can be shown to possess a certain *contraction* property, i.e., if it can be shown that there exists a constant $\alpha \in [0,1)$, referred to as the modulus, such that

$$\|\underline{T}(\underline{\theta}) - \underline{T}(\underline{\theta}')\| \leq \|\underline{\theta} - \underline{\theta}'\|, \quad \forall \underline{\theta}, \underline{\theta}' \in \Re^3 \tag{3.199}$$

where $\|\cdot\|$ denotes the Euclidean norm[8], then (3.197) is a contracting iteration and comes with certain beneficial guarantees. Namely, we obtain the following well-known result [6].

Proposition 3.12 (Convergence of Contracting Iterations)
Suppose that $\underline{T} : \mathcal{X} \mapsto \mathcal{X}$ is a contraction with modulus $\alpha \in [0,1)$ and that $\mathcal{X}$ is a closed subset of $\Re^N$. Then:
(a) *(Existence and Uniqueness of Fixed Points) The mapping T has a unique fixed point $\underline{\theta}^* \in \mathcal{X}$*
(b) *(Geometric Rate of Convergence) For any initial vector $\underline{\theta}_1 \in \mathcal{X}$, the sequence $\{\underline{\theta}_k\}$ generated by the iteration $\underline{\theta}_{k+1} = \underline{T}(\underline{\theta}_k)$ converges to $\underline{\theta}^*$ geometrically. In particular,*

$$\|\underline{\theta}_k - \underline{\theta}^*\| \leq \|\underline{\theta}_1 - \underline{\theta}^*\|, \quad k = 1, 2, \ldots \tag{3.200}$$

Actually, it is useful to express these conditions in a slightly different way since for our system the set $\mathcal{X} = \Re^3 = \Re \times \Re \times \Re$. Note that system (3.195) is of the form

$$\begin{aligned} \alpha &= T_1(\beta_0, \beta_1) \\ \beta_0 &= T_2(\alpha) \end{aligned} \tag{3.201}$$

[8]More generally, contractions can be defined for any norm

$$\beta_1 \;=\; T_3(\alpha) \tag{3.202}$$

where $\underline{T}(\underline{\theta}) = (T_1(\underline{\theta}), T_2(\underline{\theta}), T_3(\underline{\theta}))$ and $T_i : \Re^3 \mapsto \Re$. System (3.195) is decomposed into the set of scalar equations

$$\theta_i = T_i(\theta_1, \theta_2, \theta_3), \;\; i = 1, \ldots, 3 \tag{3.203}$$

which must be solved simultaneously. Iterative techniques for solving such systems are referred to as component solution methods [6]. In order for the conclusions of Proposition 3.12 to remain valid for component solution methods, there must exist an $\alpha \in [0, 1)$ such that $\underline{T}$ satisfies the condition

$$\max_{i \in \{1,2,3\}} \{|T_i(\underline{\theta}) - T_i(\underline{\theta}')|\} \leq \alpha \max_{j \in \{1,2,3\}} \{|\underline{\theta}_j - \underline{\theta}'_j|\}, \;\; \forall \underline{\theta}, \underline{\theta}' \in \Re^3 \tag{3.204}$$

This condition is known as a *block-contraction* condition. Unfortunately, demonstrating that the systems of necessary conditions which arise for Gaussian DBHT networks are block contractions appears difficult, and in fact it remains an open question whether or not any of the systems, including system (3.195) possess this property. The primary source of difficulty is that the Gaussian error functions are algebraically cumbersome, and cannot be evaluated analytically in closed-form. Clearly, verifying the block-contraction property would be of substantial value, since it would indicate that a unique solution exists to the system of necessary conditions, implying the existence of a single stationary point for the team probability of error in the Gaussian case. If it could then be demonstrated that this point were a minimum, it would be guaranteed that the error function is in fact unimodal with a single global minimum for the Gaussian problem. There is still hope that at least sufficient conditions might be determined under which the system is a block contraction, and we continue to pursue this using an approach suggested by Irving in [27].

However, despite the apparent difficulty in verifying the condition analytically, we have had success in numerical experiments implemented with the component solution method. It is useful to examine the dependence of each parameter on the others

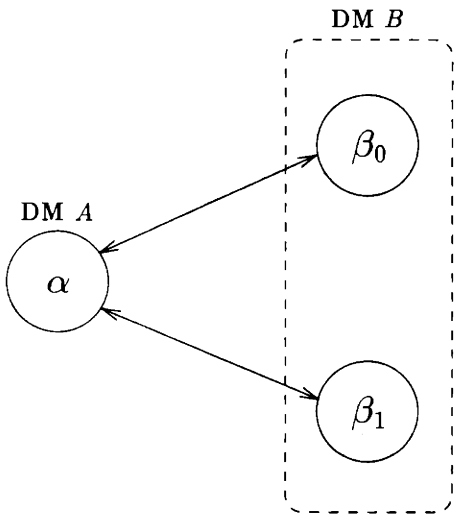


Figure 3-35: Dependency Graph for the 3 parameters of 2-Tand

graphically on a so-called dependency graph such as that shown for 2-Tand in Figure 3-35. Dependency graphs are directed acyclic graphs (DAGs), where each node in the graph corresponds to a parameter, and each directed arc (i, j) appearing in the graph indicates a functional dependence of parameter j on parameter i. If both arcs (i, j) and (j, i) are present, this is indicated with a bi-directional arc. The form of the dependency graph for 2-Tand indicates that there is no functional dependence of β_0 on β_1 or vice versa. This principle holds true in general; there is never any functional dependence between the collection of thresholds held by a single DM. This is illustrated by the dependency graphs for the other small teams we have examined, which are shown in Figures 3-36 - 3-38.

The implication of these graphs is that all the parameters controlled by a single DM may be updated simultaneously. Thus, rather than updating all parameters simultaneously as in iteration (3.197), it makes sense to update the parameters in a person-by-person manner. We may thus implement a Gauss-Seidel component solution on system (3.195) by first fixing β_0 and β_1 and then determining α, then substituting in the new α to update β_0 and β_1 and so on. The relative timing of the updates in this scheme is illustrated by the timing diagram of Figure 3-39.

This technique has proven quite successful in our experience. Numerical studies have demonstrated convergence to apparently unique fixed points at rates which

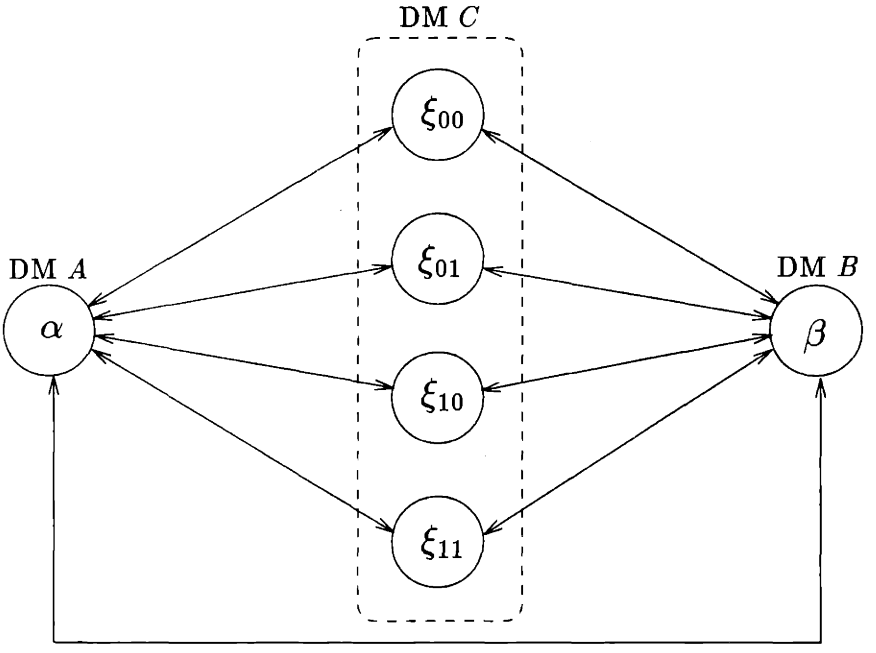


Figure 3-36: Dependency Graph for the 6 parameters of 3-Vee

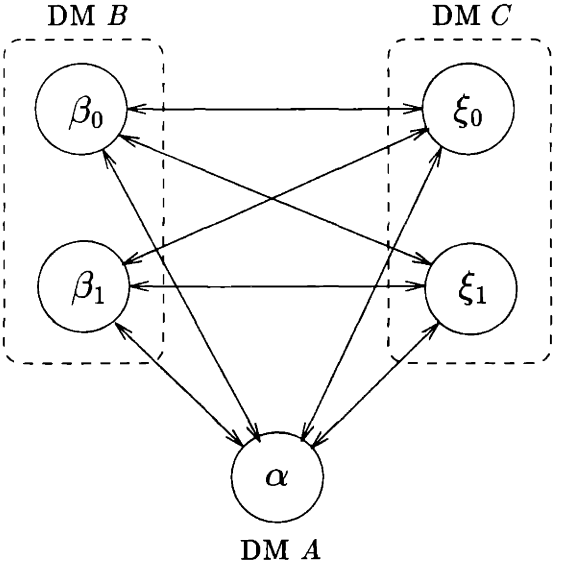


Figure 3-37: Dependency Graph for the 5 parameters of 3-Tand

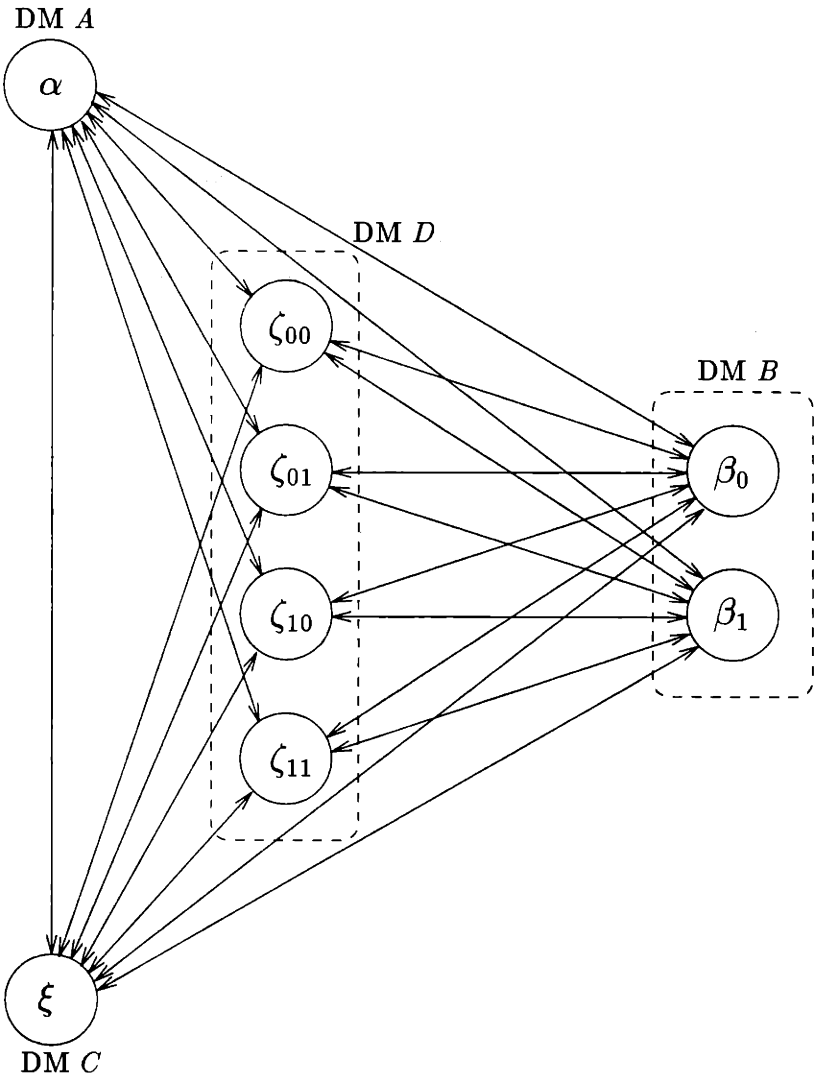


Figure 3-38: Dependency Graph for the 8 parameters of 4-Asym

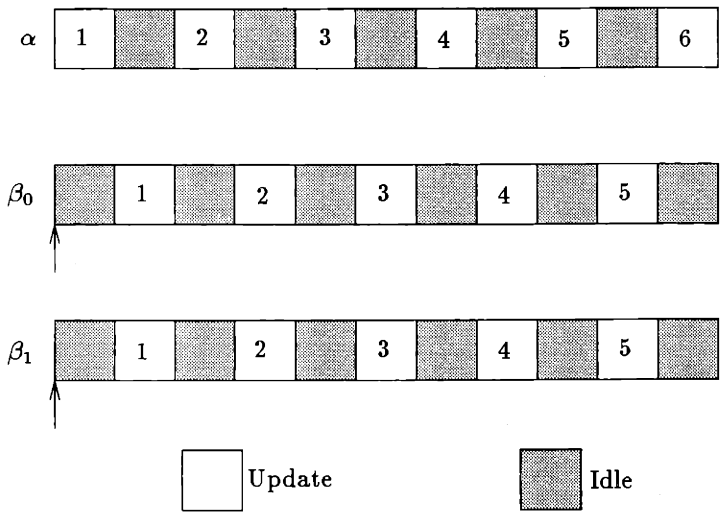


Figure 3-39: Timing Diagram for Successive Approximation Iterations on 2-Tand. Arrows indicate beginning with initial conditions.

appear geometric. Of course, such results in no way verify the existence of the block-contraction property, but indicate only that the nice properties it would otherwise guarantee do appear to hold for the cases we have examined.

A typical set of paths resulting from Gauss-Seidel componentwise iteration is shown in Figure 3-40, where the iterates for the thresholds for 2-Tand are displayed. The associated probability of error after each completed update cycle of all three thresholds is shown in Figure 3-41.

These starting conditions might be considered rather favorable since both β_0 and β_1 were chosen "within the bowl", i.e., between the means, and were initialized with the correct relative orientation to one another, and the performance was close to optimal from the outset. However, similar results are obtained from a variety of conditions, although it is beyond the scope of the present discussion to illustrate this fact. We presented this particular example because we intend to return to it later.

A second example which we also require later is shown in Figures 3-42 and 3-43. These are typical Gauss-Seidel component solution iterates for a particular Gaussian instance of 3-Vee. Notice that the thresholds ξ_{01} and ξ_{10} are initially crossed, but straighten out after a single iteration.

Since we have previously demonstrated that, if the scaled conditional densities

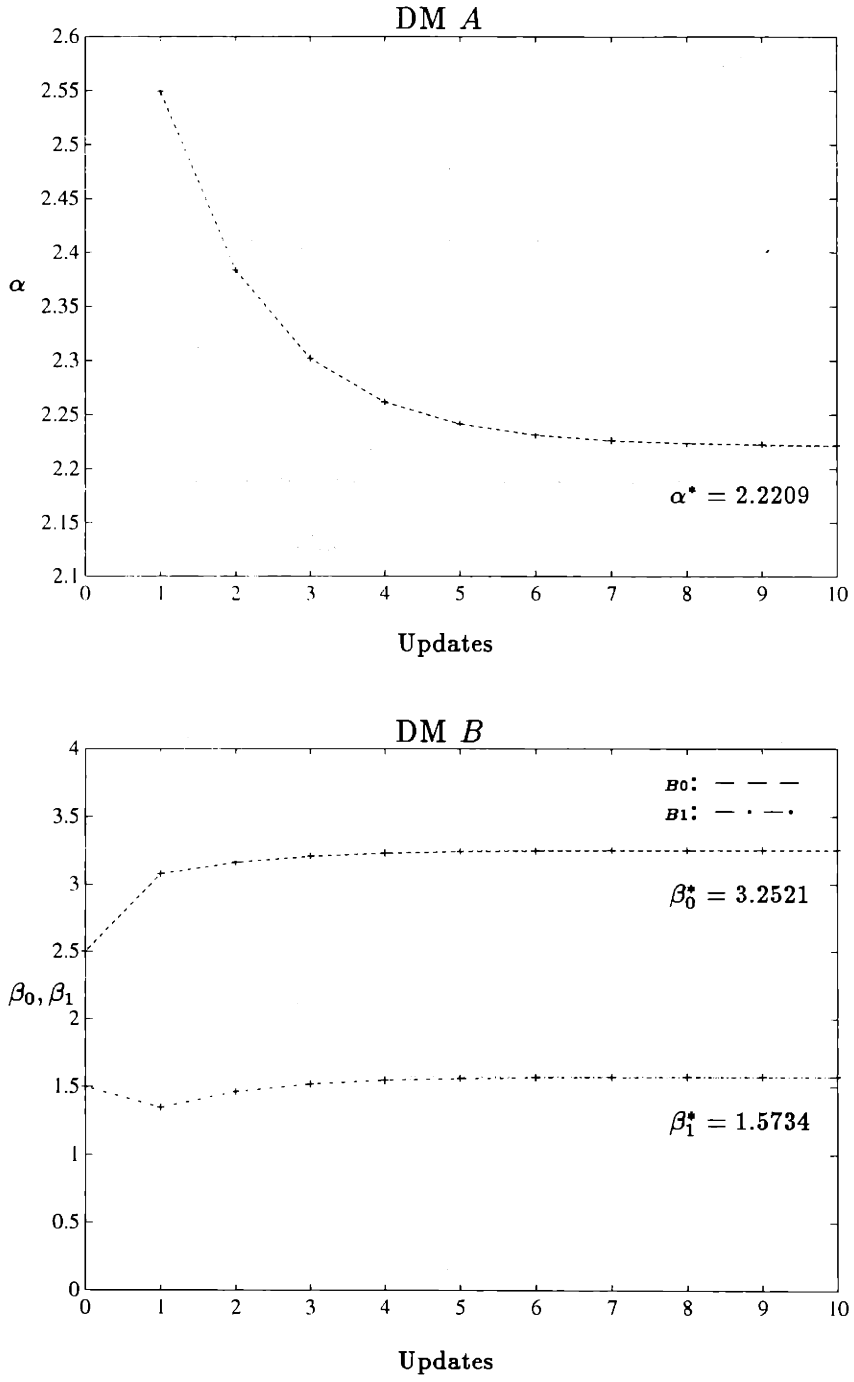


Figure 3-40: 2-Tand Fixed Point Iterations: Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$. Initial conditions were $\beta_0 = 2.5$ and $\beta_1 = 1.5$.

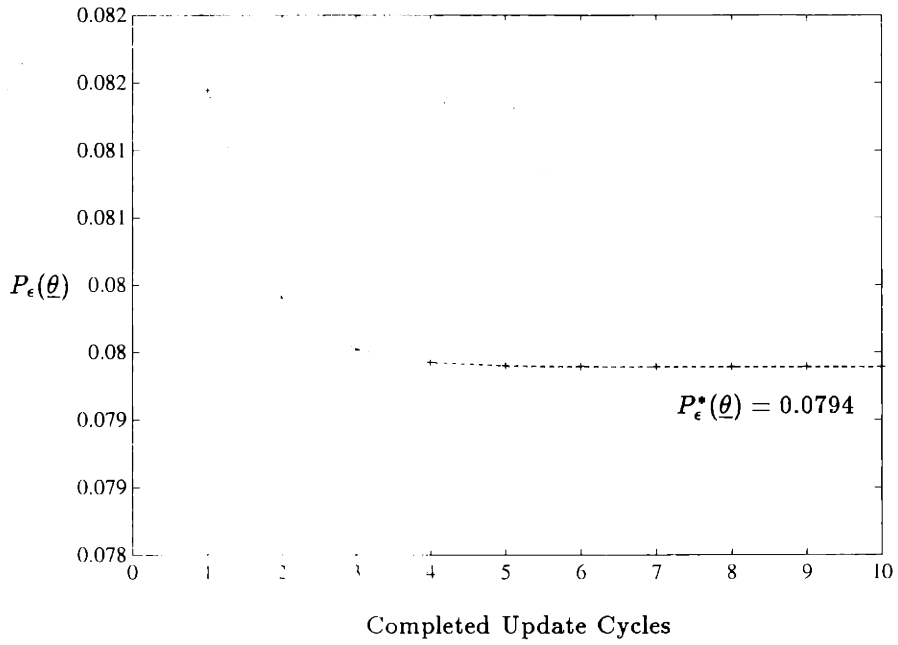


Figure 3-41: Probability of Error corresponding to 2-Tand Fixed-Point Iterations: Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$

at each DM have a single point of intersection, and the costs remain positive, there is a unique minimum solution along each coordinate, we can interpret the action of these Gauss-Seidel component solution iterations as determining a person-by-person optimal solution by solving a sequence of scaled intersection problems along each coordinate. These iterations on the necessary conditions will trace a path in threshold space identical to the path which would be taken by cyclic coordinate descent, about which we will have more to say in Chapter 5.

3.6 Chapter Conclusions

This chapter focused on the optimization of the decision rules using complete statistical information. The principal results were the following.

Several interpretations of the underlying optimization were provided. The first was a geometric view of the optimization as roughly a piecewise linear approximation to the optimal centralized decision hyperplane. The second point of view involved ex-

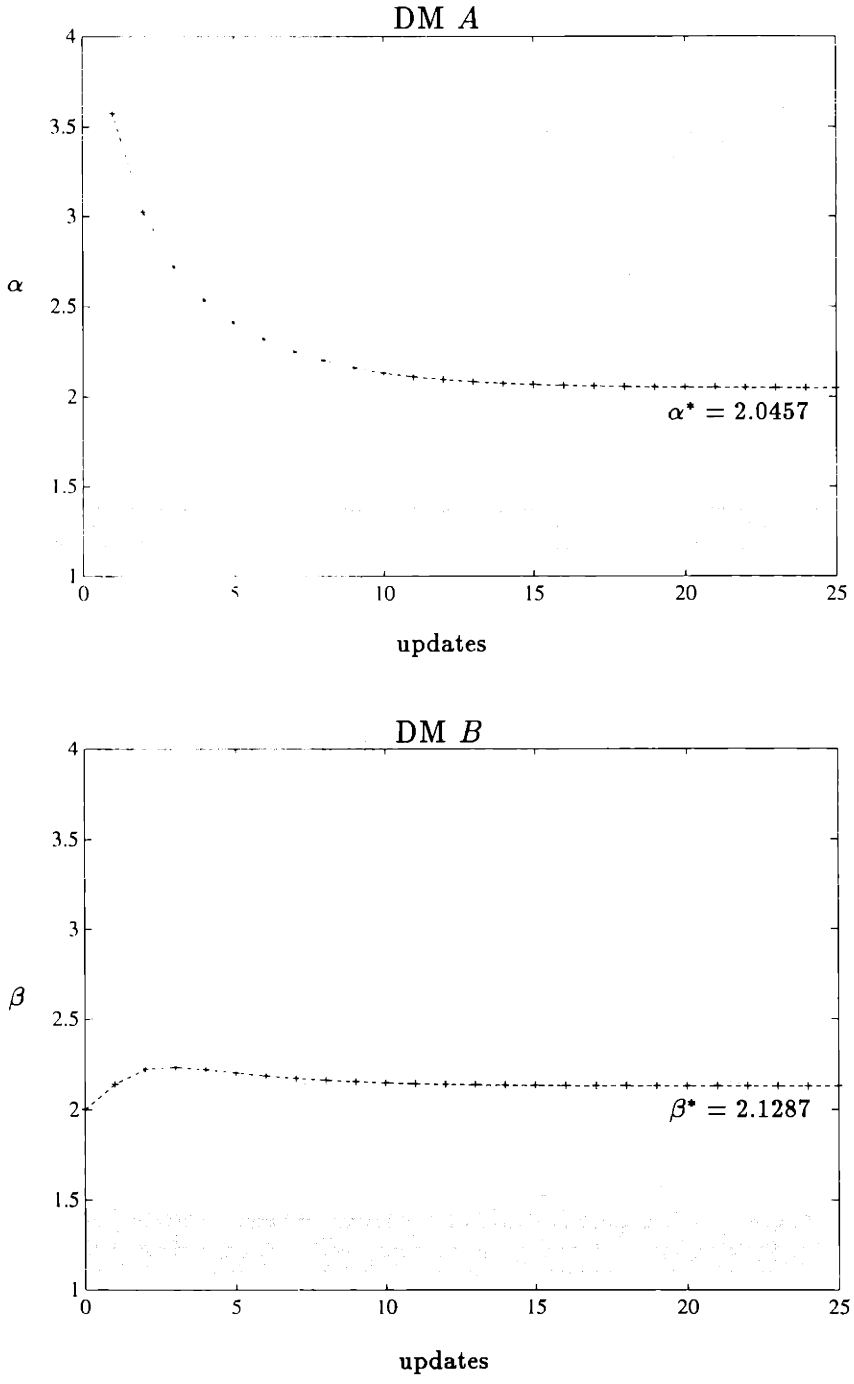


Figure 3-42: 3-Vee Fixed Point Iterations: Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = 1.5$, $\sigma_B^2 = 0.5$, $\sigma_C^2 = 1.0$, $p_0 = 0.75$.

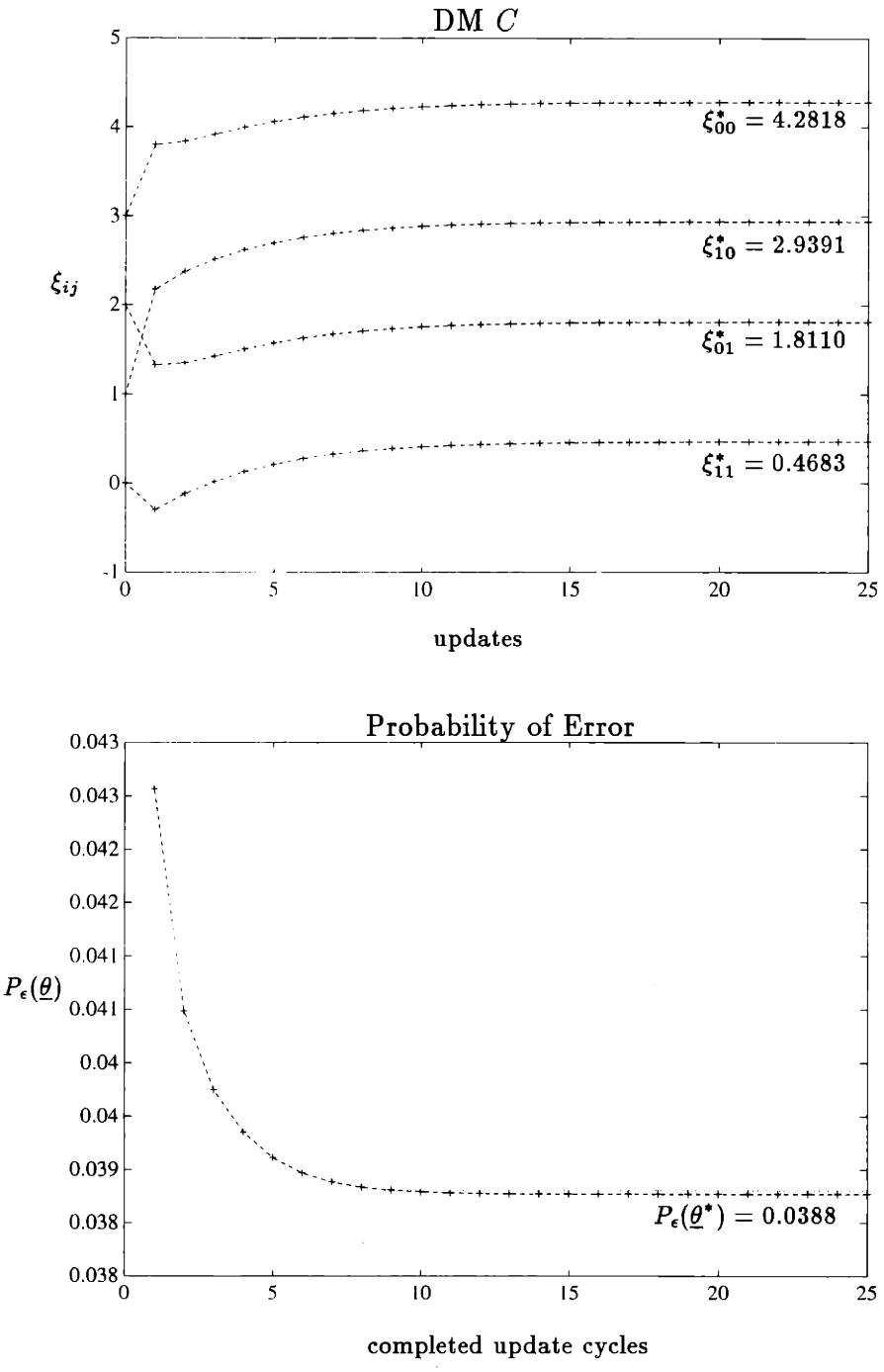


Figure 3-43: 3-Vee Fixed Point Iterations (cont'd): Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = 1.5$, $\sigma_B^2 = 0.5$, $\sigma_C^2 = 1.0$, $p_0 = 0.75$.

ploiting the conditionally independent observations and tree-type topology to display the decision process as an expanding sequential probability tree; this made immediately available the operating point formulation of the team error probability. The partial derivatives with respect to the network observation thresholds and LRTs were then easily derived. It also made clear that the analytic form of the coupling costs is determined solely by the topology of the network, while the current conditional operating points of the other network DMs determine the numerical value of the costs. Finally, the optimization was interpreted as a deterministic optimal control problem, and it was indicated how this formulation could be exploited to reduce the extensive communication needed to resolve the partial derivatives of each network DM.

A linear threshold parameterization of the decision rules was adopted, and the resulting Bayesian cost function analyzed, both through numerical experiments and analysis where admitted. The principal results were that under certain conditions on the conditional densities, namely continuity, twice differentiability, and bounded first and second derivatives, the resulting Bayesian cost also possessed nice differentiability and smoothness properties. Gaussian conditional densities were analytically shown to possess all of the desired properties.

Regarding stationary points, parameterization of the single DM with a linear threshold rule resulted in a Bayes cost function which was unimodal, with a single global minimum, provided that the conditional density functions possessed a single point of intersection. Such a property was shown to hold for the Gaussian detection problem, implying that for this case, a unimodal cost does result. On the negative side, the cost was clearly not a convex function of the observation threshold, and was seen to possess asymptotically flat regions which could cause difficulty for derivative methods.

These ideas carried over to the team problem, for which it was found that the global optimization, when viewed in a person-by-person manner, is such that the optimal value of each threshold corresponds to the point of intersection of the local scaled density functions, where the scaling is specified by the coupling costs. This lends another interpretation to the coupling costs as scaling on the conditional den-

sities of the local hypothesis test, which shift the point of intersection which defines the person-by-person optimal solution. If all of the local conditional densities have a single point of intersection, the cost under the linear threshold parameterization is unimodal with a single global minimum when considered one parameter at a time, provided that the scaling costs remain positive. This may help explain the observed numerical success of these methods [59], [27], [65] because it indicates that as a function of one parameter at a time, the problem possesses the same nice structure as the single DM problem. Unfortunately, global unimodality of the team cost remains an open question. This is the most severe setback for the application of gradient-based methods.

The fixed point iteration for directly determining a solution to the system of nonlinear equations for the network thresholds for the Gaussian detection problem gave another interpretation of the person-by-person optimization. The Gauss-Seidel component solution method applied to this system is equivalent to cyclic coordinate descent on the function itself, and numerical experiments suggest this technique often converges to a unique solution which appears globally optimal. This provides hope that mimicking this approach in the stochastic setting may prove successful.

Chapter 4

The Single DM Training Problem

This chapter begins consideration of the training problem, for which we have separated the discussion into four chapters. The present chapter investigates the training problem for the single DM (scalar parameter) case. We choose to present this material separately because all of the issues which arise for this problem also arise for the team problem, in addition to other issues which stem from the decentralization itself. However, in this chapter we wish to avoid a premature introduction of these difficulties. The present discussion introduces notation and also provides the theoretical building blocks necessary to make the extension to the network training algorithms in Chapter 5.

In the present chapter we introduce and motivate several training algorithms for the single DM problem, illustrating their respective behavior through numerical experiments. Our immediate purpose is simply to describe the methods. Convergence analysis is deferred to Chapter 6, where the single DM algorithms are handled in conjunction with the synchronous network algorithms of Chapter 5. We avoid at this time technical discussion concerning sufficient conditions for convergence, particularly with respect to the ranges of allowable stepsizes. At this point we will suggest typical choices, which we know a priori are sufficient for convergence.

Chapter 5 extends the material in this chapter to the network problem, where distributed versions of the training algorithms of this chapter are devised. Chapter 6 provides a unified convergence analysis of the algorithms of Chapters 4 and 5. Finally

Chapter 7 examines asynchronous implementations of the algorithms presented in Chapter 5, where such implementations are admitted.

4.1 Single DM Training Problem Statement

The training datum available to the DM at time k consists of a pair $\{Y_k, H^k\}$ where Y_k denotes the real-valued scalar random observation at time k, and H^k denotes the acting hypothesis corresponding to Y_k.[1] We may view H^k as a random variable for which a realization is obtained at each time k according to

$$H^k = \begin{cases} H_0 & \text{w.p. } p_0 \\ H_1 & \text{w.p. } p_1 \end{cases} \tag{4.1}$$

Then, if at time k it holds that $H^k = H_i$, $i = 0, 1$ the realization of Y_k is generated according to the corresponding conditional density $p_{Y_k|H_i}(y_k|H_i)$. Equivalently, H^k may be viewed as a label corresponding to the correct classification of observation Y_k, so that

$$Y_k \sim \begin{cases} p_{Y_k|H_1}(y_k|H_1) & \text{if } H^k = H_1 \\ p_{Y_k|H_0}(y_k|H_0) & \text{if } H^k = H_0 \end{cases} \tag{4.2}$$

An important point is that we assume the existence of prior probabilities and conditional densities according to which the data is generated, even though their functional form remains unknown.

Several potential technical difficulties in the training problem are avoided if we make the following assumptions on the conditional densities.

[1] We index time with a superscript rather than a subscript for H^k in order to avoid confusion with the standard notation H_0, H_1.

Assumption 4.1 (Conditional Density Functions)

The functions $p_{Y|H_0}(y|H_0), p_{Y|H_1}(y|H_1)$ *are:*

(a) *Continuous and twice differentiable, with bounded first and second derivatives*

(b) *Nonzero everywhere, i.e.,*

$$p_{Y|H_i}(y|H_i) > 0, \quad \forall y \in \Re, \; i = 0, 1 \tag{4.3}$$

Part (a) requires the conditional densities to possess the desirable properties already discussed in Chapter 3. Part (b) ensures that the hypothesis classes are not separable, and that the densities have infinite support, so that there is a nonzero probability of a realization of the observation at any point along the observation axis. These assumptions are made to avoid certain technical difficulties in proving convergence as described in Chapter 6.

For notational convenience, in the sequel we denote by X_k the pair $\{Y_k, H^k\}$ and refer to X_k as the *measurement*[2] at time k. Then the entire training data set consists of the sequence of measurements $\{Y_k, H^k; k = 1, 2, \ldots\}$. We make the following assumptions regarding this sequence.

Assumption 4.2 (Training Data)

For the sequence of measurements $\{Y_k, H^k; k = 1, 2, \ldots\}$ *it holds that*

(a) $\{Y_j, H^j\}$ *is independent of* $\{Y_k, H^k\}$ *for all times* $j \neq k$

(b)

$$p_{Y_k, H^k}(y_k, H) = p_{Y,H}(y, H), \quad k = 1, 2, \ldots \tag{4.4}$$

In other words, the sequence of measurements is independent and identically distributed (i.e., stationary).

[2]not to be confused with the term "observation" which we reserve exclusively for Y_k

Note that our assumptions on the training data assume the existence of a joint density function, not varying with time, according to which all the pairs are generated, so that the data set is consistent throughout. This implies that measurements corresponding to H_0 and H_1 appear in the data set in the ratio p_0/p_1. Note this also assumes the availability of an infinite number of sample pairs. We do not address questions which arise in conjunction with finite training sets in this report.

The first basic training problem that we pose is the problem of placing an observation threshold to minimize the probability of error[3]. To meaningfully formulate this problem, it is necessary to make the additional assumption that we know a priori the relative spatial relations of the hypotheses. In particular, we must know whether the assignment H_0 corresponds to values of the observation Y which generally lie above or below values of Y generated in accordance with H_1. Thus we assume that H_0 corresponds to values of the observation generally lying below (to the left on the real axis) of the values corresponding to H_1.

Problem 4.1 (Single DM, Minimum Error Training Problem)
Given only a measurement sequence $\{Y_k, H^k; k = 1, 2, \ldots\}$ *satisfying Assumption 4.2, determine the minimum probability of error decision rule* γ^* *over the class of linear threshold classifiers* $\mathcal{T}$

$$\gamma^* = \arg \min_{\gamma \in \mathcal{T}} P_\epsilon(\gamma) \tag{4.5}$$

Equivalently, find

$$\theta^* = \arg \min_{\theta \in \Re} P_\epsilon(\theta) \tag{4.6}$$

where $P_\epsilon(\theta) : \Re \mapsto \Re$ *is the function*

$$P_\epsilon(\theta) = p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\, dy + p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\, dy \tag{4.7}$$

[3]We first present the minimum error solution rather than the general Bayes solution for simplicity in introducing the concepts.

The solution to this training problem will in general be suboptimal with respect to the best performance achievable due to the restriction to a linear threshold rule. In particular, the optimal decision rules are not linear threshold rules for all binary hypothesis tests.

Although we assume the existence of a deterministic cost function P_ϵ we wish to optimize, it is in general unspecified, or at best only partially specified. In particular, for Problem 4.1 the unspecified information is the conditional densities $p_{Y|H_i}(y|H_i), i = 0, 1$ and the prior probabilities $p_i, i = 0, 1$. Hence, a nonparametric training algorithm is indicated. The training data must be used to overcome the lack of information, but enough information is only acquired asymptotically, and thus the best we can hope for is that our algorithms provide the optimal solution in the limit as infinitely many measurements are examined. It should not be surprising that the training algorithms we present are adaptations of techniques for determining the optimal thresholds in the presence of full statistical information. These techniques are commonly referred to in the literature as *stochastic optimization techniques.*

We wish to point out that there is an entire class of approaches with which we do not concern ourselves here. These approaches involve performing density estimation, that is estimating the distribution of the data, and then proceeding as if the statistics were known. The reason that these techniques are not of interest to us is not that they would not have been effective. For example, since the class of Gaussian distributions admits parameterization by two scalar parameters, mean and variance, a reasonable approach, in the case that Gaussian statistics appear to be a good model for the problem, would be to compute the sample means and variances at each node, obtain estimates of p_0 and p_1, and use these estimated densities in the LRT's. However, since our goal is to investigate *nonparametric* learning and adaptation, we have avoided such approaches.

4.2 Iterative Stochastic Gradient Algorithms

4.2.1 Preliminaries

As was discussed in Chapter 3, if the derivative of the function $P_\epsilon(\theta)$ exists and is known precisely, well-known iterative gradient-based optimization techniques can be applied to determine a stationary point, i.e., a value of θ satisfying the necessary condition

$$\frac{dP_\epsilon}{d\theta}(\theta) = 0 \tag{4.8}$$

where

$$\frac{dP_\epsilon}{d\theta}(\theta) = -p_0 p_{Y|H_0}(\theta|H_0) + p_1 p_{Y|H_1}(\theta|H_1) \tag{4.9}$$

For example, the method of steepest descent

$$\theta_{k+1} = \theta_k - \rho_k \frac{dP_\epsilon}{d\theta}(\theta_k), \quad k = 1, 2, \ldots \tag{4.10}$$

where either $\rho_k = \rho$ is a constant positive stepsize, or $\{\rho_k\}$ is a sequence of positive step size or gain parameters, could be used to iteratively determine a solution to (4.8), provided certain conditions are met.

Although we don't have precise information about the function $P_\epsilon(\theta)$, it is possible, under certain conditions, to do something analogous. Suppose that we can compute at each time k a random variable Z_k, which depends on the current value of the threshold θ_k and the current measurement $X_k = \{Y_k, H^k\}$, such that

$$\frac{dP_\epsilon}{d\theta}(\theta_k) = E_{X_k}\{Z_k(X_k, \Theta_k)|\Theta_k = \theta_k\} \tag{4.11}$$

Then, it may be possible to find a value of θ satisfying (4.8) using the iterative stochastic successive approximation algorithm

$$\Theta_{k+1} = \Theta_k - \rho_k Z_k(Y_k, H^k, \Theta_k), \quad k = 1, 2, \ldots \tag{4.12}$$

provided that certain conditions which we discuss in the following are met, and pro-

vided we specify what we mean by "find". This modified steepest descent algorithm is a recursion described by a stochastic difference equation, and produces a sequence of random thresholds $\{\Theta_k\}$. The hope is that the sequence can be made to migrate, on the average, in the direction of steepest descent, and converge in some probabilistic sense to a value which satisfies (4.8).

4.2.2 The General Methodology

The measurement X_k represents the on-line information required for computing Z_k at time k. Thus far we have indicated that the measurement X_k consists of a single $\{Y_k, H^k\}$ pair, although in general the step Z_k could depend on past pairs as well. For example, this would occur if computation of Z_k involved some averaging over past data. Since we won't focus on such techniques, we will use the notation

$$\bigcup_{i=1}^{k-1} \{Y_i, H^i\} \tag{4.13}$$

to indicate explicitly when more than one measurement is involved.

A notational reminder is that when we write $E_X\{\cdot\}$ this means $E_{Y,H}\{\cdot\}$. For example,

$$\begin{aligned} E_{X_k}\{Z_k(X_k, \Theta_k)|\Theta_k\} &= E_{Y_k,H^k}\{Z_k(Y_k, H^k, \Theta_k)|\Theta_k\} \\ &= E_{H^k}\{E_{Y_k}\{Z_k(Y_k, H^k)|H_k, \Theta_k\}|\Theta_k\} \end{aligned} \tag{4.14}$$

The intermediate random variable Z_k normally arises as the result of some deterministic processing of the measurement X_k.[4] Thus, realizations of Z_k will depend on the measurement X_k and the parameter Θ_k. Figure 4-1 indicates the relationship between the random variables H, Y, X and Z. A step Z_k is assumed to be generated with every arrival of a measurement X_k, meaning that the indexing of measurements coincides with the indexing of updates. Thus, we use the time variable k as a common

[4]Thus, to be strictly correct we should write $Z = f(X_k, \Theta_k)$ where f is a deterministic function, or simply Z, but *not* $Z(X_k, \Theta_k)$ which is in fact what we write. We do this in order to remind the reader that all the quantities involved are random variables.

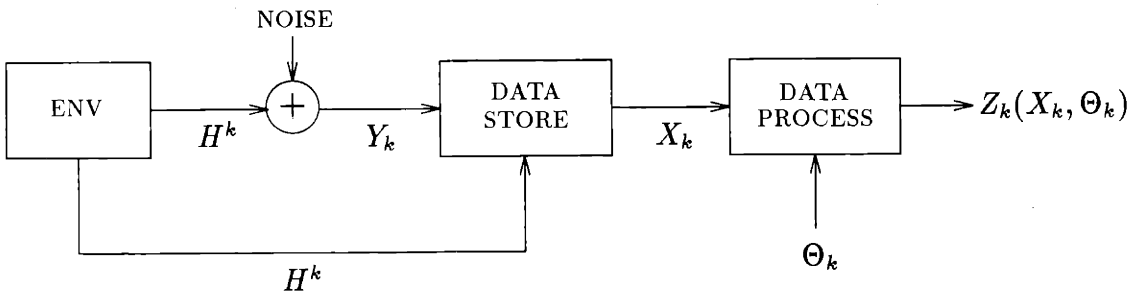


Figure 4-1: Data Processing

index for both.

There are a variety of alternative stochastic algorithms suitable for minimizing imprecisely known functionals, all of which can be expressed in the general form

$$\Theta_{k+1} = \Theta_k - \rho_k Z_k(X_k, \Theta_k), \quad k = 1, 2, \ldots \tag{4.15}$$

where Z_k is some approximation to the derivative of the cost with respect to θ evaluated at θ_k. The various algorithms differ on the construction of the intermediate random variable Z. In this report we consider variations of the two well-known classes initiated by Robbins and Monro [53] and Kiefer and Wolfowitz [30].

- *Robbins-Monro (RM) Setting:* Robbins-Monro type algorithms correspond to the example above, namely stochastic generalizations of the well-known gradient descent method. The algorithm was originally developed as a stochastic counterpart to successive approximation for determining the roots of equations of the form

$$f(x) = a \tag{4.16}$$

where $f(x) : \Re \mapsto \Re$ is such that

$$f(x) = E\{Y|X = x\} = \int_{-\infty}^{+\infty} y dP(y|x) \tag{4.17}$$

and only the random observations Y are available to the experimenter. When

$$f(x) = \frac{dF}{dx}(x) \tag{4.18}$$

and $a = 0$, the technique corresponds to gradient descent.

Application of the RM technique requires a stochastic realization of the gradient. So in order to apply the technique to Problem 4.1, there must exist a random variable Z such that (4.11) holds. One possibility for determining such a Z is to try and derive it from a second random variable Q for which it holds that

$$E_X\{Q(X,\Theta)|\Theta = \theta\} = P_\epsilon(\theta),\ \forall\theta \in \Re \tag{4.19}$$

This is possible if it is true that

$$\frac{dP_\epsilon}{d\theta}(\theta) = \frac{d}{d\theta}E_X\{Q(X,\Theta)|\Theta = \theta\} = E_X\left\{\frac{d}{d\theta}Q(X,\Theta)|\Theta = \theta\right\},\ \forall\theta \in \Re \tag{4.20}$$

In other words, a differentiable unbiased estimator of the function we wish to minimize can be used to generate Z, as long as the indicated interchange of differentiation and expectation is valid. Of course, there may be alternative ways of deriving a suitable Z which do not depend on this property.

A function $f(x)$ which is expressible as the expectation of some random variable depending on the data and the parameter x is referred to in the statistics literature as a *regression function.* Thus, application of the RM technique to Problem 4.1 requires that $dP_\epsilon(\theta)/d\theta$ be a regression function.

- *Kiefer-Wolfowitz (KW) Setting:* In the Kiefer-Wolfowitz setting, only stochastic realizations of the function itself are available. There is a function Q such that

$$E_X\{Q(X,\Theta)|\Theta = \theta\} = P_\epsilon(\theta),\ \forall\theta \in \Re \tag{4.21}$$

but Q may not be differentiable as a function of θ. Another possibility is to estimate $dP_\epsilon(\theta)/d\theta$ using a finite-difference approximation based on the samples

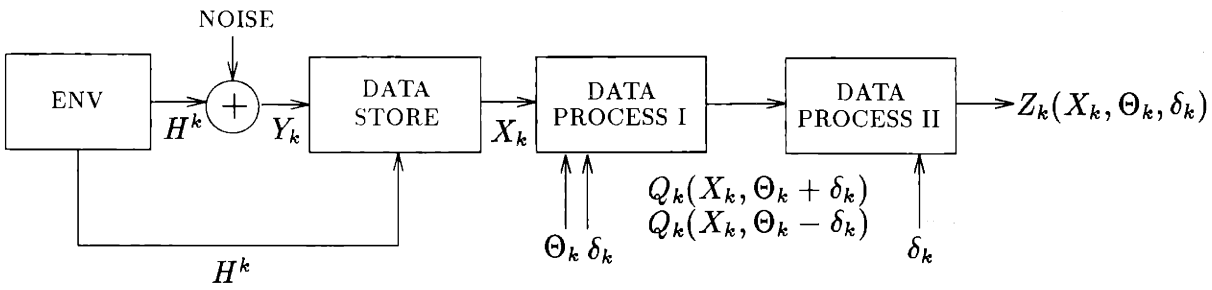


Figure 4-2: Data Processing for KW Setting

Q, for example using the central (two-sided) finite difference

$$Z(X, \theta, \delta) = (Q(X, \theta + \delta) - Q(X, \theta - \delta))/2\delta \tag{4.22}$$

or the forward (one-sided) finite difference

$$Z(X, \theta, \delta) = (Q(X, \theta + \delta) - Q(X, \theta))/\delta \tag{4.23}$$

where δ is a positive real-valued scalar which is allowed to decrease. The same realization of X may be used to generate both realizations of Q, or distinct realizations of X may be used to generate each perturbed realization of Q, although these choices may result in different rates of convergence [56]. This method is the well-known Kiefer-Wolfowitz stochastic approximation technique for optimizing a regression function [30]. Techniques of this variety have been referred to by Tsypkin [69], [70] and Polyak and Tsypkin [50] as "search algorithms". The data processing for this technique is depicted in Figure 4-2, and can be seen to require an additional data processing stage as compared to the RM technique.

Because the KW method is essentially an approximation to the RM technique, its convergence properties are in general inferior. Furthermore, since two samples of the function (realizations of Q) must be obtained to generate a single value of Z, the technique involves more computational overhead than the RM technique and may also require more on-line observations, depending on the

implementation. For these reasons it is generally preferable to apply the RM technique if possible. However, there are situations for which a direct realization of the gradient is simply not available, or for which the RV $\frac{d}{d\theta}Q(X,\theta)$ is not well-behaved, i.e., the interchange of integration and differentiation suggested in (4.20) is not valid. In these cases, the KW technique provides a viable alternative.

4.2.3 Comments

We defer convergence analysis of the scalar-parameter algorithms of this chapter until after their multidimensional counterparts are presented in Chapter 5. Then, in Chapter 6, we will present a single method of proof of convergence for all of the algorithms, both for the scalar and vector case. The method of proof utilizes the fact that all of the algorithms possess a generalized stochastic descent property. The analysis we provide relies heavily on the use of results on the convergence properties of martingale sequences and their various extensions.

Although we defer convergence analysis of the algorithms of this chapter until Chapter 6, it is worth commenting at this juncture on the type of conditions which are required to establish convergence. Convergence analysis generally amounts to establishing that certain sufficient conditions for convergence be satisfied. These conditions involve restrictions on the function being optimized, the noise properties of the measurement sequence, the step generation technique of the algorithm, and the choice of update stepsize. The first two types of conditions are generally not under the algorithm designer's control, so it must be established that these properties are in fact satisfied in the application of interest. The second two properties do come under the designer's control, so that a satisfactory algorithm may usually be constructed if the conditions on the function and the measurement sequence are satisfied.

The purpose of the extensive development in Chapter 3 was to establish that the required properties on the function are satisfied for the linear threshold parameterization of the Bayes cost. Assumption 4.2 is sufficient to ensure that the necessary requirements on the measurement sequence are satisfied.

It turns out that we have already established, in Chapter 3, all the conditions on the function $P_\epsilon(\theta)$ and its derivatives that are required to prove convergence with probability one[5] of the algorithms of this chapter to the globally optimal solution for a broad class of conditional densities. That is, under some mild assumptions on the conditional densities, we will be able to guarantee that the optimal threshold value is determined w.p.1.

4.3 Modified Robbins-Monro or Window (WIN) Training Algorithm

In this section, we begin our investigation of the scalar parameter training problem by attempting to apply the RM technique to optimize $P_\epsilon(\theta)$. Recall that the first step in applying the RM technique is to determine a stochastic realization of $dP_\epsilon/d\theta$. In other words, we seek a random variable Z such that

$$E_X\{Z(X,\Theta)|\Theta=\theta\} = \frac{dP_\epsilon}{d\theta}(\theta),\ \forall\theta\in\Re \tag{4.24}$$

Recall from Chapter 3 that

$$\frac{dP_\epsilon}{d\theta}(\theta) = -p_0 p_{Y|H_0}(\theta|H_0) + p_1 p_{Y|H_1}(\theta|H_1) \tag{4.25}$$

An equivalent expression is

$$\frac{dP_\epsilon}{d\theta} = -\int_{-\infty}^{+\infty}\delta(y-\theta)p_{Y|H_0}(y|H_0)dy + \int_{-\infty}^{+\infty}\delta(y-\theta)p_{Y|H_1}(y|H_1)dy \tag{4.26}$$

where $\delta(\cdot)$ denotes the delta-Dirac function.

It is difficult to determine a suitable Z for this derivative because of the exact function evaluations it requires. The source of these exact function evaluations is the

[5]Notions of convergence for sequences of random variables are discussed briefly in Appendix A

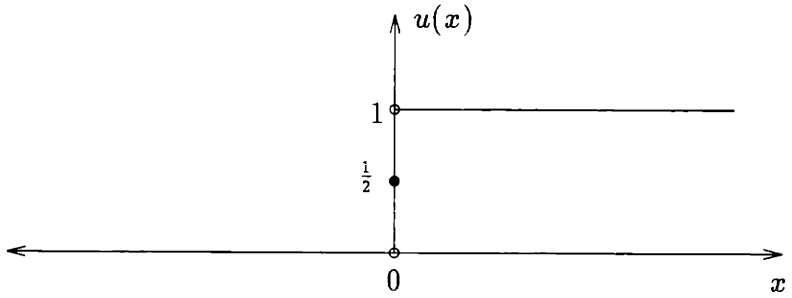


Figure 4-3: Hard Limiting 0-1 Threshold

hard-limiting threshold rules employed in the parameterization. Recall that

$$P_\epsilon(\theta) = p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{4.27}$$

which can be equivalently expressed as

$$P_\epsilon(\theta) = p_1 + p_0 \int_{-\infty}^\infty u(y-\theta)p(y|H_0)\,dy - p_1 \int_{-\infty}^{+\infty} u(y-\theta)p(y|H_1)\,dy \tag{4.28}$$

where $u(x)$ is the unit step function defined by

$$u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0.5 & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{4.29}$$

and shown in Figure 4-3. The derivatives of the unit step function give rise to the delta Dirac functions in (4.26).

The problem is evident from another point of view. Define the random penalty functional

$$Q(X,\Theta) = \begin{cases} 1 & \text{if } y > \theta \text{ and } H = H_0 \\ 1 & \text{if } y < \theta \text{ and } H = H_1 \\ 0 & \text{else} \end{cases} \tag{4.30}$$

This function assigns unit penalty to each type of error, regardless of the actual

location of y relative to θ. It is clear that

$$E_X\{Q(X,\Theta)|\Theta=\theta\} = P_\epsilon(\theta),\ \forall\theta\in\Re \tag{4.31}$$

so that the criterion $P_\epsilon(\theta)$ is the expectation of the empirical cost Q. If we now define the labeling random variable

$$\tilde{Q}(X) = \begin{cases} -1 & \text{if } y \text{ from } H_0 \\ +1 & \text{if } y \text{ from } H_1 \end{cases} \tag{4.32}$$

then, with $u(x)$ the unit step as defined above, we can express the probability of error in the alternate form

$$P_\epsilon(\theta) = p_1 - E_{Y,\tilde{Q}}\{\tilde{Q}u(Y-\Theta)|\Theta=\theta\} \tag{4.33}$$

Since all the θ dependence is captured in the argument of $u(\cdot)$,

$$\frac{dP_\epsilon}{d\theta}(\theta) = E_{Y,\tilde{Q}}\{\tilde{Q}\delta(Y-\Theta)|\Theta=\theta\} \tag{4.34}$$

and we arrive at the same difficulty. Thus, the only obvious choice of empirical cost is nondifferentiable at θ, so that is not clear that a suitable regression for the derivative can be determined via (4.20).

The above discussion indicates that no unbiased realization of the derivative appears to exist, i.e., it appears that it is not possible to directly measure the derivative. However, the discussion also suggests a potential solution to this problem. The mathematical difficulties introduced by the hard limiting threshold would certainly disappear if we could replace $u(x)$ by a function which was differentiable at zero. This suggests that it may be possible to define a sequence of differentiable functions which converge pointwise to $u(\cdot)$, and then use these approximations to derive a sequence of derivative estimates converging pointwise to the true derivative of $P_\epsilon(\theta)$. We could then employ an RM technique using this sequence of estimated derivatives. For this approach to be viable, it is essential that the algorithm be able to tolerate the result-

ing nonzero bias in the gradient measurement, which can be made to disappear only asymptotically.

We wish to point out that this idea is not new. The inherent difficulties with non-parametric optimization of linear thresholds to minimize the Bayes risk have been addressed by Do-Tu and Installe in [14], and Wassel and Sklansky in [74],[73], [57]. A reasonable question which comes to mind in view of these mathematical difficulties is: Why have we chosen this criterion and not some easier alternative, such as mean-squared-error? Most training algorithms employed by neural network researchers, for example, are steepest descent algorithms for minimizing a mean-square-error (MSE) criterion. Realizations of the criterion are given by the sum of the squares of differences between the actual and desired outputs over the set of training patterns. Each network weight is incrementally adjusted in the direction of the derivative of the MSE with respect to the weight. The attractiveness of this criterion results from the simple form of its gradient, for which a suitable estimator is readily available. However, it is clear that minimizing the MSE does *not*, in general, minimize the probability of error [57], [73], [74]. In fact, the two are only equivalent for the special case of equal prior probabilities. Thus, if one desires a truly minimum error solution, the associated mathematical difficulties must be addressed. The minimum error problem for classification in the neural network setting has been addressed by several researchers [2], [39].

We now return to the suggestion of replacing the hard-limiting threshold with an approximation which is differentiable at the origin. To illustrate the underlying effect of this on the training problem, we make the following definitions. Let $\hat{u}(x,\delta)$ be an approximation of $u(x)$ such that

$$\lim_{\delta \to 0} \hat{u}(x,\delta) = u(x) \tag{4.35}$$

pointwise, and define $\hat{u}(x,\delta)$ from a function h through the integral relationship

$$\hat{u}(x,\delta) = \int_{-\infty}^{x} h(s,\delta)ds \tag{4.36}$$

where h is such that

$$\lim_{\delta \to 0} h(x, \delta) = \delta(x) \tag{4.37}$$

$$\int_{-\infty}^{\infty} h(s, \delta) ds = 1, \quad \forall \delta \geq 0 \tag{4.38}$$

The function h is referred to as a "window function" for reasons that will become clear shortly. It must satisfy some additional conditions to ensure convergence of the training algorithm, but we defer discussion of these issues to Chapter 6. The correspondence between several useful $\hat{u}(x, \delta)$, $h(s, \delta)$ pairs is depicted in Figure 4-4. Notice that δ provides a measure of the window width. The various windows differ on how errors on observations which fall near the threshold are penalized. The function $u(\cdot)$ penalizes all errors equally, regardless of their distance from the threshold, while the others have lower penalties for observations which are erroneously classified, but which fall close to the threshold. It remains an open question which of these choices results in the best performance, although clearly the rectangular window requires the least computation and is the easiest to implement. For more discussion of such issues see [2].

If one of the approximating functions $\hat{u}(x, \delta)$ is substituted for $u(x)$ in expression (4.33), we obtain the modified performance measure

$$\hat{P}_\epsilon(\theta, \delta) = p_1 - E_{Y,\tilde{Q}}\{\tilde{Q}\hat{u}(Y - \Theta, \delta) | \Theta = \theta\} \tag{4.39}$$

with derivative

$$\frac{\partial \hat{P}_\epsilon}{\partial \theta}(\theta, \delta) = E_{Y,\tilde{Q}}\{\tilde{Q}h(Y - \Theta, \delta) | \Theta = \theta\} \tag{4.40}$$

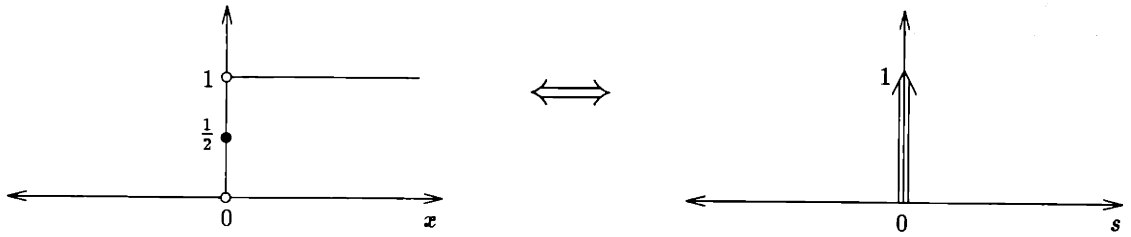
so that $\tilde{Q}h(Y - \Theta, \delta)$ is an unbiased estimate of the derivative of the modified performance measure (4.39). It is shown in Chapter 6 that the modified performance measure is related to the true performance measure by

$$\hat{P}_\epsilon(\theta, \delta) = \int_{-\infty}^{+\infty} P_\epsilon(\theta + s) h(s, \delta) ds \tag{4.41}$$

so that the modified performance measure may be interpreted as the original per-

$$u(x) = \begin{cases} 1 & \text{if } x > 0 \\ .5 & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

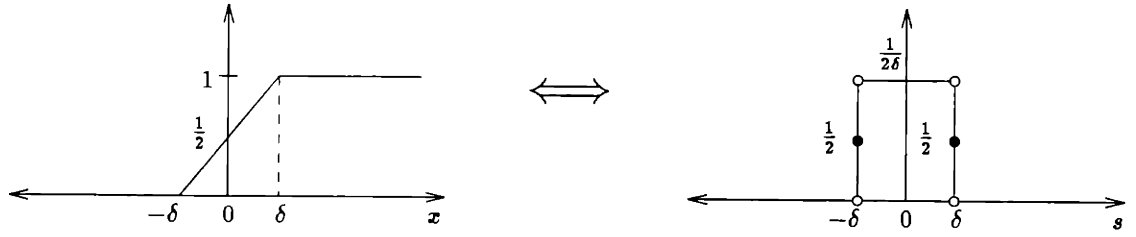
$$h(s,\delta) = \delta(s)$$



(a)

$$\hat{u}(x,\delta) = \begin{cases} 0 & \text{if } x < -\delta \\ (\frac{1}{2\delta})x + \frac{1}{2} & \text{if } -\delta \le x \le \delta \\ 1 & \text{if } x > \delta \end{cases}$$

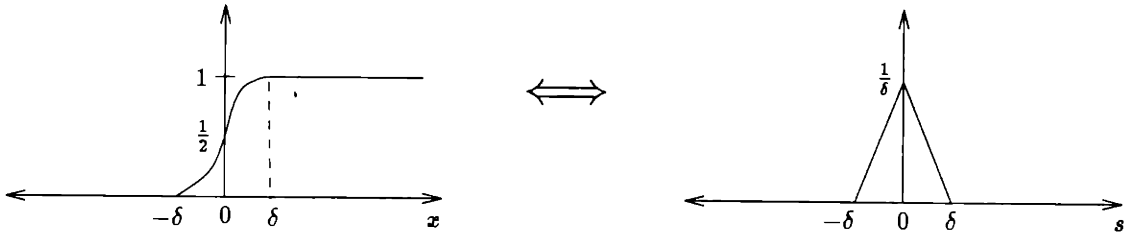
$$h(s,\delta) = \begin{cases} \frac{1}{2\delta} & \text{if } |s| \le \delta \\ 0 & \text{else} \end{cases}$$



(b)

$$\hat{u}(x,\delta) = \begin{cases} 0 & \text{if } x < -\delta \\ +\frac{1}{2\delta^2}(x+\delta)^2 & \text{if } -\delta \le x < 0 \\ -\frac{1}{2\delta^2}(x-\delta)^2 + 1 & \text{if } 0 < x \le \delta \\ 1 & \text{if } x > \delta \end{cases}$$

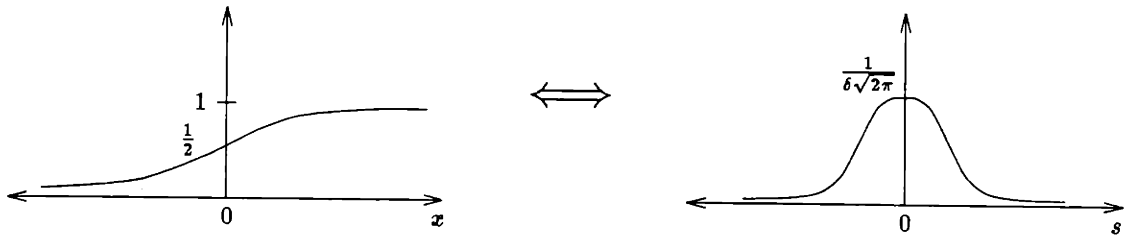
$$h(s,\delta) = \begin{cases} +\frac{1}{\delta^2}s + \frac{1}{\delta} & \text{if } -\delta \le s \le 0 \\ -\frac{1}{\delta^2}s + \frac{1}{\delta} & \text{if } 0 \le s \le \delta \\ 0 & \text{else} \end{cases}$$



(c)

$$\hat{u}(x,\delta) = \int_{-\infty}^{x} \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{s^2}{2\delta^2}}\, ds$$

$$h(s,\delta) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{s^2}{2\delta^2}}$$



(d)

Figure 4-4: Several useful u, h pairs. (a)Delta-Dirac window, (b) rectangular window, (c) triangular window, (d) Gaussian window

formance measure smoothed by the window function h. Equations (4.37) and (4.41) together imply that

$$\lim_{\delta \to 0} \hat{P}_\epsilon(\theta, \delta) = P_\epsilon(\theta) \tag{4.42}$$

Furthermore, the derivatives are similarly related by

$$\frac{\partial \hat{P}_\epsilon}{\partial \theta}(\theta, \delta) = \int_{-\infty}^{+\infty} \frac{dP_\epsilon}{d\theta}(\theta + s) h(s, \delta) ds \tag{4.43}$$

and

$$\lim_{\delta \to 0} \frac{\partial \hat{P}_\epsilon}{\partial \theta}(\theta, \delta) = \lim_{\delta \to 0} E_{Y,\tilde{Q}}\{\tilde{Q} h(Y - \Theta, \delta)\} = \frac{dP_\epsilon}{d\theta}(\theta) \tag{4.44}$$

so that the approximation in (4.40) converges to an unbiased estimate of the gradient of $P_\epsilon(\theta)$. This fact led Wassel and Sklansky in [73] to refer to these algorithms as modified RM algorithms, and interpret the action of the algorithms as operating on a sequence of regression functions which converge to the true derivative, which itself is not a regression function. In this manner, the windowing technique can be used to extend the applicability of the RM technique to those functions which may be expressed as the limit of a sequence of regression functions.

The so-called window algorithms for adapting a linear threshold rule to minimize the probability of error are of the general form

$$\Theta_{k+1} = \Theta_k - \rho_k Z_k(X_k, \Theta_k, \delta_k) \tag{4.45}$$

where the step Z computed as

$$\begin{aligned} Z_k(X_k, \Theta_k, \delta_k) &= \tilde{Q}_k h(Y_k - \Theta_k, \delta_k) \\ &= \begin{cases} -h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ +h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{4.46}$$

with $\{\rho_k\}$ and $\{\delta_k\}$ positive real-valued sequences decreasing toward zero. This yields

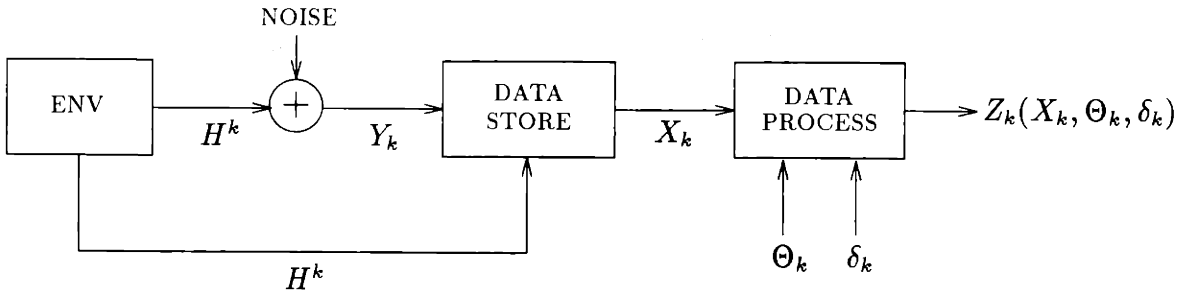


Figure 4-5: Data Processing WIN

an algorithm of the form

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ \theta_k - \rho_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{4.47}$$

The data processing required by the window algorithms is indicated in Figure 4-5. Notice that since

$$\lim_{\delta \to 0} h(0, \delta) = \infty \tag{4.48}$$

the steps Z can become unbounded in the limit. Under suitable conditions on the relationship between the sequences $\{\rho_k\}$ and $\{\delta_k\}$ the algorithm can still be proven to converge. However, Wassel and Sklansky in [73] investigate a related alternative choice of step Z_k given by

$$\begin{aligned} Z_k(X_k, \Theta_k, \delta_k) &= 2\delta_k \tilde{Q}_k h(Y_k - \Theta_k, \delta_k) \\ &= \begin{cases} -2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ +2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{4.49}$$

which yields the algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k 2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ \theta_k - \rho_k 2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{4.50}$$

For the choice of Z_k indicated in (4.49) it holds that

$$\frac{\partial \hat{P}_\epsilon}{\partial \theta}(\theta, \delta) = \frac{1}{2\delta} E_{Y,\tilde{Q}}\{2\delta \tilde{Q} h(Y - \Theta, \delta) | \Theta = \theta\} \tag{4.51}$$

The convergence properties which result from using the modified steps (4.49) in algorithm (4.45) rather than the steps (4.46) are generally superior, as we argue in Chapter 6. This is essentially a result of the normalization effect of the additional division by δ. Furthermore, the algorithm can be shown to converge under less restrictive conditions on the stepsize sequences.

As an example, steps for the window algorithm with no normalization and a rectangular window are of the form

$$Z_k(X_k, \Theta_k, \delta_k) = \begin{cases} -\frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +\frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.52}$$

giving rise to the algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k \frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ \theta_k & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - \rho_k \frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.53}$$

For the normalized case and rectangular window, steps are of the form

$$Z_k(X_k, \Theta_k, \delta_k) = \begin{cases} -1 & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +1 & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.54}$$

leading to the algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ \theta_k & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - \rho_k & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.55}$$

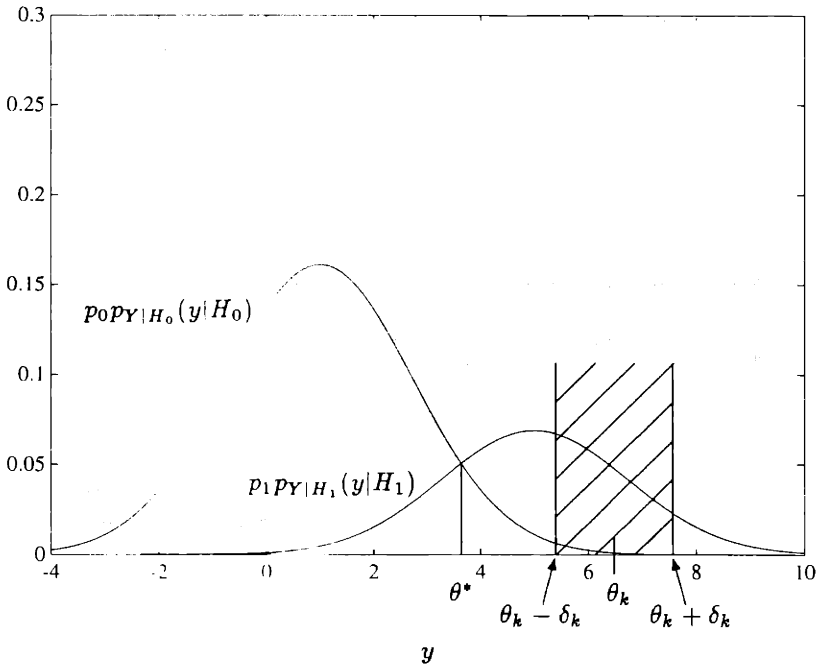


Figure 4-6: Window Algorithm using Rectangular Window

Operation of these algorithms is depicted in Figure 4-6 for the Gaussian detection problem.

There are several noteworthy properties of the window algorithms which are not specific to the rectangular window, but hold for general choices of window. First, corrections to the current threshold are made up or down depending on whether or not an observation corresponding to a particular acting hypothesis falls within the window, and do not depend on which side of the threshold the observation falls. This means that the actual decisions are not explicitly required in the training process. The effect of the windowing is to rely only on those observations which fall near the threshold during the training process. Furthermore, since the width of the window is continually shrinking, observations must fall closer and closer to the threshold as time advances in order to impact the adjustment of the threshold. Second, the solution is completely nonparametric. The functional forms of the densities are not required, and neither are the prior probabilities. Since they were assumed to be active in the creation of the measurement data set, the necessary information can be inferred by the algorithm. No explicit modeling of the densities or prior probabilities is required.

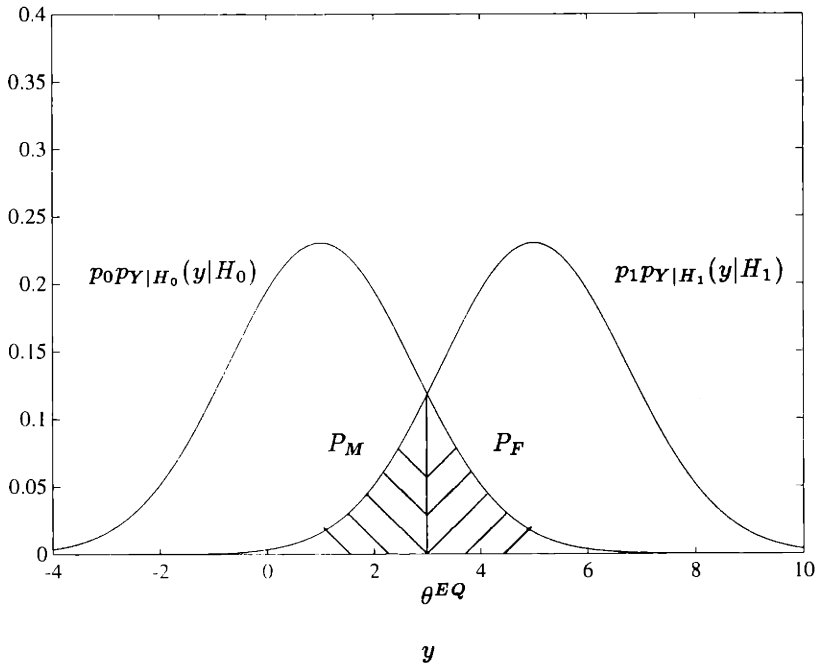


Figure 4-7: Equal Error Point

Special Case: Equal Error Training

An important special case occurs when the prior probabilities are equal. For this case, determining the root of (4.25) is equivalent to finding a value of θ such that the probabilities of false alarm and miss are equal. Such a point is known as the equal error point. It is shown in Figure 4-7. The equal error point can be determined by a simple error correction scheme which simply adapts the threshold up and down until the proportions of the two types of error become equal. It is optimal only in the case of equal prior probabilities.

This problem admits the following particularly simple choice of Z

$$Z_k(X_k, \Theta_k) = \begin{cases} -1 & \text{if } y_k > \theta_k,\ H^k = H_0 \\ +1 & \text{if } y_k \leq \theta_k,\ H^k = H_1 \end{cases} \tag{4.56}$$

leading to the algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k & \text{if } y_k > \theta_k,\ H^k = H_0 \\ \theta_k - \rho_k & \text{if } y_k \leq \theta_k,\ H^k = H_1 \end{cases} \tag{4.57}$$

Note that for this choice of Z, the actual values of the decisions have become relevant, that is, it is now significant whether the DM decided hypothesis H_0 or H_1. Of course, the equal error problem may also be solved by any of the window algorithms of the previous section, but those approaches necessitate the introduction of the additional sequence $\{\delta_k\}$ and have additional computational overhead.

4.3.1 Unequal Cost Version

Window algorithms are easily adapted to handle unequal costs on the classification errors such as arise in the general Bayes risk formulation

$$J_B = \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\, dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\, dy \tag{4.58}$$

where λ_0 and λ_1 are finite real-valued positive constants. The derivative of this cost function is given by

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 p_{Y|H_0}(\theta|H_0) + \lambda_1 p_1 p_{Y|H_1}(\theta|H_1) \tag{4.59}$$

We illustrate the development for the unnormalized steps. In an analogous fashion to the preceding discussion, we can define the random variable

$$Q_B(X, \lambda_0, \lambda_1, \Theta) = \begin{cases} \lambda_0 & \text{if } y > \theta \text{ and } H = H_0 \\ \lambda_1 & \text{if } y < \theta \text{ and } H = H_1 \\ 0 & \text{else} \end{cases} \tag{4.60}$$

With this choice of Q_B it holds that

$$E_X\{Q_B(X, \lambda_0, \lambda_1, \Theta)|\Theta = \theta\} = J_B(\theta),\ \forall \theta \in \Re \tag{4.61}$$

If we define the labeling random variable

$$\tilde{Q}_B(X, \lambda_0, \lambda_1) = \begin{cases} -\lambda_0 & \text{if } H^k = H_0 \\ +\lambda_1 & \text{if } H^k = H_1 \end{cases} \tag{4.62}$$

then, we can express the Bayes cost in the alternate form

$$J_B(\theta) = \lambda_1 p_1 - E_{Y,\tilde{Q}_B}\{\tilde{Q}_B u(Y - \Theta)|\Theta = \theta\} \tag{4.63}$$

Proceeding as before, we obtain the modified performance index

$$\hat{J}_B(\theta, \delta) = \lambda_1 p_1 - E_{Y,\tilde{Q}_B}\{\tilde{Q}_B \hat{u}(Y - \Theta, \delta)|\Theta = \theta\} \tag{4.64}$$

with derivative

$$\frac{\partial \hat{J}_B}{\partial \theta}(\theta, \delta) = E_{Y,\tilde{Q}_B}\{\tilde{Q}_B h(Y - \Theta, \delta)|\Theta = \theta\} \tag{4.65}$$

We may then use the steps

$$\begin{aligned} Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) &= \tilde{Q}_{B(k)} h(Y_k - \Theta_k, \delta_k) \\ &= \begin{cases} -\lambda_0 h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ +\lambda_1 h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{4.66}$$

leading to the algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k \lambda_0 h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ \theta_k - \rho_k \lambda_1 h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{4.67}$$

Similar arguments lead to normalized steps of the form

$$\begin{aligned} Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) &= 2\delta_k \tilde{Q}_{B(k)} h(Y_k - \Theta_k, \delta_k) \\ &= \begin{cases} -\lambda_0 2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ +\lambda_1 2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{4.68}$$

giving the algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k \lambda_0 2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ \theta_k - \rho_k \lambda_1 2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{4.69}$$

Thus, the costs λ_0 and λ_1 enter the algorithms as stepsize bias. In contrast to the conditional densities and prior probabilities, the cost information had to be modeled explicitly in the algorithm since it was not inherent in the data. Actually, the important quantity is the ratio of these costs, so to minimize the effect of the cost magnitudes on the gradient, the costs may be normalized, although this is not critical to the algorithm's validity in the sense of asymptotic convergence. If we define the quantity

$$L = \frac{\lambda_0}{\lambda_0 + \lambda_1} \tag{4.70}$$

then the unnormalized steps may be expressed using the normalized costs as

$$Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) = \begin{cases} -Lh(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ +(1-L)h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{4.71}$$

and for the normalized steps

$$Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) = \begin{cases} -L2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_0 \\ +(1-L)2\delta_k h(y_k - \theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{4.72}$$

For example, for the case of a rectangular window, the unnormalized steps take the form

$$Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) = \begin{cases} -L\frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +(1-L)\frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.73}$$

while the normalized steps take

$$Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) = \begin{cases} -L & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +(1-L) & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.74}$$

4.3.2 Numerical Experiments

In this section we experiment with the window algorithms to investigate how they perform in practice. The results of this section are intended to be only illustrative in nature, and represent neither an extensive parametric study of the algorithms, nor a detailed experimental comparison of the algorithms. We believe such study is better handled along with a rate of convergence analysis, and this was beyond the scope of this report.

Those readers who have had some experience with stochastic approximation algorithms know that, in spite of theoretical assurances of convergence from arbitrary starting values, and for a wide range of stepsize sequences, the algorithms can perform quite poorly if these parameters are not properly chosen. Unfortunately, optimum design of the algorithms requires knowledge of the functional form of the cost [72], and this information is typically unavailable in the nonparametric problem. Sklansky and Wassel [57] investigate some heuristic methods for choosing a good starting point, and good stepsize sequences. Their techniques rely on properties of the data set, such as observed sample means and variances. For example, it is not unreasonable to assume that the sample means of the two classes may be approximately determined, so that it is generally possible to initialize the algorithm between the means. The most important factor concerning initialization is that the algorithm begin in a region of significant probability density.

We will consider stepsize sequences of the general form

$$\rho_k = \frac{\rho_1}{k^a}, \quad \delta_k = \frac{\delta_1}{k^b} \tag{4.75}$$

where ρ_1 and δ_1 are static gain coefficients, and a, b are constant exponents. The static

gains determine the initial stepsize and window widths. It turns out that the proof of asymptotic convergence does not depend on the choice of static gains, although they certainly impact the convergence rate of the algorithms. In Chapter 6 we will specify allowable ranges of the exponents a, b which are sufficient for convergence. The allowed choice of one of these parameters is a function of the other. At present we will simply indicate the choice of sequence used.

Unnormalized vs. Normalized Variant

In this section we briefly compare the unnormalized and normalized variants of the window algorithm, using a rectangular window function. We consider the Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3, \quad \sigma^2 = 1, \quad p_0 = 0.5 \tag{4.76}$$

The optimal observation threshold for this problem is $\theta^* = 2.0$ with probability of error $P_\epsilon(\theta^*) = 0.1586$.

We first consider the unnormalized variant. We employ stepsize and window-width sequences of the form

$$\rho_1 = 1, \quad a = 1 \tag{4.77}$$

$$\delta_1 = 2.25, \quad b = 1/3 \tag{4.78}$$

where the static gain coefficients have been chosen as they were in [73] for a similar problem.

Figure 4-8 illustrates several typical sample paths of the algorithm. Most of the major variation in the sample paths has died out by 200 iterations. Two of the paths still exhibit error after 1000 iterations, but are continuing to move in the correct direction.

Figure 4-9 illustrates an average sample path, where the average is taken over 15 Monte-Carlo runs, i.e., over 15 independent sample paths. Each plotted threshold value is therefore an average over 15 values. Averaging results in significant improvement; now convergence appears to be achieved within the first 50 trials. The probability of error for the average sample path is depicted in Figure 4-10.

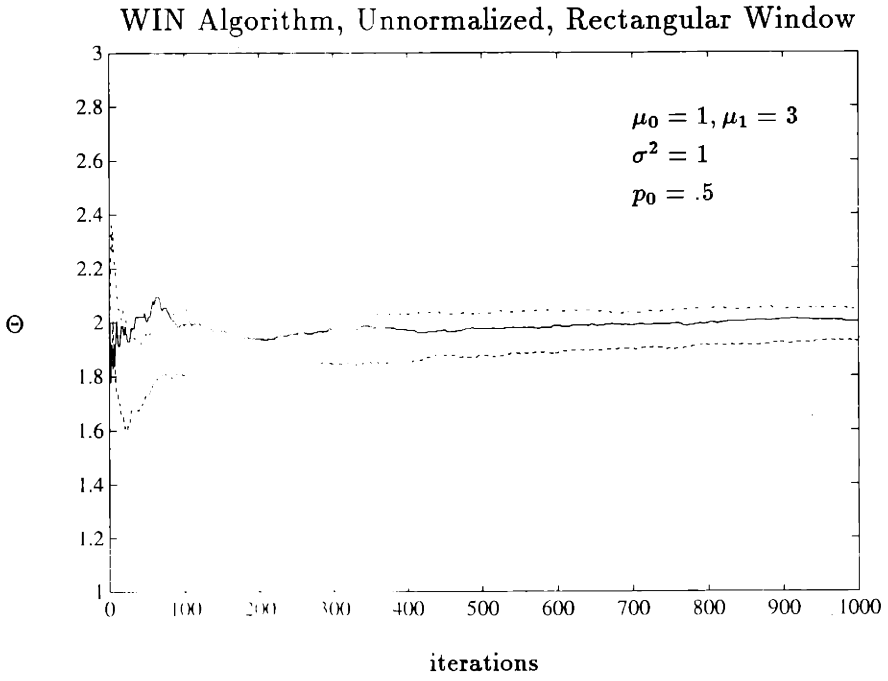


Figure 4-8: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/k$, $\delta_k = 2.25/(k)^{1/3}$; Gaussian case.

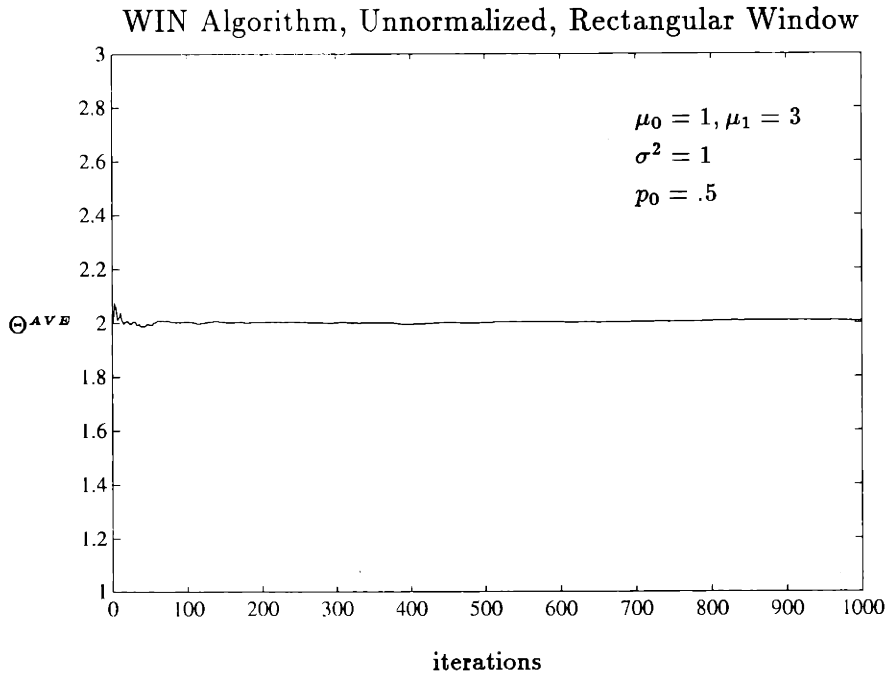


Figure 4-9: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. The optimal value is $\theta^* = 2.0$ (dashed).

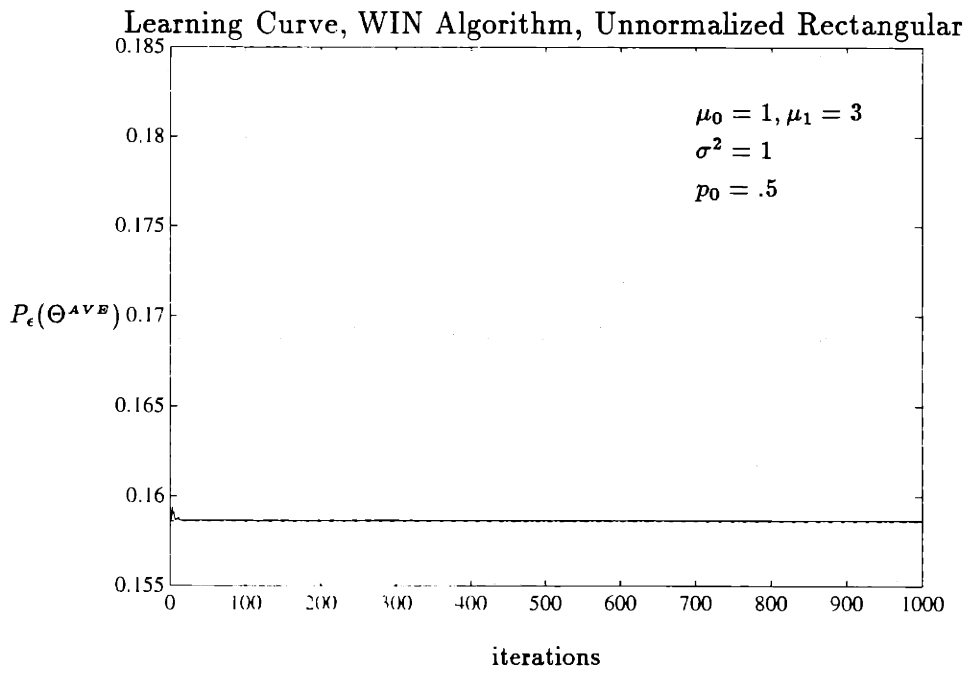


Figure 4-10: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). Optimal value $P_\epsilon(\Theta^*) = 0.1586$ (dotted and dashed).

For the normalized variant, we will use stepsize and window-width sequences of the form

$$\rho_1 = 1, \quad a = 1/2 \tag{4.79}$$

$$\delta_1 = 2.25, \quad b = 1/2 \tag{4.80}$$

Some typical sample paths are shown in Figure 4-11. In comparison with the sample paths of the unnormalized variant in Figure 4-8, these paths are more staircase in nature, and possess larger variance, although our analysis in Chapter 6 suggests that asymptotically the normalized variant may possess a superior convergence rate. The transient variance of the unnormalized variant is definitely larger due to its more slowly decreasing stepsize. An average sample path is depicted in Figure 4-12, where again a dramatic improvement in performance is observed. Here approximate convergence is evident within 200 iterations. The corresponding probability of error is depicted in Figure 4-13. There is a clear initial blip in the error probability corresponding to the large transient variance.

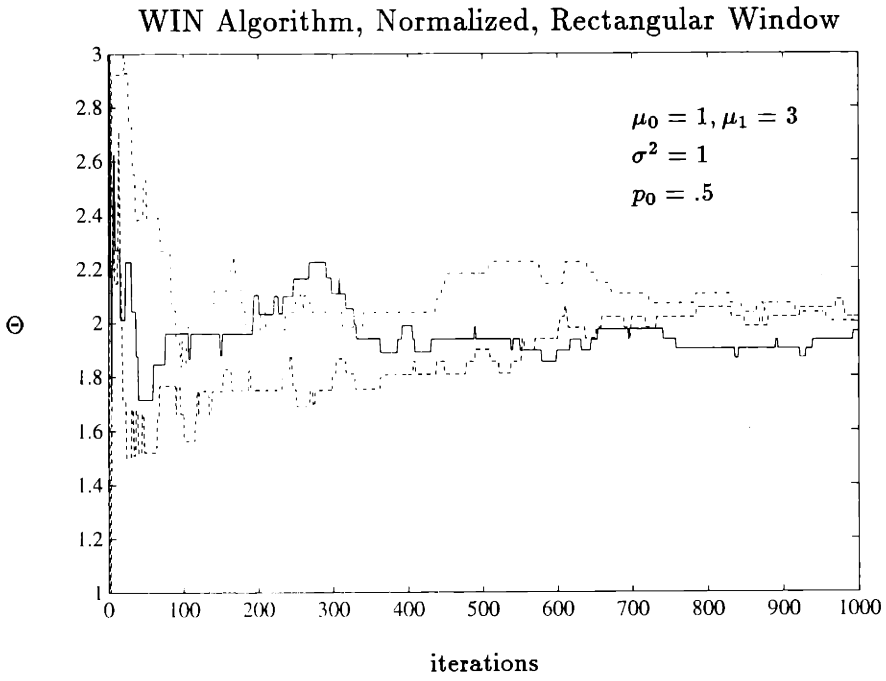


Figure 4-11: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

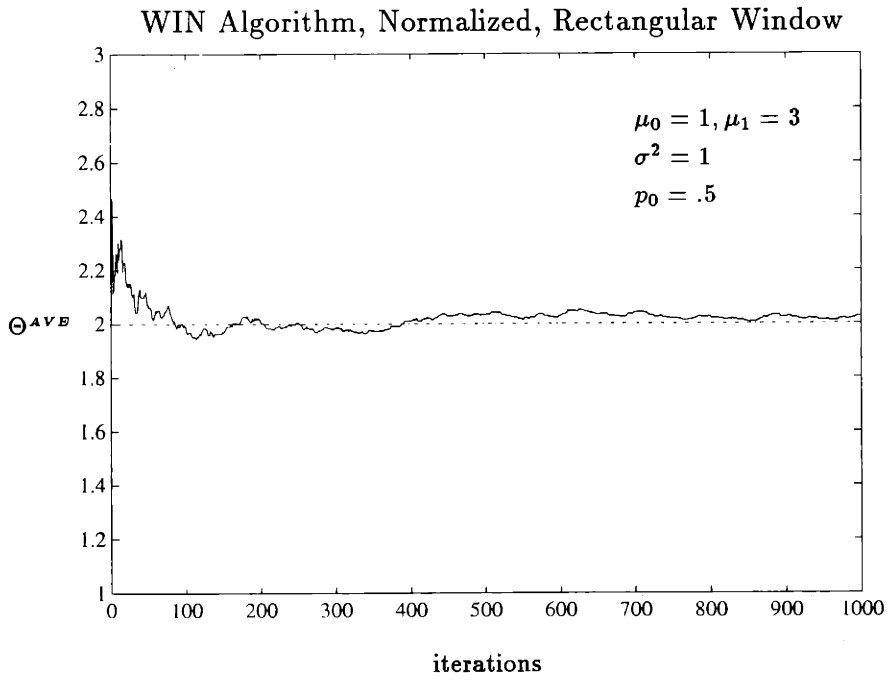


Figure 4-12: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2 (dashed and dotted); Gaussian case.

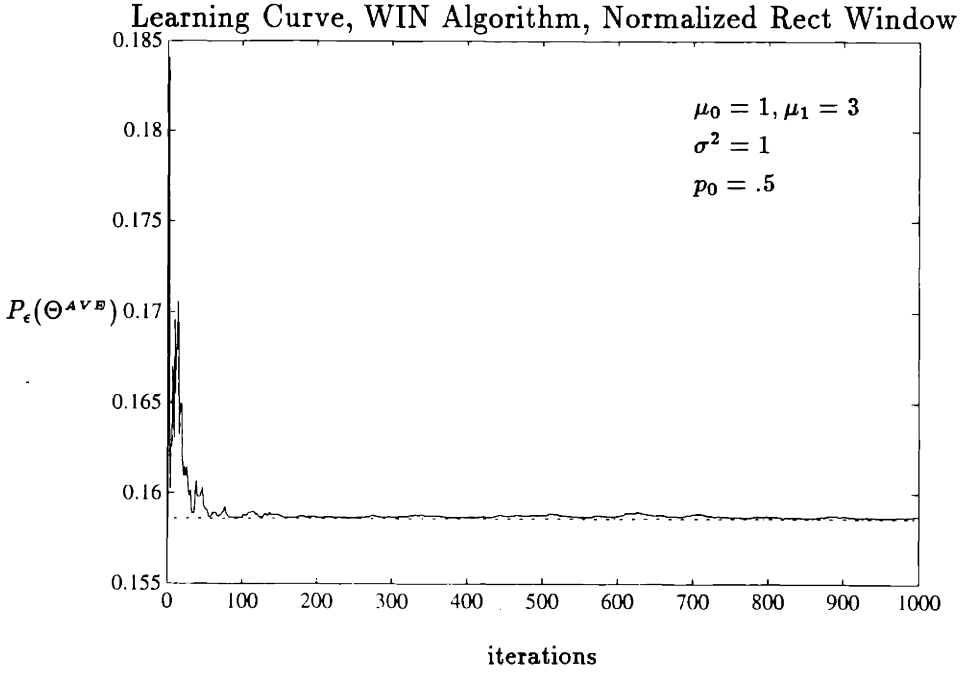


Figure 4-13: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.1586 (dashed and dotted).

In summary, the paths generated by the unnormalized and normalized variants are distinct, with the paths of the unnormalized variant being smoother, and less responsive than those of the normalized variant. Henceforth, we will exclusively use the normalized variant when presenting numerical results for the WIN algorithms. This is due to the close relationship between the unnormalized variant and the two-sided Kiefer-Wolfowitz algorithm which we will be considering shortly.

Window Functions

In this section we compare the sample paths of some different window functions on the problem

$$\mu_0 = 1, \quad \mu_1 = 3 \tag{4.81}$$

$$\sigma^2 = 1, \quad p_0 = 0.75 \tag{4.82}$$

where now we have incorporated a priori bias. The optimal observation threshold for this problem is $\theta^* = 2.55$ with minimum probability of error $P_\epsilon(\theta^*) = 0.127$. We

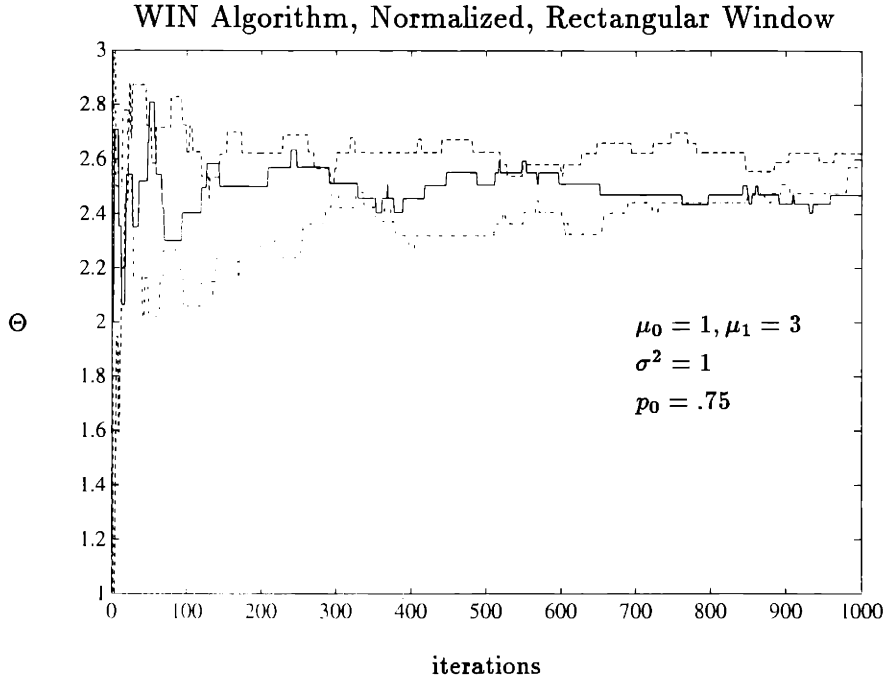


Figure 4-14: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$.

have two purposes in this section; we wish to determine whether the WIN algorithm is effective at determining the proper biasing of the threshold dictated by the prior probabilities, and we wish to examine the effect of using triangular and Gaussian windows, to see if there is enough benefit to justify their additional overhead.

In Figure 4-14 several sample paths of the normalized window algorithm, using a rectangular window function, are illustrated. Notice that the algorithm has clearly responded to the prior bias in the data. Figure 4-15 illustrates an averaged sample path, while Figure 4-16 shows the corresponding probability of error. Again there is a transient blip corresponding to the large initial variance.

Figure 4-17 illustrates several sample paths of the normalized window algorithm using a triangular window. These paths are hardly distinguishable from those of the rectangular window. Figures 4-18 and 4-19 illustrate the averaged path and corresponding probability of error, respectively. A slightly smaller transient error is evident, but otherwise the performance is very similar to that of the rectangular window.

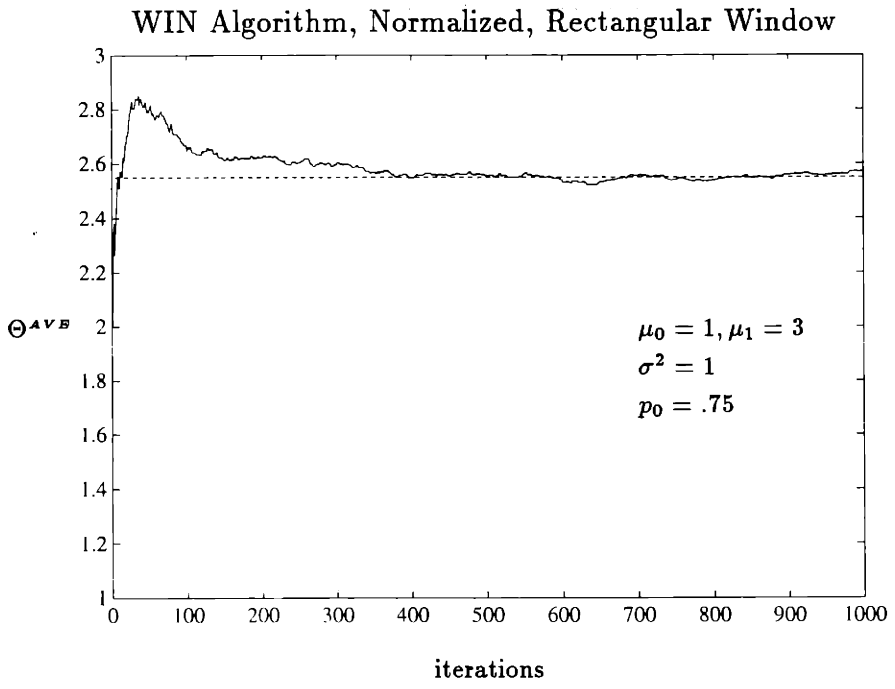


Figure 4-15: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2.55 (dashed)

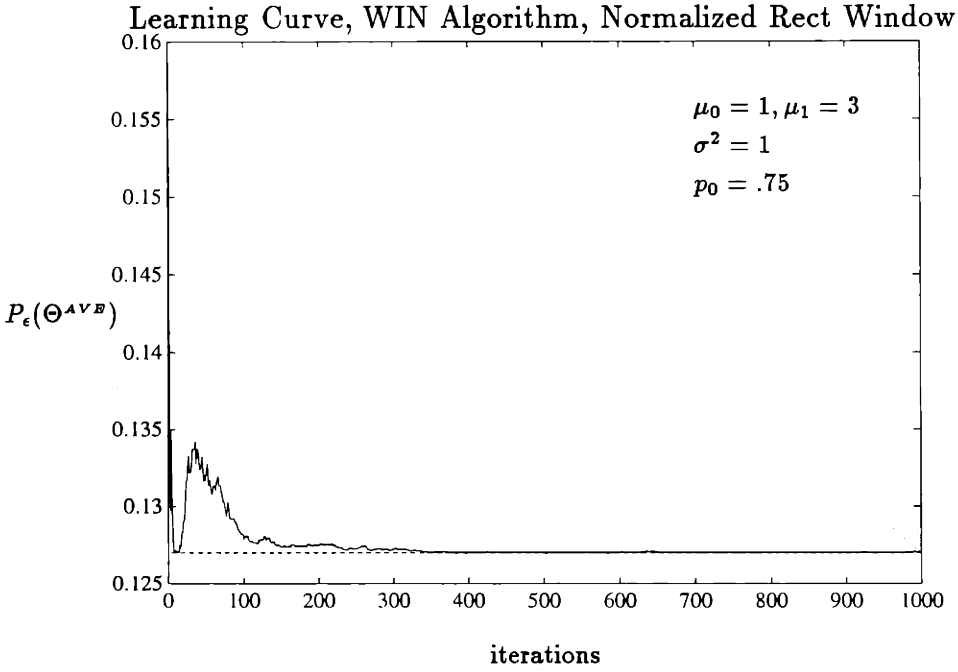


Figure 4-16: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.127 (dashed).

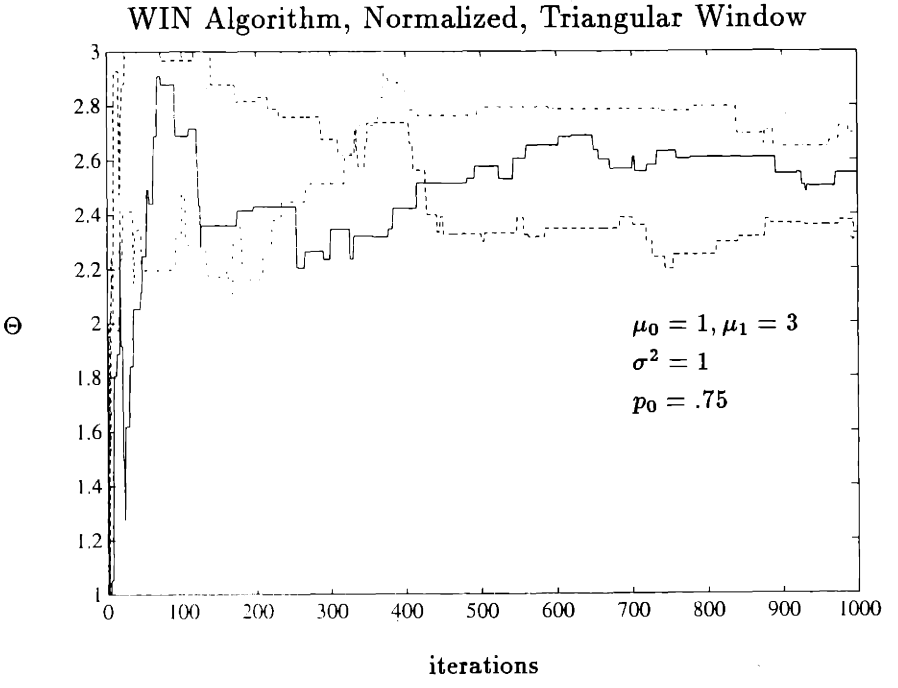


Figure 4-17: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

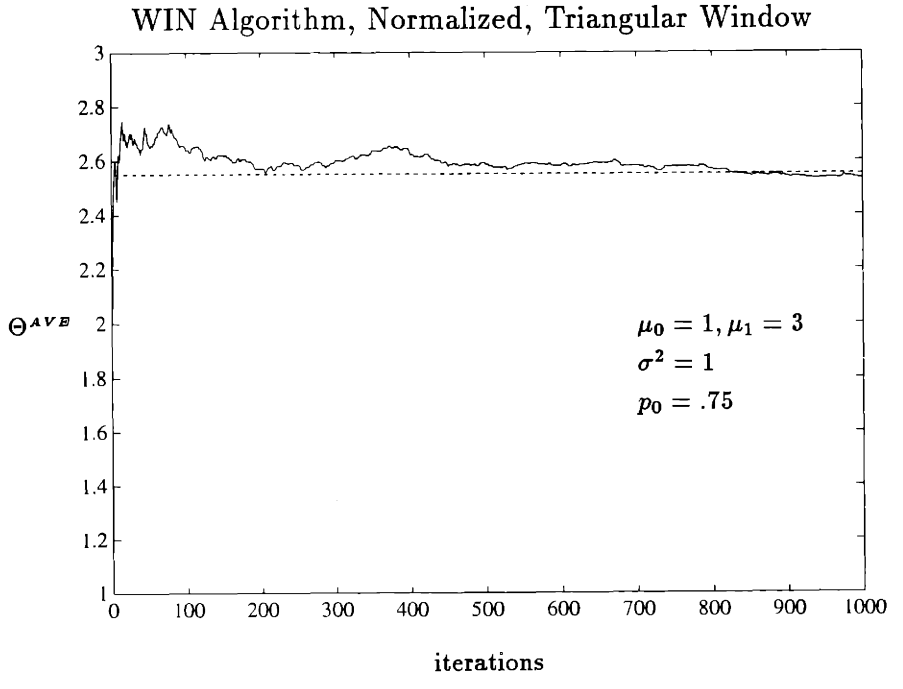


Figure 4-18: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2.55 (dashed); Gaussian case.

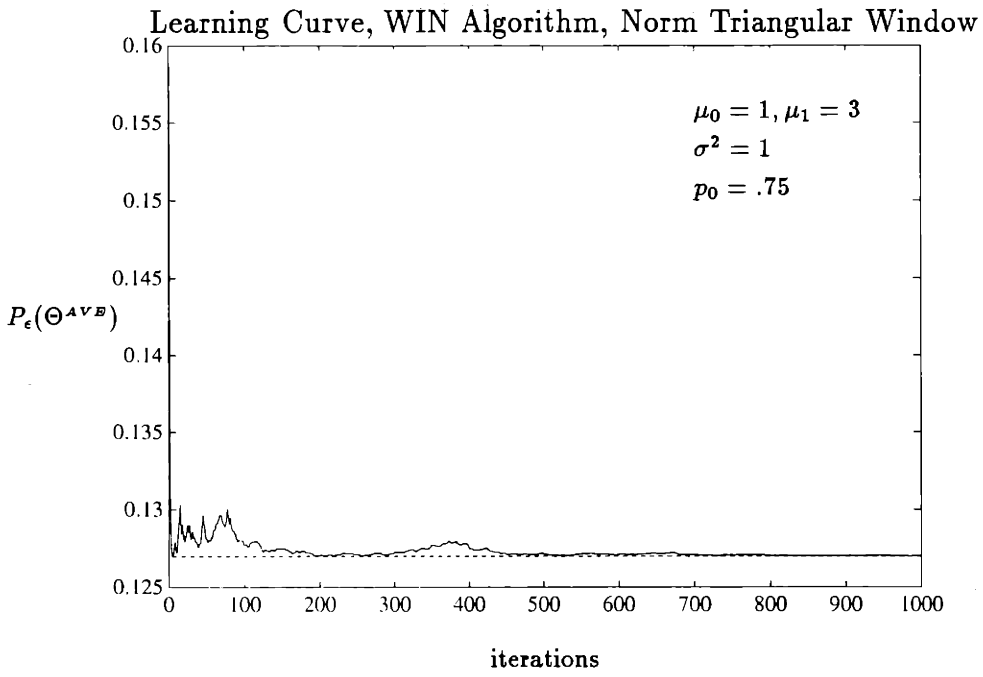


Figure 4-19: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.127 (dashed).

Several sample paths corresponding to the Gaussian window function are illustrated in Figure 4-20. In comparison with the previous windows, these paths appear slightly smoother, although again not enough to justify the additional computational overhead. The corresponding average path and probability of error appear in Figures 4-21 and 4-22, respectively.

More extensive numerical experience with these algorithms indicates that on the average, the performance of these windows is in fact in the order of their computational complexity, namely from best to worst Gaussian, triangular, rectangular, although the differences are very slight on the Gaussian problems we consider. Thus, we henceforth use the normalized rectangular window.

Variances

In this section we examine the impact of different variances on the performance of the normalized WIN algorithm, using a rectangular window. We consider the problem

$$\mu_0 = 1, \quad \mu_1 = 3, \quad p_0 = 0.5 \tag{4.83}$$

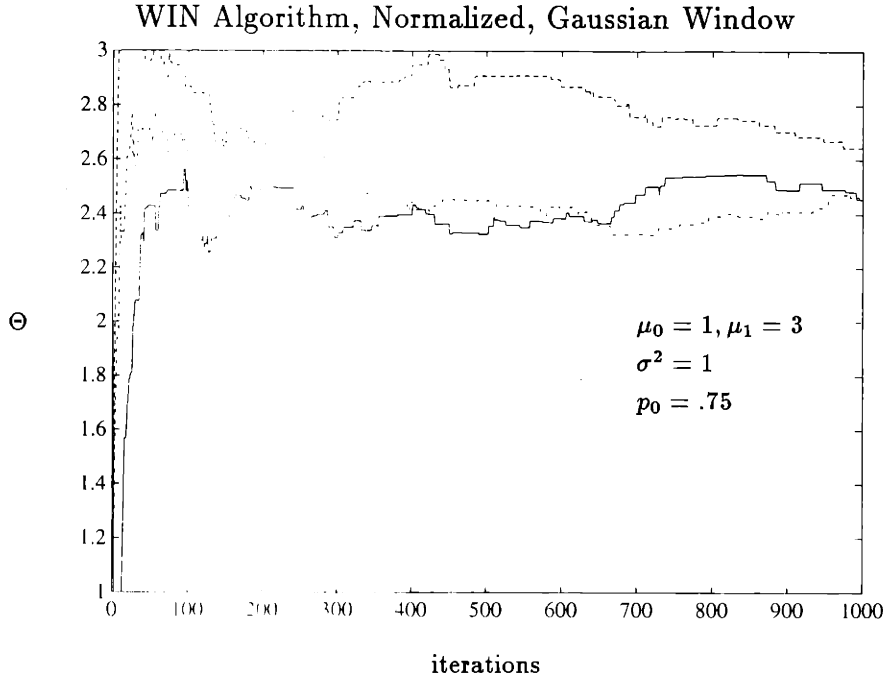


Figure 4-20: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

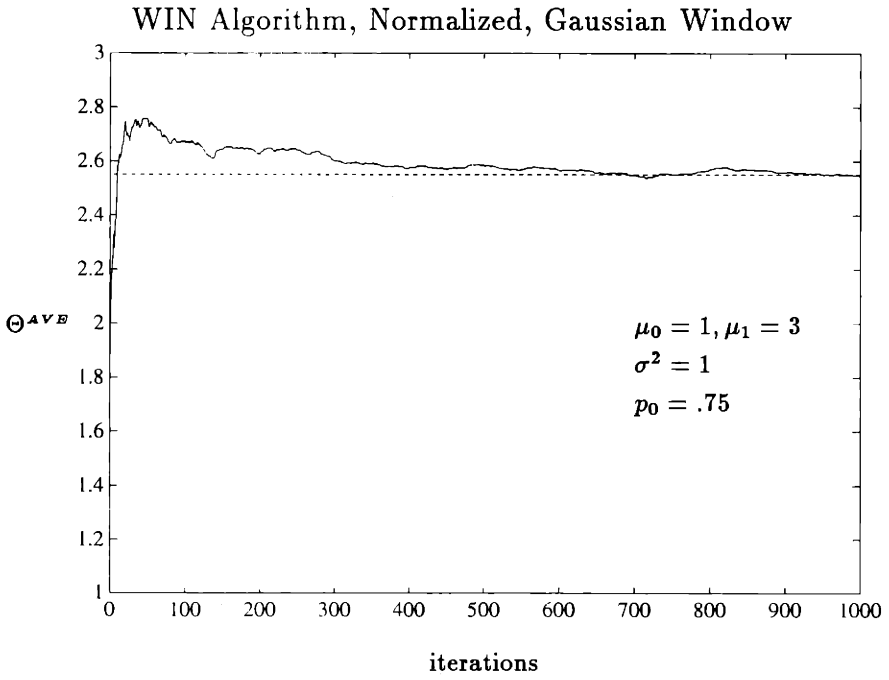


Figure 4-21: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2.55 (dashed); Gaussian case.

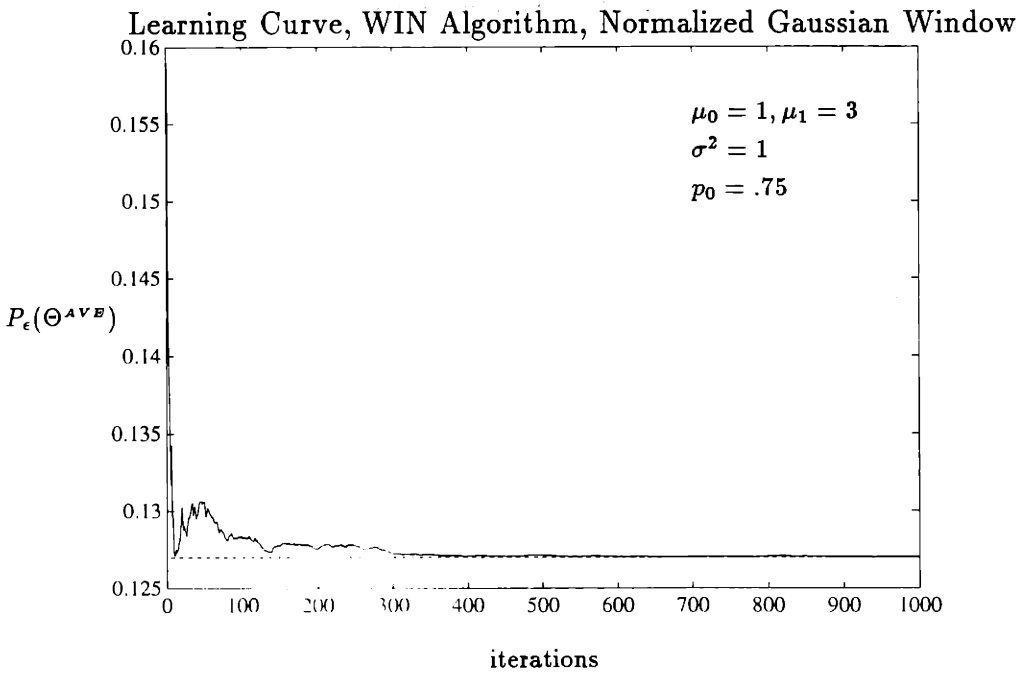


Figure 4-22: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.127 (dashed).

for the cases

$$\sigma^2 = 0.5, \text{ and } \sigma^2 = 2 \tag{4.84}$$

The case $\sigma^2 = 1$ has already been considered in Figures 4-11 - 4-13. The optimal observation threshold for the case $\sigma^2 = 0.5$ is $\theta^* = 2.0$ with minimum probability of error $P_\epsilon(\theta^*) = 0.0786$, while for the case $\sigma^2 = 2.0$ the optimal threshold is again $\theta^* = 2.0$ with $P_\epsilon(\theta^*) = 0.2398$.

In Figure 4-23 several sample paths for the lower variance case are depicted. In comparison to Figure 4-11, the sample paths are more tightly clustered around $\theta = 2$. However, the average sample path in Figure 4-24 and corresponding probability of error in Figure 4-25 indicate that averaging minimizes these differences.

In contrast, the higher variance case results in sample paths of the form shown in Figure 4-26. There is clearly much more variability in the individual sample paths, and several of the paths do not appear to be converging to $\theta = 2.0$, even after 1000 iterations. The reason for this is that sides of the cost bowl are shallower, so that the sample paths are not as confined as in the other cases. However, on the average, the

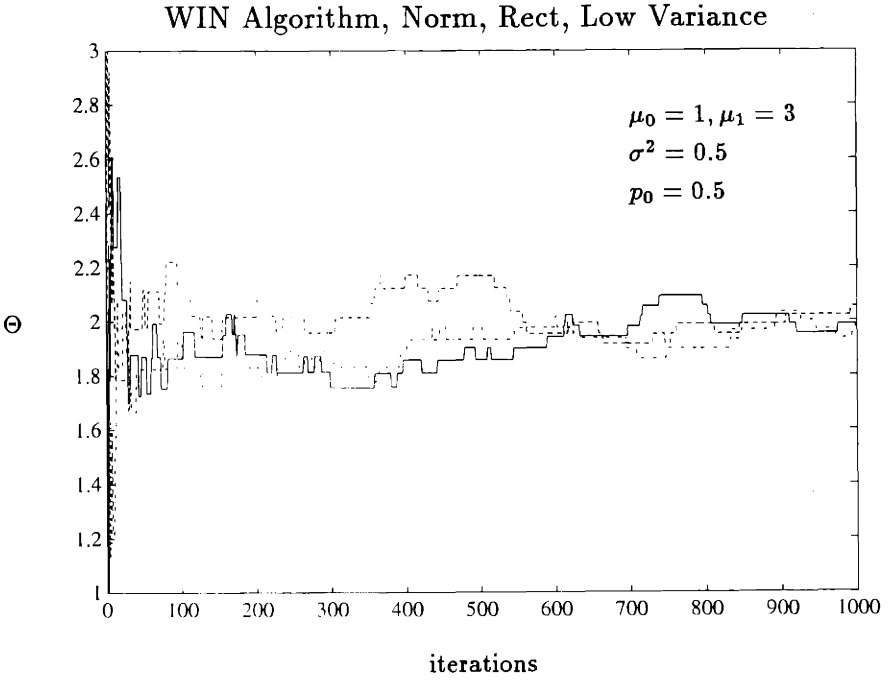


Figure 4-23: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

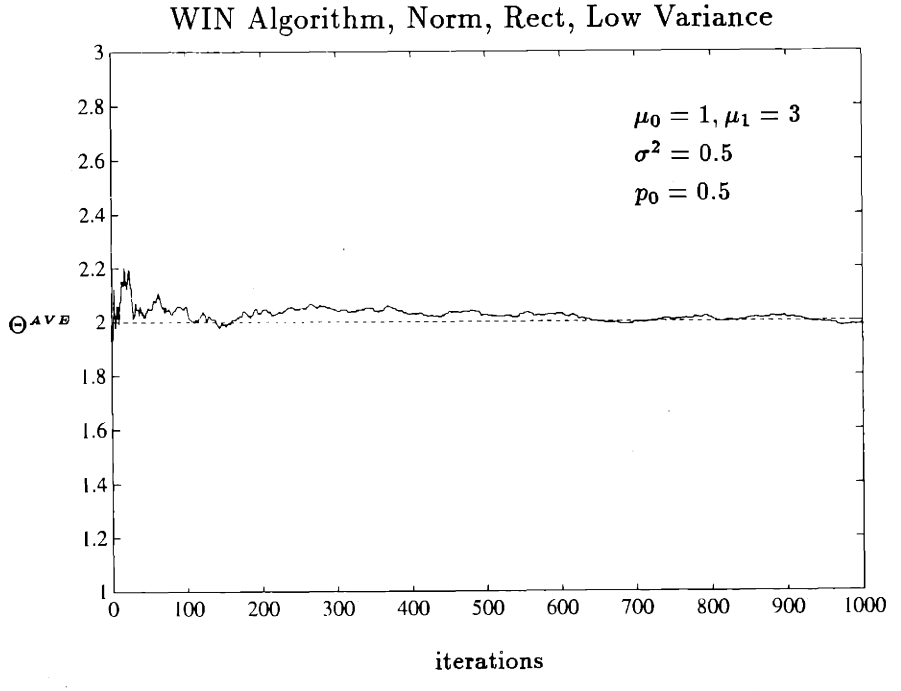


Figure 4-24: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2.0 (dashed); Gaussian case.

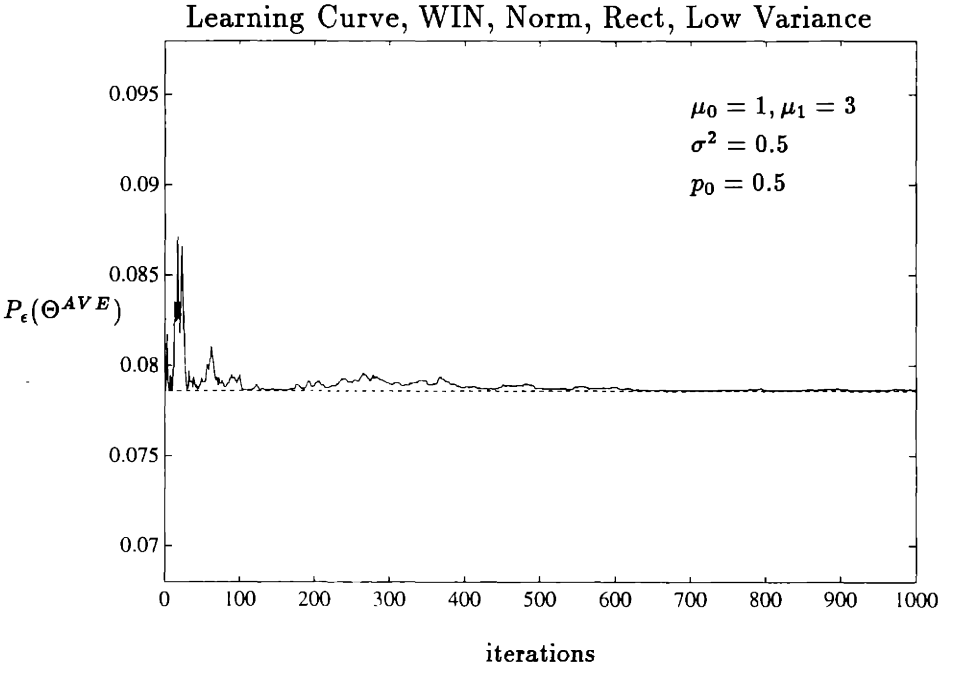


Figure 4-25: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.0786 (dashed).

performance of the algorithm is similar to the previous cases, as evidenced by Figures 4-27 and 4-28.

Thus, it appears that higher variance in the underlying hypothesis test results in sample paths which may wander significantly, although sample path averaging successfully mitigates these effects.

Costs

In this final section, we examine the effect of unequal costs on the performance of the algorithms. Our central purposes are to determine whether addition of the stepsize bias can in fact compensate for unequal costs on the errors, and to examine the ill conditioning which may result if widely different costs are chosen. We consider the problem

$$\mu_0 = 1, \quad \mu_1 = 3, \quad p_0 = 0.5, \quad \sigma^2 = 1 \tag{4.85}$$

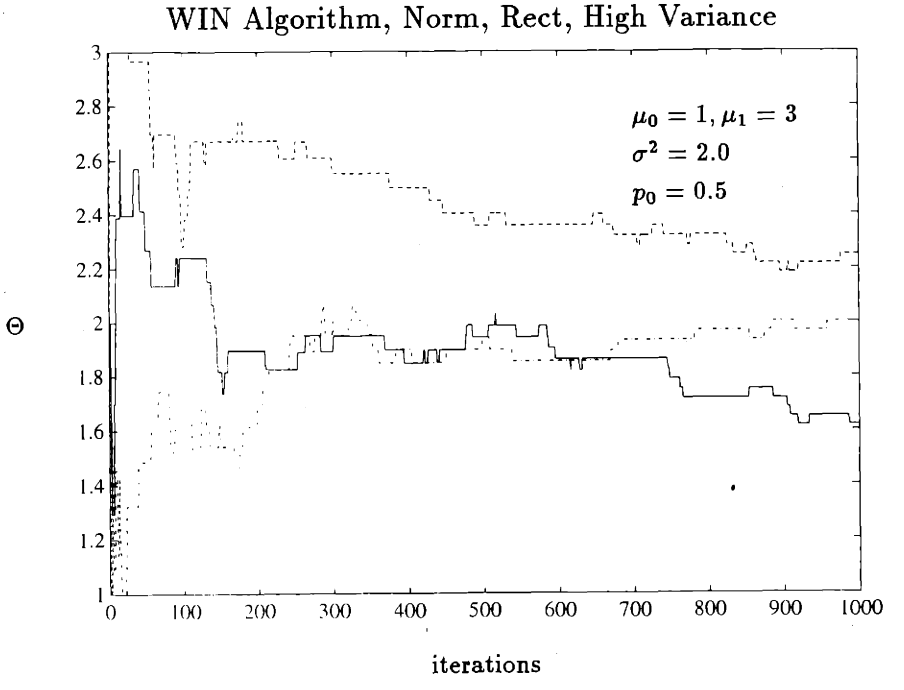


Figure 4-26: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

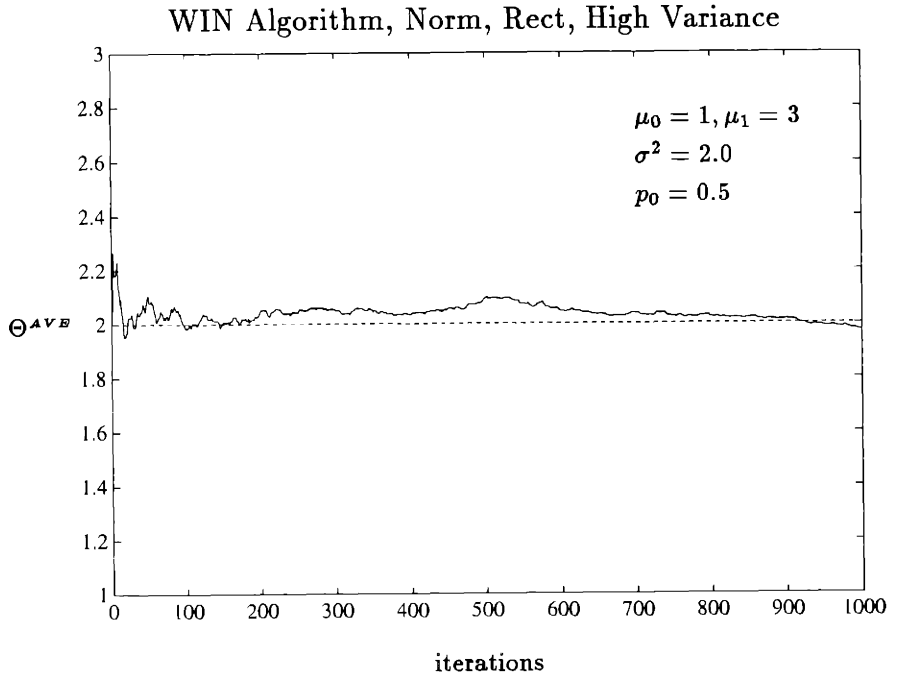


Figure 4-27: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2.0 (dashed); Gaussian case.

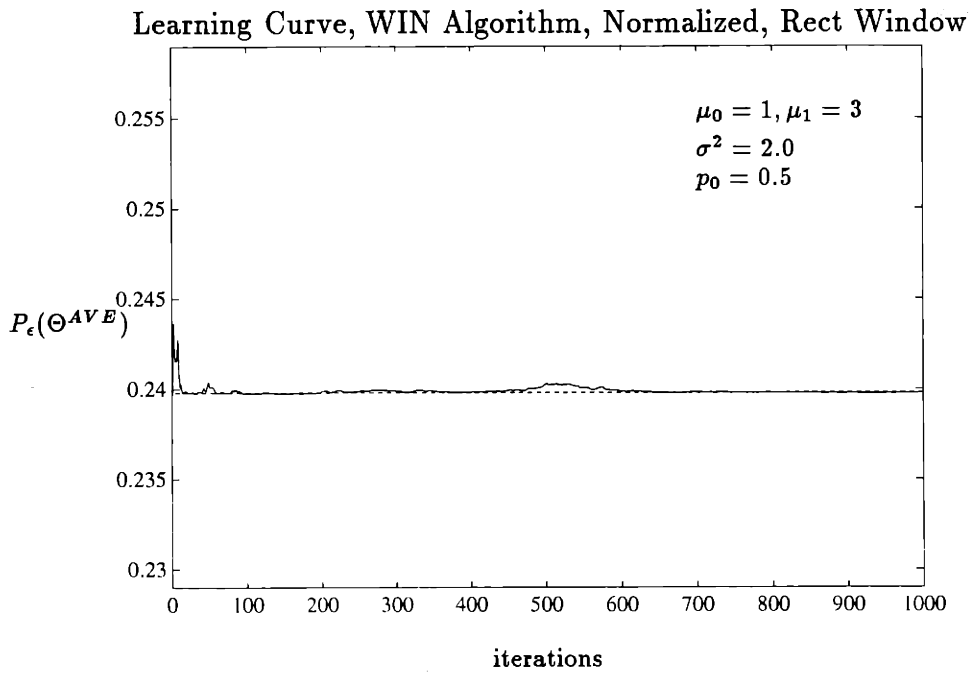


Figure 4-28: Sample Path of $\{P_\epsilon(\theta_k^{AVE})\}$ (solid). The optimal value is 0.2398 (dashed).

for the cases $\lambda_0 = 4.0, \lambda_1 = 0.25$ and $\lambda_0 = 2.0, \lambda_1 = 0.5$. The optimal observation threshold for the first case is $\theta^* = 3.38$ with minimum Bayes risk $J_B(\theta^*) = 0.0938$, while for the second case the optimal threshold is $\theta^* = 2.69$ with risk $J_B(\theta^*) = 0.1401$.

Several sample paths for the case $\lambda_0 = 4.0, \lambda_1 = 0.25$, using a normalized rectangular window and normalized costs, are shown in Figure 4-29. It was predicted in Chapter 3 that wide disparities in the magnitude of the costs can cause difficulty for convergence of the parameter sequence since a large flat region of approximately equal cost is created. For this particular case, the costs differ by a factor of 16. The problem is evident in Figure 4-30 where it is seen that convergence of the parameter sequence to the optimal value is not even observed on the averaged sample path. However, the cost is very near its optimal value as indicated in Figure 4-31.

For the second case, $\lambda_0 = 2, \lambda_1 = 0.5$, the problem is better posed as the costs differ only by a factor of 4. It is not surprising that the performance of the algorithm is significantly better on this case.

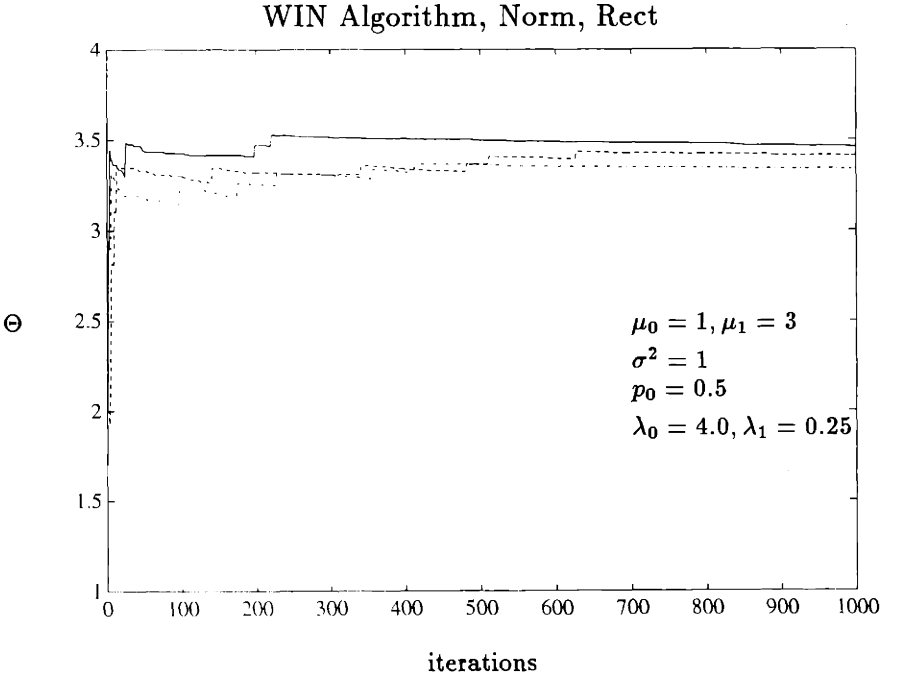


Figure 4-29: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

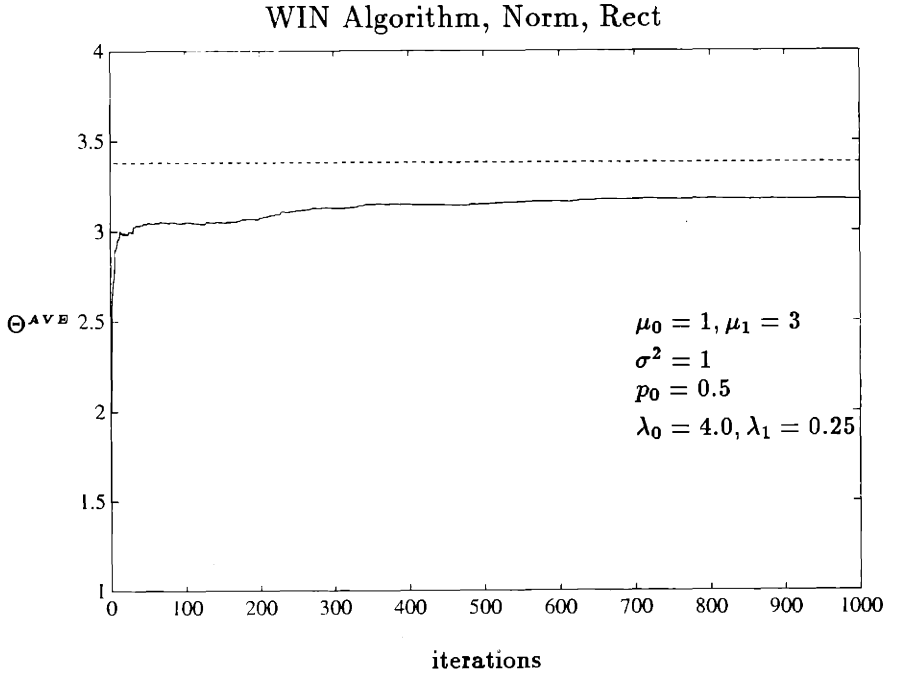


Figure 4-30: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 3.38 (dashed); Gaussian case.

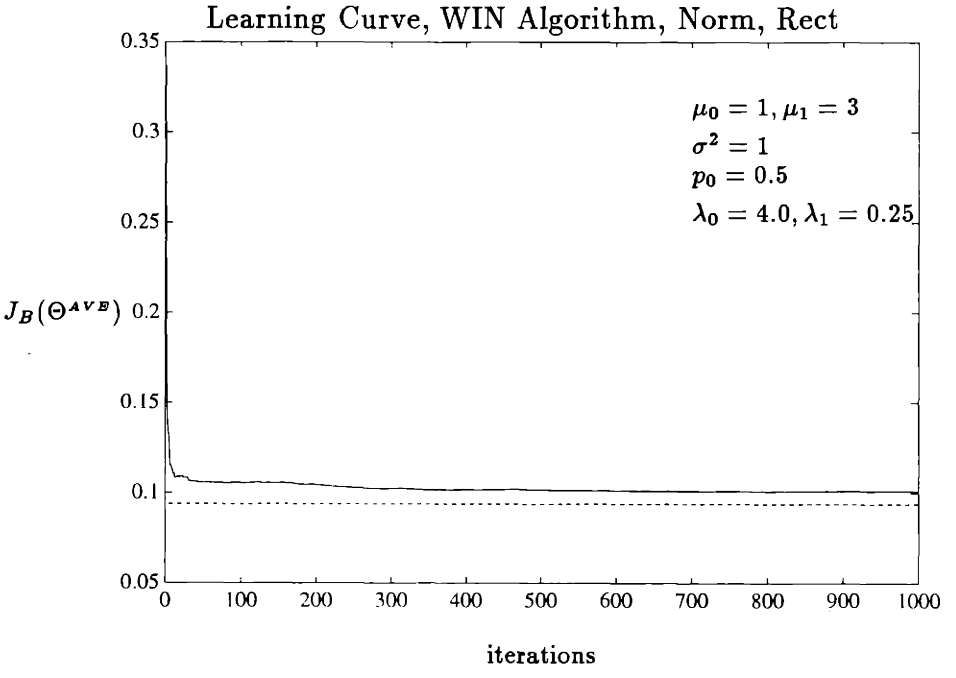


Figure 4-31: Sample Path of $\{J_B(\theta_k^{AVE})\}$ (solid). The optimal value is 0.0938 (dashed).

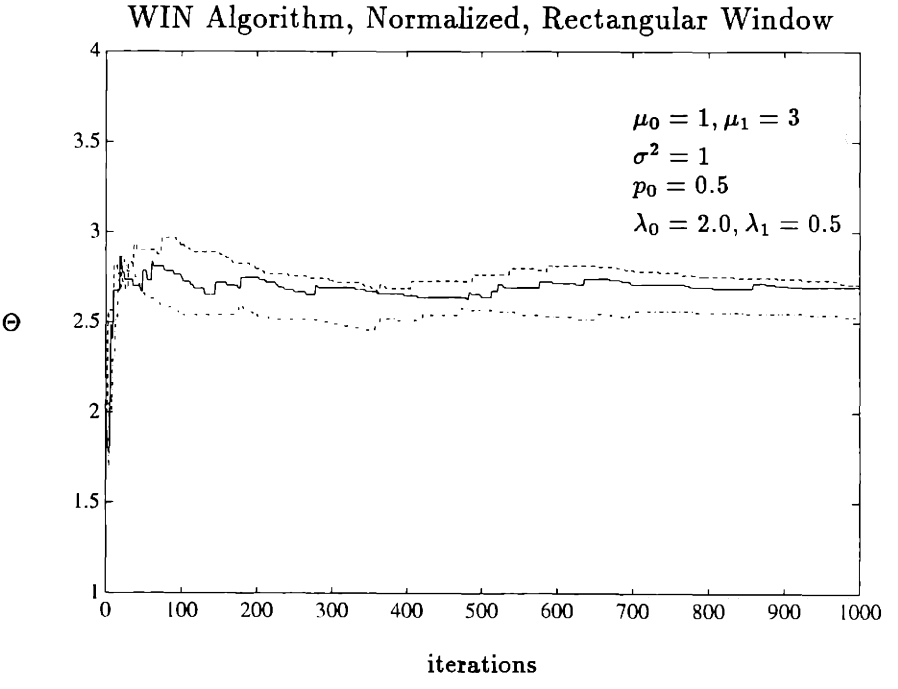


Figure 4-32: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$; Gaussian case.

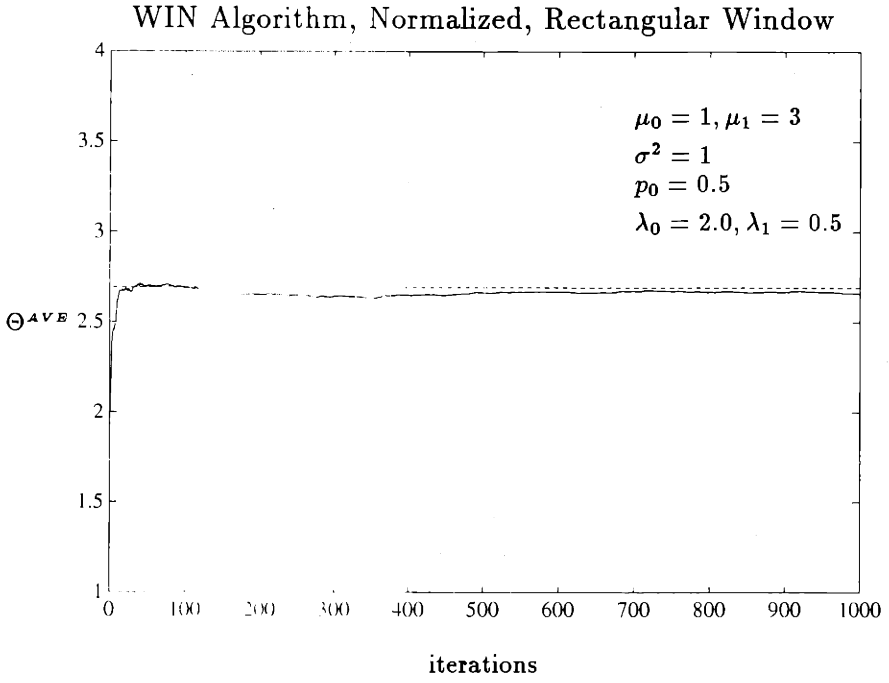


Figure 4-33: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/\sqrt{k}$, $\delta_k = 2.25/\sqrt{k}$. The optimal value of the threshold is 2.69 (dashed); Gaussian case.

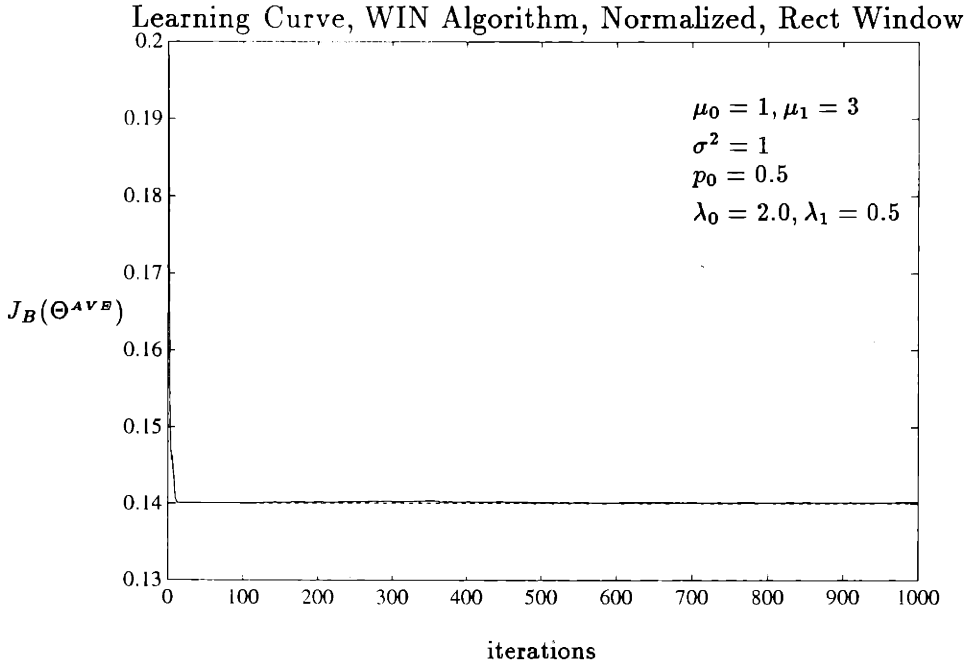


Figure 4-34: Sample Path of $\{J_B(\theta_k^{AVE})\}$ (solid). The optimal value is 0.1401 (dashed).

4.4 Kiefer-Wolfowitz (KW) Training Algorithm

Formulation of the training problem in the Kiefer-Wolfowitz setting requires only a stochastic realization, or unbiased estimator, of the $P_\epsilon(\theta)$ function itself. As discussed in the previous section, it suffices to take the indicator function for an error

$$Q(X,\Theta) = \begin{cases} 1 & \text{if } y > \theta \text{ and } H^k = H_0 \\ 1 & \text{if } y < \theta \text{ and } H^k = H_1 \\ 0 & \text{else} \end{cases} \tag{4.86}$$

for which it is clear that

$$E_X\{Q(X,\Theta)|\Theta = \theta\} = P_\epsilon(\theta),\ \forall\theta \in \Re \tag{4.87}$$

Note that the "sampling error" in this estimate cannot be characterized as either additive or relative. It results from the fact that the relative frequency estimate of a fixed, but unknown, probability is binomially distributed with mean equal to the true value (see Appendix A). In this case, we use a relative frequency estimate with a single sample. So it holds that

$$Q(X,\Theta) = \begin{cases} 1 & \text{w.p. } P_\epsilon(\Theta) \\ 0 & \text{w.p. } (1 - P_\epsilon(\Theta)) \end{cases} \tag{4.88}$$

We may then approximate the derivative $dP_\epsilon(\theta_k)/d\theta$ by using the realizations of Q_k in a finite-difference approximation. For example, a forward-difference (one-sided) approximation of the derivative at time k is given by

$$Z_k(X_k,\Theta_k,\delta_k) = [Q_k(X_k,\Theta_k+\delta_k) - Q_k(X_k,\Theta_k)]/\delta_k \tag{4.89}$$

where $\{\delta_k\}$ is a decreasing real-valued nonnegative perturbation sequence. Note that the probability of error is sampled at the value of the current iterate θ_k and the

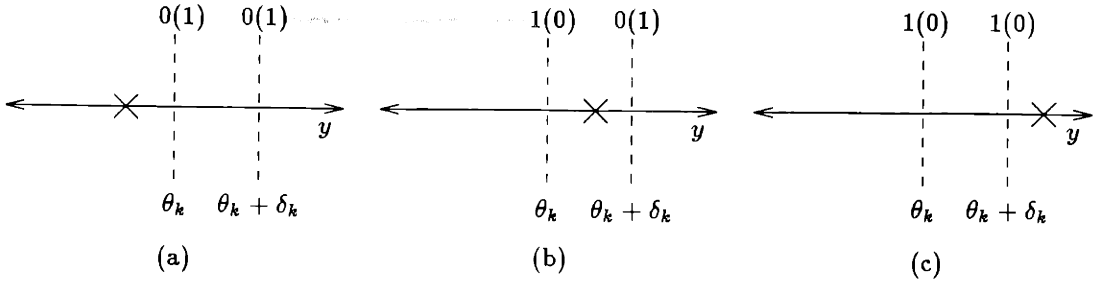


Figure 4-35: Possible positions of a realization of Y, denoted with an $\times$, with respect to the current sampling locations θ_k and $\theta_k + \delta_k$. Values of the sample function Q corresponding to a given threshold setting, and given that $H^k = H_0$, are shown above each threshold, while values given that $H^k = H_1$ are shown beside in parenthesis.

perturbed value $\theta_k + \delta_k$. A central-difference (two-sided) approximation is given by

$$Z_k(X_k, \Theta_k, \delta_k) = [Q_k(X_k, \Theta_k + \delta_k) - Q_k(X_k, \theta_k - \delta_k)]/2\delta_k \tag{4.90}$$

The finite-difference estimate may be constructed using two samples based on the same measurement X_k, or a different measurement may be used to generate each sample [56]. There is more discussion of this in Chapter 6. For notational simplicity, we choose to use the same measurement throughout this report.

The derivative estimate Z_k generated at time k by the finite difference techniques just described takes the following form. For the one-sided approximation, there are three possible positions for a realization of the observation y relative to the sample locations θ_k and $\theta_k + \delta_k$ as shown in Figure 4-35.

For case (a), the realization falls below both θ_k and the forward perturbed value $\theta_k + \delta_k$, resulting in $Q_k = 0$ in both cases if $H^k = H_0$, and $Q_k = 1$ in both cases if $H^k = H_1$. In either case, the corresponding value of Z_k would be zero, and the parameter θ_k would not be updated. For case (b), the realization of the observation lands between the current value of the iterate and its forward perturbed value, meaning that the decision $u_k = 1$ is made with the threshold setting θ_k, and $u_k = 0$ is made with setting $\theta_k + \delta_k$. The difference results in a different value of Q_k corresponding to each, so that Z_k is now nonzero. The situation for case (c) mimics case (a), and results in

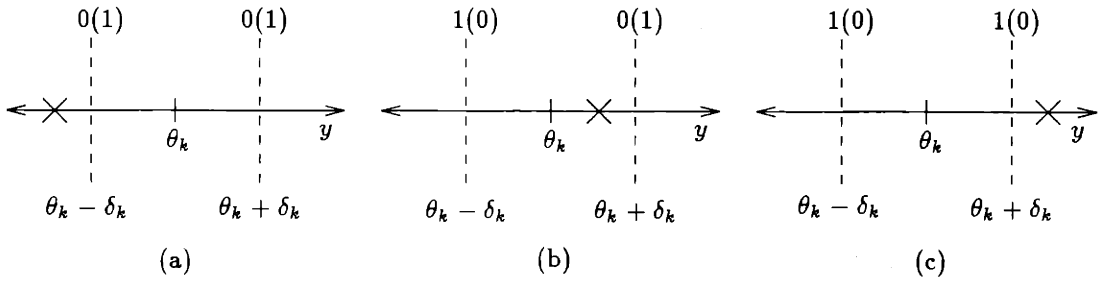


Figure 4-36: Possible positions of a realization of Y, denoted with an $\times$, with respect to the current sampling locations $\theta_k + \delta_k$ and $\theta_k - \delta_k$. Values of the sample function Q corresponding to a given threshold setting, and given that $H^k = H_0$, are shown above each threshold, while values given that $H^k = H_1$ are shown beside in parenthesis.

$Z_k = 0$ as well. In summary, we can express Z_k for the one-sided technique as

$$Z_k(X_k, \Theta_k, \delta_k) = \begin{cases} -\frac{1}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +\frac{1}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.91}$$

where we see the same windowing type-behavior which characterized the WIN algorithm using a rectangular window function.

Similarly, for the two-sided variant we obtain the situation depicted in Figure 4-36 which results in the step

$$Z_k(X_k, \Theta_k, \delta_k) = \begin{cases} -\frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +\frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.92}$$

It is interesting that the form of this algorithm is identical to that derived using the WIN algorithm with unnormalized steps and a rectangular window. Thus, the KW algorithm has appeared as a special case of the window algorithm. For the particular case of smoothing with a rectangular window function, we see that exactly computing the gradient of the smoothed function is equivalent to estimating the gradient of the exact function. Interestingly, this indicates that the KW technique, which is not strictly a gradient method, can be treated as such on a related smoothed function.

Finally, we note that the Z_k's derived above result in the following very similar algorithms. For the one-sided finite difference approximation

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k \frac{1}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ \theta_k & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - \rho_k \frac{1}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.93}$$

and for the two-sided approximation

$$\theta_{k+1} = \begin{cases} \theta_k + \rho_k \frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ \theta_k & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - \rho_k \frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.94}$$

4.4.1 Unequal Cost Version

The above technique is also easily extended to the unequal cost (general Bayes) problem. Choosing

$$Q(X, \lambda_0, \lambda_1, \Theta) = \begin{cases} \lambda_0 & \text{if } y > \theta \text{ and } H^k = H_0 \\ \lambda_1 & \text{if } y < \theta \text{ and } H^k = H_1 \\ 0 & \text{else} \end{cases} \tag{4.95}$$

steps for the one-sided variant become

$$Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) = \begin{cases} -\lambda_0 \frac{1}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +\lambda_1 \frac{1}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.96}$$

and for the two-sided variant

$$Z_k(X_k, \lambda_0, \lambda_1, \Theta_k, \delta_k) = \begin{cases} -\lambda_0 \frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ +\lambda_1 \frac{1}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.97}$$

These steps give rise to the algorithms

$$\theta_{k+1} = \begin{cases} \theta_k + \lambda_0 \frac{\rho_k}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - \lambda_1 \frac{\rho_k}{\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.98}$$

for the one-sided approximation and

$$\theta_{k+1} = \begin{cases} \theta_k + \lambda_0 \frac{\rho_k}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_0 \\ 0 & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - \lambda_1 \frac{\rho_k}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{4.99}$$

for the two-sided.

As before, we may also normalize the costs yielding for the two-sided algorithm

$$\theta_{k+1} = \begin{cases} \theta_k + L \frac{\rho_k}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ y_k \text{ from } H_0 \\ \theta_k & \text{if } |\theta_k - y_k| > \delta_k \\ \theta_k - (1 - L) \frac{\rho_k}{2\delta_k} & \text{if } |\theta_k - y_k| \leq \delta_k,\ y_k \text{ from } H_1 \end{cases} \tag{4.100}$$

where

$$L = \frac{\lambda_0}{\lambda_0 + \lambda_1} \tag{4.101}$$

Again one arrives at the unnormalized WIN algorithm using a rectangular window for the unequal cost problem.

4.4.2 Numerical Experiments

In this section we perform a few numerical experiments primarily aimed at comparing the sample paths of the KW algorithm to those of the WIN algorithm, and comparing the performance of the one-sided and two-sided KW methods. Sensitivity of the KW algorithms to prior probabilities, changing variances, and costs is analogous to the window algorithms, and is therefore not repeated here.

We consider the Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3, \quad \sigma^2 = 1, \quad p_0 = 0.75 \tag{4.102}$$

This is the same test problem used in Section 4.3.2 to investigate whether the WIN algorithms could successfully infer a priori bias in the data. The optimal observation threshold for this problem is $\theta^* = 2.55$ with minimum probability of error $P_\epsilon(\theta^*) = 0.1270$.

We first consider the one-sided technique, for which we use the stepsize and perturbation sequences

$$a = 1, \quad \rho_1 = 1 \tag{4.103}$$

$$b = 1/4, \quad \delta_1 = 2.25 \tag{4.104}$$

Several sample paths of the one-sided KW algorithm are shown in Figure 4-37. The paths resemble those obtained for the unnormalized window variant in Figure 4-8 . In particular, they are much smoother than those obtained for the normalized WIN algorithm in Figure 4-14. Comparison of the average paths in Figures 4-38 and 4-15 indicates that the convergence of the average parameter sample path to the optimal value is significantly slower for the KW technique. This is supported by a comparison of Figures 4-39 and 4-16, in which it is evident that convergence of the cost for the KW algorithm is slower than that of the WIN algorithm. However, the KW algorithm does exhibit lower transient error due to its more rapidly decreasing stepsize.

We now consider the two-sided KW technique applied to the same problem. For the two-sided technique we use the stepsize and perturbation sequences

$$a = 1, \quad \rho_1 = 1 \tag{4.105}$$

$$b = 1/6, \quad \delta_1 = 2.25 \tag{4.106}$$

Notice that we require that the perturbations decrease more slowly for the two-sided

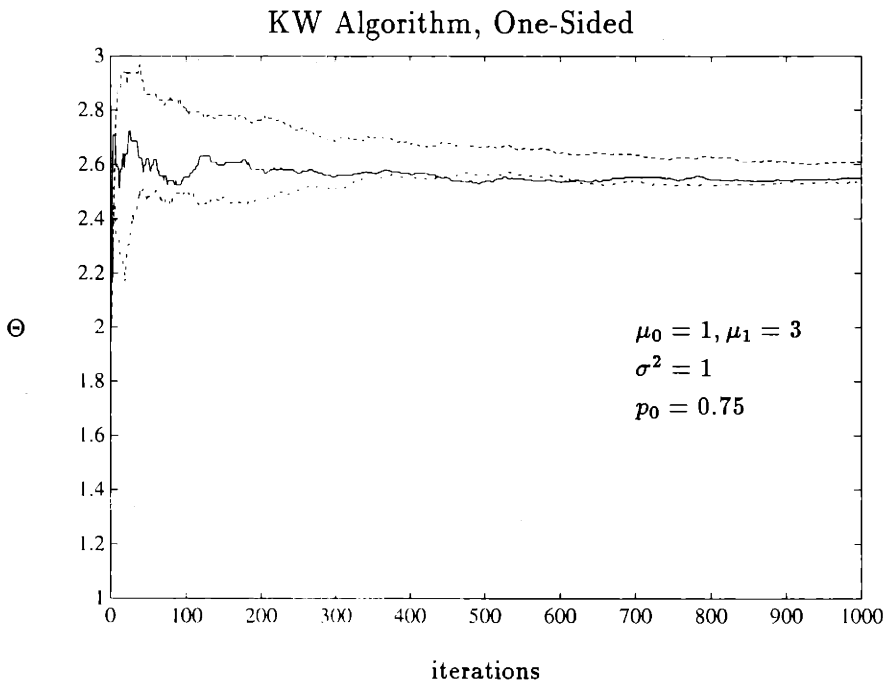


Figure 4-37: Several Sample Paths of $\{\Theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/k$, $\delta_k = 2.25/(k)^{1/4}$; Gaussian case.

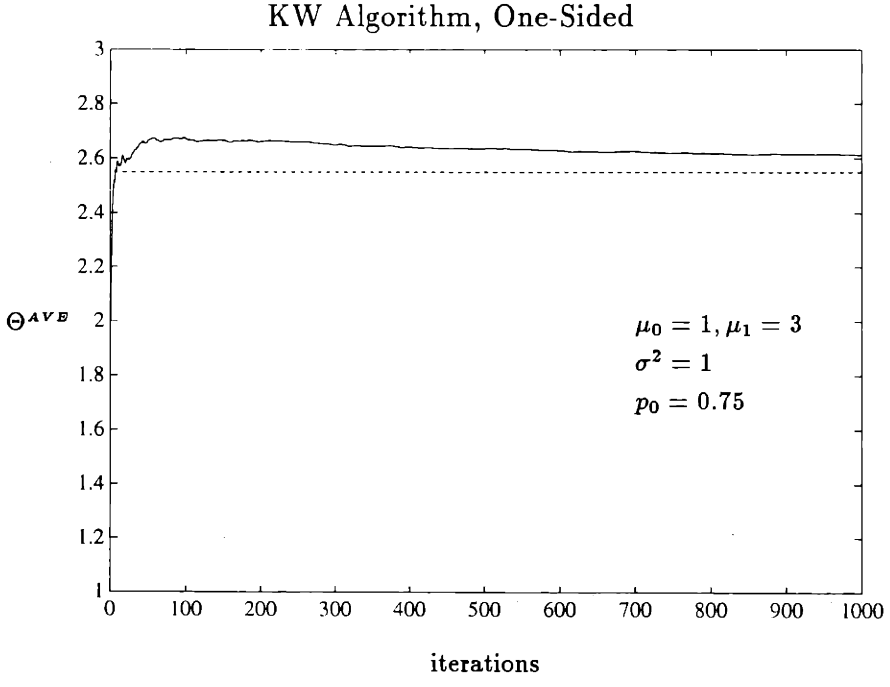


Figure 4-38: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/k$, $\delta_k = 2.25/(k)^{1/4}$. The optimal value of the threshold is 2.55 (dashed); Gaussian case.

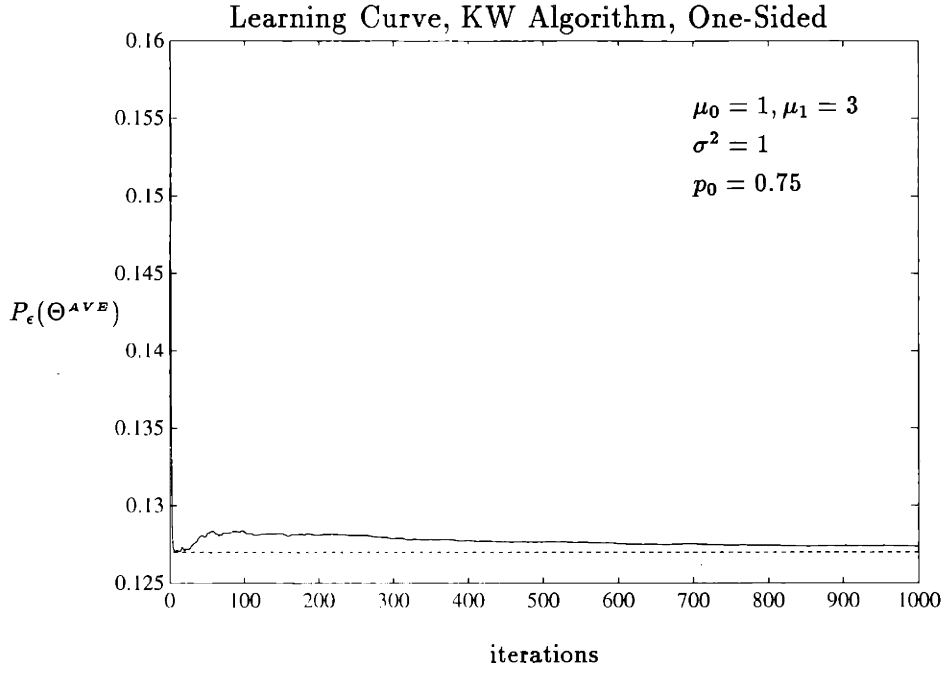


Figure 4-39: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.1270 (dashed).

method. In fact, now 10 million iterations are required to reduce the size of the perturbation step δ_k by a factor of 10. Practically speaking, the steps are computed with a constant $\delta_k = \delta$.

Several sample paths for the two-sided method are shown in Figure 4-40. In comparison to the paths for the one-sided method, these paths appear more sluggish, typically approaching the optimal value from below, rather than overshooting and coming back down. The averaged path in Figure 4-41 remains strictly below the optimum even after 1000 iterations. However, the performance of the algorithm in terms of cost reduction is still quite good as shown by Figure 4-42.

In summary, the KW algorithms were found to exhibit a slower convergence rate than the WIN algorithm on a similar problem instance, and the performance of the one-sided and two-sided methods was found to be comparable on the single example attempted here.

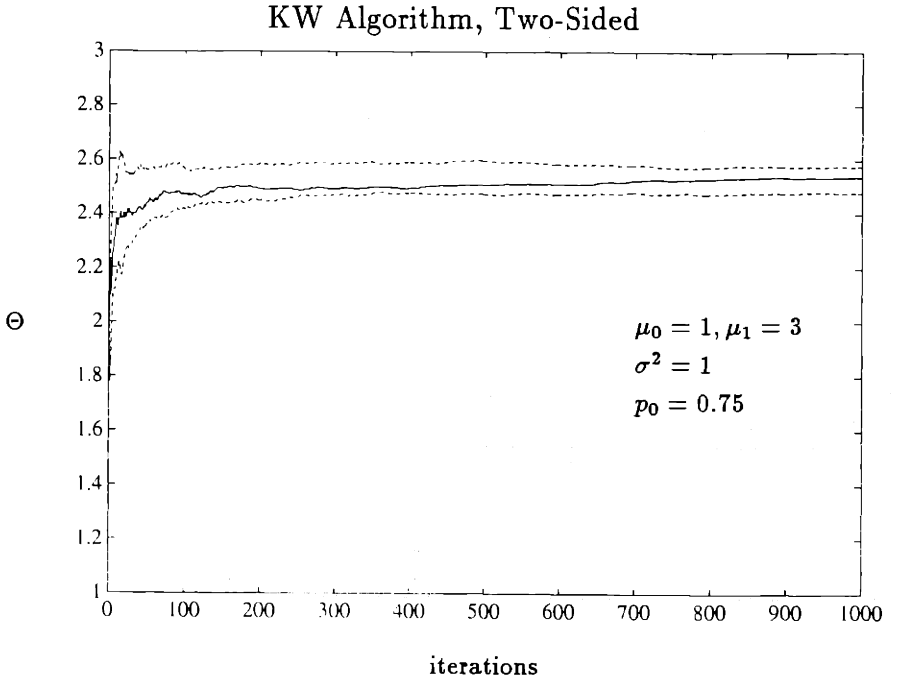


Figure 4-40: Several Sample Paths of $\{\theta_k\}$ during training. $\Theta_1 = 2$ for all paths, and $\rho_k = 1/k$, $\delta_k = 2.25/(k)^{1/6}$; Gaussian case.

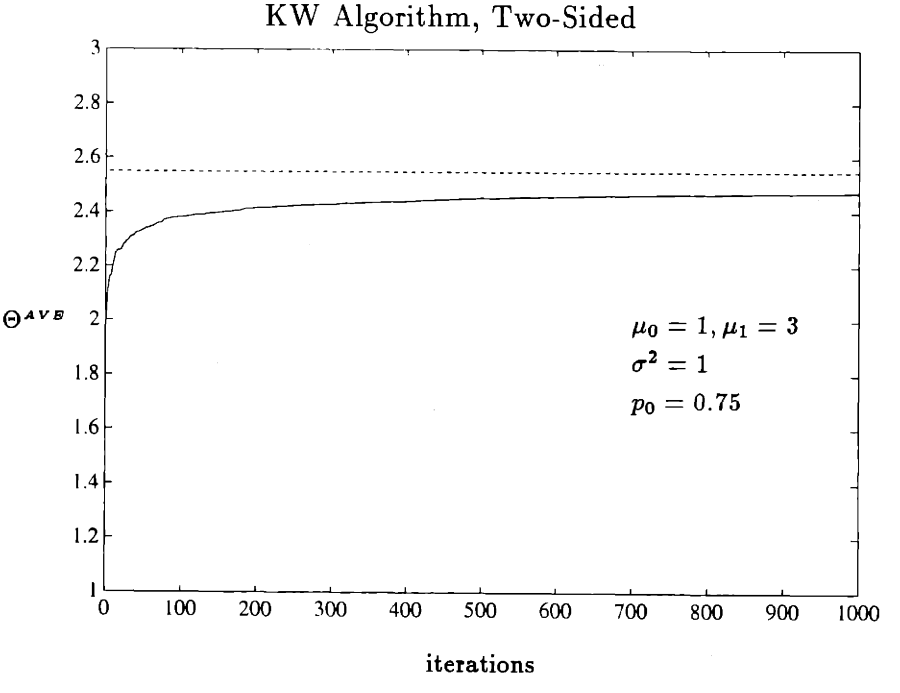


Figure 4-41: Motion of $\{\Theta_k^{AVE}\}$ during training (solid). Each point on the path represents an average over 15 Monte-Carlo runs. Each sample path began at $\Theta_1 = 2$, with $\rho_k = 1/k$, $\delta_k = 2.25/(k)^{1/6}$. The optimal value of the threshold is 2.55 (dashed); Gaussian case.

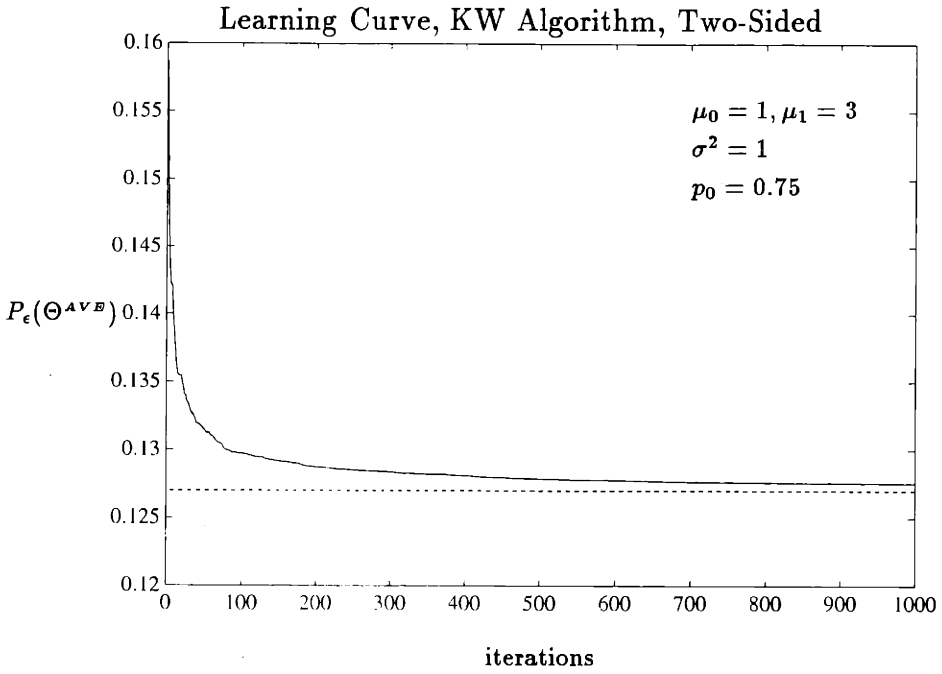


Figure 4-42: Sample Path of $\{P_\epsilon(\Theta_k^{AVE})\}$ (solid). The optimal value is 0.1270 (dashed).

4.5 Chapter Conclusions

In this chapter we considered the training problem for a single DM. After specifying certain conditions on the class of allowable density functions, and requiring the measurement sequence to be independent and identically distributed, we attempted to formulate stochastic gradient training algorithms. We immediately encountered the difficulty that it appears impossible to formulate the optimization of the Bayes risk, for a linear threshold rule, using the RM technique. That is, the derivative of the cost with respect to the threshold does not appear to be a regression function. However, we were able to successfully formulate two classes of training algorithms, window algorithms (WIN) and Kiefer-Wolfowitz (KW) algorithms.

The WIN algorithms, also referred to as modified RM techniques, circumvented the difficulty of the derivative not being a regression function by constructing a sequence of approximating regression functions which converged to the true derivative. The action of these algorithms was alternatively interpreted as operating on a modi-

fied performance measure which was a smoothed version of the original measure. We considered both normalized and unnormalized variants of this technique. It is interesting that despite the highly nonlinear nature of the derivative, the updates required by the technique were quite simple.

Formulation of KW solutions was found to be easy, as these techniques rely only on samples of the cost function itself, which in our setting were easily obtained. The samples were used to construct finite difference approximations to the derivative.

Despite being derived in dissimilar fashion, the two classes of algorithms strongly resembled one another. In particular, the two-sided KW algorithm appeared as a special case of the unnormalized window algorithm using a rectangular window. We did not provide a detailed comparison of the two techniques, as our primary goal in this report is to suggest and validate various solution methodologies. However, convergence rate analyses in the literature [14] suggest that the performance of the window algorithm is superior to that of the KW algorithm. Both types of algorithms performed extremely well in our numerical experiments.

Chapter 5

Synchronous Network (Team) Training Algorithms

We have now arrived at the *network* or *team* training problem. With regard to the material in the preceding chapter, the network problem represents an extension to optimization of an imprecisely known *multivariable* function. However, this step is more than simply going from the scalar parameter to the vector parameter case because we assume the components of the parameter vector are under *distributed* control by the collection of networked DMs.

By distributed, we mean that the thresholds of each DM are under local control, with the optimization dispersed throughout the network, as opposed to having the thresholds under some form of central authority which has access to all of the information in the network. In addition, the observation and thresholds of each DM are available only locally. In a distributed setting, each DM must obtain information not only from the environment through its observation, but must somehow also obtain information about the values of other network thresholds. We are specifically interested in distributed implementations because they make it possible to incorporate notions of incomplete information and autonomous action by each DM into our models of team training, making them much more interesting and useful.

The ultimate effect of the decentralization of information in the network is to

impose an additional constraint on the training problem which must usually[1] be overcome through *communication* between the DMs. There are two key aspects of this communication to be specified; what is allowed to be communicated (content), and when it is allowed to be communicated (frequency and timing). In the present chapter, we do not wish communication constraints to be binding. Our purpose at this time is simply to investigate some feasible methods of solving the network training problem, with the primary goal of identifying *what* must be communicated, in the context of a particular distributed algorithm, in order to resolve the coupling. With respect to communication quantity, we assume that the communication is uncorrupted and unlimited in capacity. In particular, we allow for communication of real-valued quantities, despite this being admittedly at odds with the fundamental assumption giving rise to the decentralized binary hypothesis testing problem in the first place, namely that communication between DMs is restricted. However, for the time being, we defer consideration of techniques which minimize this communication and assume that during the training phase at least, full communication capacity is available. If desired, the training can be thought of as off-line.

With respect to the frequency and timing of communications, we assume that communication is instantaneous whenever it is required, so that information is received with no delay. Furthermore, we assume that whatever synchronization requirements are required by a specific algorithm are assumed to be in effect. For the most part, this consists of designating specific phases in the algorithm during which a particular activity is to be performed.

The previous stringent assumptions on the network communications effectively convert the decentralized problem into a centralized one. That is, the effects of decentralization are nullified through the unrestricted use of communication. The result of imposing these strong assumptions is that any results which guarantee convergence of the centralized versions of these algorithms will guarantee the convergence of the fully-synchronized distributed versions as well [6]. Thus, we may begin to understand the distributed problem by thinking about the impact of the decentralization of in-

[1]This term will be qualified shortly.

formation and the distributed nature of the updates before worrying about how this has impacted the convergence of the algorithms.

The algorithms of this chapter divide into two broad classes. The first class, containing the so-called WIN-type algorithms, operate by observing the effect of parameter variation only on the local output, while requiring that the rest of the network be locally modeled or represented. These techniques require extensive communication between DMs to update the local models as the parameters of the network change.

The second class, containing KW-type algorithms, operates by observing the effect of parameter variation directly at the network output. Interestingly, these techniques do not require any local model or representation of the network to be available at each node, and thus do not require any communication. The KW techniques may be termed model-free, while the WIN techniques are model-dependent. A trade-off is evident in the fact that the model-dependent techniques are able to update the parameters based only on local feedback, i.e., observing only local decision output, but require that a local model be maintained and updated through communication, while the model-free techniques require no model, and hence no communication, but do require global feedback, i.e., require that the network output be observable by every DM. For either class, however, every DM must have access to ground truth.

5.1 Network Training Problem Statement

The fact that each DM may control several threshold parameters, as discussed in Section 3.1, is rather inconvenient from a notational point of view, particularly with respect to indexing parameters. To simplify, we will distinguish between a *processor* which we define to be an updating entity which controls a single threshold parameter, and a DM, which may have associated with it several processors if it utilizes several thresholds. In other words, in the dependency graphs of Section 3.5, we associate a processor with each parameter node. We assume that processor i controls a single threshold parameter θ_i, so that our distributed implementation corresponds to what is referred to in the distributed computation literature as *specialized computation.*

There is no loss of generality in adopting this notational scheme because we have simply associated a single DM controlling $K > 1$ thresholds with K processors each controlling one threshold, and there is no coupling between the set of thresholds controlled by the same DM (Section 3.5). This conforms to our earlier interpretation of a DM receiving messages as having distinct modes of operation selected by the combination of incoming messages. Of course, the K processors all receive the same observation since they do in fact correspond to the same DM.

For a network of M DMs parameterized by N thresholds, we denote the vector of network threshold parameters by

$$\underline{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]^T \tag{5.1}$$

where N represents the total number of network thresholds, and hence the number of processors, and assume that component i is controlled by processor i.

Suppose that the network contains M DMs. Then the training datum available to the network at time k consists of the set $\{Y_{1(k)}, \ldots, Y_{M(k)}, H^k\}$, where $Y_{i(k)}$ denotes the real-valued scalar random observation of DM i at time k, and H^k denotes the acting hypothesis corresponding to the set $\{Y_{1(k)}, \ldots, Y_{M(k)}\}$. Again we assume the existence of conditional densities such that

$$Y_{i(k)} \sim \begin{cases} p_{Y_i|H_1}(y_i|H_1) & H^k = H_1 \\ p_{Y_i|H_0}(y_i|H_0) & H^k = H_0 \end{cases} \tag{5.2}$$

We make analogous assumptions on the local conditional densities to those made for the single DM case in Assumption 4.1.

Assumption 5.1 (Conditional Density Functions)

The functions $p_{Y_i|H_0}(y_i|H_0), p_{Y_i|H_1}(y_i|H_1)$ *for* $i = 1, \ldots, M$ *are:*

(a) *Continuous and twice differentiable, with bounded first and second derivatives*

(b) *Nonzero everywhere, i.e.,*

$$p_{Y_i|H_j}(y_i|H_j) > 0, \quad \forall y_i \in \Re, \; j = 0, 1 \tag{5.3}$$

The entire training data set consists of the infinite sequence $\{Y_{1(k)}, \ldots, Y_{M(k)}, H^k; \; k = 1, 2, \ldots\}$. We make analogous assumptions on the training data to those in Section 4.1.

Assumption 5.2 (Network Training Data)

Let M *be the number of network DMs. For the sequence of training data* $\{Y_{1(k)}, \ldots, Y_{M(k)}, H^k; \; k = 1, 2, \ldots\}$ *it holds that*

(a) $\{Y_{1(j)}, \ldots, Y_{M(j)}, H^j\}$ *is independent of* $\{Y_{1(k)}, \ldots, Y_{M(k)}, H^k\}$ *for all times* $j \neq k$

(b)

$$p_{Y_{i(k)}, H^k}(y_i, H) = p_{Y_i, H}(y_i, H) \tag{5.4}$$

for $i = 1, \ldots, M, \; k = 1, 2, \ldots$

(c) $Y_1, \ldots, Y_M$ *are conditionally independent given either hypothesis*

In other words, the sequence of training data is independent, identically distributed (i.e., stationary), and obeys the conditional independence assumption of Section 2.4.1.

Again we have have assumed the existence of conditional density functions according to which the data set is generated. For notational convenience, we denote by $\underline{X}_k$ the set $\{Y_1, \ldots, Y_M, H^k\}$ which we refer to as the *network measurement* at time

k. As in Chapter 4, we write $E_{\underline{X}}\{\cdot\}$ to denote $E_{Y_1,\ldots,Y_M,H}\{\cdot\}$.

In the distributed setting, we assume DM i has access to the local measurement $X_{i(k)} = \{Y_{i(k)}, H^k\}$ only, with the entire network measurement given by

$$\begin{aligned} \underline{X}_k &= \{Y_{1(k)}, \ldots, Y_{M(k)}, H^k\} \\ &= \{Y_{1(k)}, H^k\} \bigcup \{Y_{2(k)}, H^k\} \cdots \bigcup \{Y_{M(k)}, H^k\} \\ &= X_{1(k)} \bigcup X_{2(k)} \cdots \bigcup X_{M(k)} \end{aligned} \tag{5.5}$$

Note that the network measurements are not statistically independent across DMs, because of the common acting hypothesis, but conditioned on the hypothesis, the measurements are independent due to the conditional independence assumption on the network observations specified in Assumption 5.2(c).

The main problem addressed in this chapter is the following.

Problem 5.1 (Synchronous Minimum Error Network Training Problem) *Assume a DBHT network with topology Team containing M DMs which obeys the restrictions of Section 2.4.1. Also assume the availability of unrestricted synchronized communication between DMs. Then, given only a sequence of network training data* $\{Y_{1(k)}, \ldots, Y_{M(k)}, H^k; k = 1, 2, \ldots\}$ *satisfying Assumptions 5.2, determine the minimum probability of error network strategy* $\underline{\gamma}^*$ *over the class of networks of linear threshold classifiers* $\mathcal{T}$ *defined in Section 3.1, i.e., determine*

$$\underline{\gamma}^* = \arg \min_{\underline{\gamma} \in \mathcal{T}} P_\epsilon^{Team}(\underline{\gamma}) \tag{5.6}$$

Equivalently, find

$$\underline{\theta}^* = \arg \min_{\underline{\theta} \in \Re^N} P_\epsilon^{Team}(\underline{\theta}) \tag{5.7}$$

where $P_\epsilon^{Team}(\underline{\theta})$ *is the linear threshold parameterization of the team probability of error defined in Section 3.1.*

The solution to this training problem will in general be suboptimal with respect to

the best performance achievable due to the restriction to networks of linear threshold rules. In addition, in the absence of any guarantee of unimodality, our iterative gradient based techniques can at best determine only a local minimum.

5.2 Example: A Distributed Stochastic Gradient (RM-type) Algorithm

As described in Chapter 4, the RM technique may not be directly applied to Bayes risk classification problems when linear threshold parameterizations of the decision rules are used. Nevertheless, it is useful to briefly consider the hypothetical application of such a technique to the network training problem, because it is the easiest context in which to illustrate the issues which arise. Our central purpose here is to clarify what we mean by distributed stochastic optimization.

Assume we have M DMs with N associated parameters, and that it is desired to perform unconstrained optimization of $P_\epsilon(\underline{\theta}) : \Re^N \rightarrow \Re$ where $\underline{\theta} = [\theta_1, \ldots, \theta_N]$ using an RM technique, where the technique is assumed to apply. Since

$$\nabla P_\epsilon(\underline{\theta}) = \left[\frac{\partial P_\epsilon}{\partial \theta_1}(\underline{\theta}), \ldots, \frac{\partial P_\epsilon}{\partial \theta_N}(\underline{\theta})\right]^T \tag{5.8}$$

is assumed to be a vector regression function, there exists a vector-valued random variable $\underline{Z}$ such that $\nabla P_\epsilon(\underline{\theta}) = E_{\underline{X}}\{\underline{Z}(\underline{X}, \underline{\Theta}) | \underline{\Theta} = \underline{\theta}\}$ for some random vector $\underline{Z}(\underline{X}, \underline{\Theta}) = [Z_1(\underline{X}, \underline{\Theta}), \ldots, Z_N(\underline{X}, \underline{\Theta})]$. We can then solve the system of necessary conditions

$$\nabla J(\underline{\theta}) = \underline{0} \tag{5.9}$$

where $\underline{0}$ denotes the N-dimensional column vector of zeros, using the vector RM (stochastic steepest descent) algorithm

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}(\underline{X}_k, \underline{\Theta}_k); \quad k = 1, 2, \ldots \tag{5.10}$$

The data processing for this (centralized) case is illustrated in Figure 5-1.

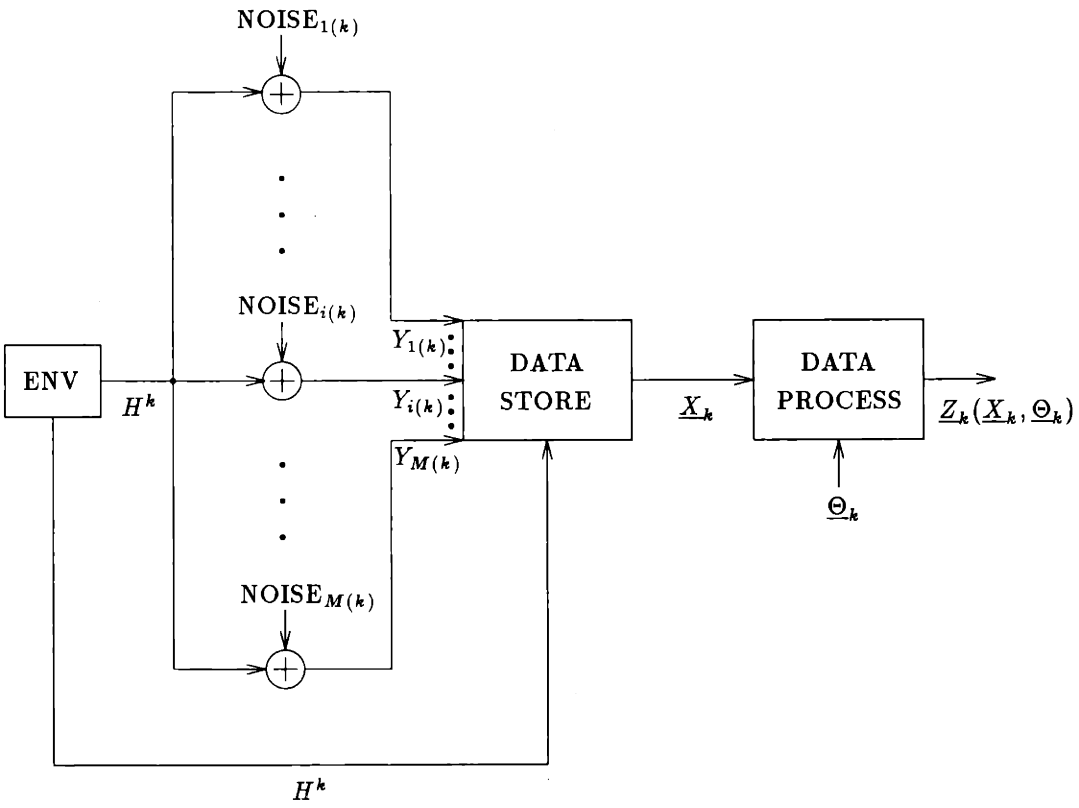


Figure 5-1: Network Data Processing: Centralized Case

Notice that in order to compute the vector-valued step, the measurement information must be centrally processed. Hence the observed fan-in of measurement information.

Now suppose that $M = 2$, $N = 2$, $P_\epsilon : \Re^2 \mapsto \Re$, and that the optimization is to be solved in distributed fashion by two cooperating processors, each of which has access to a part of the measurement vector $\underline{X}_k$, and each of which controls one of the threshold parameters (specialized computation). For example, suppose processor 1 has access to X_1 and updates θ_1 while processor 2 has access to X_2 and updates θ_2, where $\underline{X} = X_1 \cup X_2$. The schematic of Figure 5-2(a) illustrates this scenario. Information concerning the parameter values and measurements is assumed to be available only locally, so that the other processor must always be informed of the values through communication. In general, the partial derivative $\partial P_\epsilon(\underline{\theta})/\partial\theta_i, i = 1,2$ is a function of $\theta_j, j \neq i$, so that the value of θ_j must be communicated from processor j to processor i at each iteration. In order to compute estimates of the partial derivatives $Z_{1(k)}$ and $Z_{2(k)}$, complete measurement information is required by each processor as well.

If the two processors can be synchronized in time, then at each time k they can communicate the necessary information to one another, compute the partial derivatives, and update their respective components simultaneously. In other words, the coupled local iterations

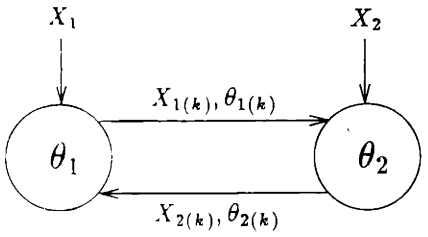
$$\Theta_{1(k+1)} = \Theta_{1(k)} - \rho_k Z_{1(k)}(X_{1(k)}, X_{2(k)}, \Theta_{1(k)}, \Theta_{2(k)}) \tag{5.11}$$

$$\Theta_{2(k+1)} = \Theta_{2(k)} - \rho_k Z_{2(k)}(X_{1(k)}, X_{2(k)}, \Theta_{1(k)}, \Theta_{2(k)}), \quad k = 1,2,\ldots \tag{5.12}$$

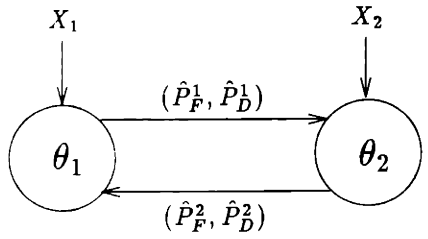
running synchronously would converge, under assumption of perfect communication channels[2], precisely under those conditions for which the fully centralized solution would converge, since the updates computed by the two would in fact be identical.

As a prelude to the discussion of the WIN algorithms, suppose that the dependence of $Z_{i(k)}$ on both $\theta_{j(k)}, X_{j(k)}$, where $i \neq j$, can be summarized by evaluating

[2]The precise meaning of this statement is no noise and no delays.



(a)



(b)

Figure 5-2: Specialized Distributed Computation. (a) Mimicking the fully centralized case by communicating all required quantities. (b) Communication of locally computed estimated operating points which summarizes all information required by other processor to update its parameter.

another parameter, and communicating that parameter instead. Such a parameter is reminiscent of a sufficient statistic. This is the case for the model-based WIN algorithms, for which the estimated operating point $(\hat{P}_F^j, \hat{P}_D^j)$ effectively summarizes all the information about $\theta_{j(k)}, X_{j(k)}$ needed for the updates $Z_{i(k)}$, $i \neq j$ to be computed. Communication of the estimated operating points in place of the threshold parameters and observations is illustrated in Figure 5-2(b).

The key point of this discussion is that distributed implementations of stochastic optimization algorithms are complicated by the necessity of each processor to obtain information concerning the *measurements* of the other processors, as well as the value of their parameters, to perform each local update.

5.2.1 Distributed Computation: Key Implementation Issues

This simple example of distributed computation has raised a variety of implementation issues. We wish to comment briefly at this point on some of the ways that the implementations of the synchronous algorithms we are about to present could be generalized, particularly with the modeling issues and applications suggested in Chapter 1 in mind. We do this now in order to sensitize the reader who is unfamiliar with distributed algorithms to some of these generalizations, so that they are kept in mind during the ensuing discussion. These topics are subsequently addressed in Chapters 6 and 7.

Timing and Synchronization of Updates and Communication The primary difficulties in distributed implementations of optimization algorithms center around coordination issues; in order to analyze these algorithms as equivalent to their centralized counterparts, the information required by a particular processor from the other processors must arrive uncorrupted and in time for all the processors to *simultaneously* execute an update. The difficulties in ensuring that such conditions prevail in a particular situation may be insurmountable, and we would certainly like to demonstrate convergence of the algorithms under far less restrictive conditions

than these. Demonstrating that convergence is preserved in the presence of random updating times and outdated information is the central topic of Chapter 7.

Processor Activities As the overall effort required to perform the optimization is parsed among the processors in distributed implementations, it is easy to imagine that all processors might not be engaged in the same activity at the same time.

As will become clear shortly, the network training algorithms we consider involve each processor executing one or more of the following activities at each instant of time. Each processor

(1) *updates* the parameter under its local control

(2) *estimates* the operating point corresponding to its parameter

(3) *receives* an external measurement X, or an internal measurement generated from within the network

(4) *transmits* information to another network DM

(5) remains *idle*

Although it is not necessary when discussing synchronous algorithms to develop a formal mathematical characterization of the processors' behavior, in Chapter 7 we develop such a framework for the asynchronous setting.

Local Stepsizes Although in the present chapter we take the stepsize and window width/perturbation sequences to be the same for every processor, it may be desirable to allow each component to be updated using a different local stepsize rule, so that for example iterations (5.11), (5.12) might become

$$\begin{aligned} \Theta_{1(k+1)} &= \Theta_{1(k)} - \rho_k^{(1)} Z_{1(k)}(X_{1(k)}, X_{2(k)}, \Theta_{1(k)}, \Theta_{2(k)}) \\ \Theta_{2(k+1)} &= \Theta_{2(k)} - \rho_k^{(2)} Z_{2(k)}(X_{1(k)}, X_{2(k)}, \Theta_{1(k)}, \Theta_{2(k)}) \\ & \qquad k = 1, 2, \ldots \end{aligned} \tag{5.13}$$

where $\{\rho_k^{(1)}\}$ and $\{\rho_k^{(2)}\}$ are possibly different stepsize sequences. So long as certain conditions on the stepsize rules are obeyed this generalization presents no mathematical difficulty other than to complicate the algebra in the analysis. Analysis using

different stepsize rules to update each component is presented in Chapter 6.

5.3 WIN-Type Training Algorithms

In Chapter 4 it was shown that the RM approach could be modified to determine a root of the equation

$$\frac{dJ_B}{d\theta}(\theta) = 0 \tag{5.14}$$

where

$$\frac{dJ_B}{d\theta}(\theta) = -\lambda_0 p_0 p_{Y|H_0}(\theta|H_0) + \lambda_1 p_1 p_{Y|H_1}(\theta|H_1) \tag{5.15}$$

despite the fact that this derivative is not a regression function. The method operated on a sequence of regression functions which converge to the true derivative in the limit.

The success of the WIN algorithm in determining a root of the scalar parameter necessary condition in the single DM problem suggests that it may also prove effective for determining a solution to the nonlinear gradient equation

$$\nabla P_\epsilon^{Team}(\underline{\theta}) = \underline{0} \tag{5.16}$$

where $\nabla P_\epsilon^{Team}(\underline{\theta})$ is the N-dimensional vector

$$\nabla P_\epsilon^{Team}(\underline{\theta}) = \left[\frac{\partial P_\epsilon^{Team}}{\partial \theta_1}(\underline{\theta}), \frac{\partial P_\epsilon^{Team}}{\partial \theta_2}(\underline{\theta}), \ldots, \frac{\partial P_\epsilon^{Team}}{\partial \theta_N}(\underline{\theta})\right]^T \tag{5.17}$$

and each component is of the form

$$\frac{\partial P_\epsilon^{Team}}{\partial \theta_l}(\underline{\theta}) = -\lambda_0^{il}(\underline{\theta}) p_0 p_{Y_i|H_0}(\theta_l|H_0) + \lambda_1^{il}(\underline{\theta}) p_1 p_{Y_i|H_1}(\theta_l|H_1) \tag{5.18}$$

where θ_l corresponds to DM i in topology $Team$. The form of the partial derivative was presented in Proposition 3.6, and indicates that dependence of each local threshold parameter on the value of the other thresholds is captured by the scalar coupling costs $\lambda_0^{il}(\underline{\theta})$ and $\lambda_1^{il}(\underline{\theta})$. Solutions to this system of equations satisfy the necessary conditions for optimality for $Team$.

Specifically, recall that for the linear parameterization of the network decision rules, the partial derivatives for 2-Tand were given by

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \alpha}(\alpha,\beta_0,\beta_1) &= -[P_F^{B1}(\beta_1) - P_F^{B0}(\beta_0)]p_0 p_{Y_A|H_0}(\alpha|H_0) \\
&\quad +[P_D^{B1}(\beta_1) - P_D^{B0}(\beta_0)]p_1 p_{Y_A|H_1}(\alpha|H_1) \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_0}(\alpha,\beta_0,\beta_1) &= -[1 - P_F^A(\alpha)]p_0 p_{Y_B|H_0}(\beta_0|H_0) + [1 - P_D^A(\alpha)]p_1 p_{Y_B|H_1}(\beta_0|H_1) \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_1}(\alpha,\beta_0,\beta_1) &= -P_F^A(\alpha)p_0 p_{Y_B|H_0}(\beta_1|H_0) + P_D^A(\alpha)p_1 p_{Y_B|H_1}(\beta_1|H_1) \qquad (5.19)
\end{aligned}$$

Expressing these equations in terms of the coupling costs, and making the dependence of the coupling costs on the network operating points clear, we can express system (5.19) as

$$\begin{aligned}
\frac{\partial P_\epsilon^{2-Tand}}{\partial \alpha}(\alpha,\beta_0,\beta_1) &= -\lambda_0^A(P_F^{B0}, P_F^{B1})\, p_0 p_{Y_A|H_0}(\alpha|H_0) + \lambda_1^A(P_D^{B0}, P_D^{B1})\, p_1 p_{Y_A|H_1}(\alpha|H_1) \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_0}(\alpha,\beta_0,\beta_1) &= -\lambda_0^{B0}(P_F^A)\, p_0 p_{Y_B|H_0}(\beta_0|H_0) + \lambda_1^{B0}(P_D^A)\, p_1 p_{Y_B|H_1}(\beta_0|H_1) \\
\frac{\partial P_\epsilon^{2-Tand}}{\partial \beta_1}(\alpha,\beta_0,\beta_1) &= -\lambda_0^{B1}(P_F^A)\, p_0 p_{Y_B|H_0}(\beta_1|H_0) + \lambda_1^{B1}(P_D^A)\, p_1 p_{Y_B|H_1}(\beta_1|H_1) \qquad (5.20)
\end{aligned}$$

where

$$\begin{aligned}
\lambda_0^A(P_F^{B0}, P_F^{B1}) &= [P_F^{B1} - P_F^{B0}] \\
\lambda_1^A(P_D^{B0}, P_D^{B1}) &= [P_D^{B1} - P_D^{B0}] \qquad (5.21)
\end{aligned}$$

$$\lambda_0^{B0}(P_F^A) = [1 - P_F^A], \quad \lambda_1^{B0}(P_D^A) = [1 - P_D^A] \qquad (5.22)$$

$$\lambda_0^{B1}(P_F^A) = P_F^A, \quad \lambda_1^{B1}(P_D^A) = P_D^A \qquad (5.23)$$

In a distributed implementation in which a single processor is dedicated to update each threshold parameter, the updates of threshold θ_l require evaluation of the coeffi-

cients λ_0^{il} and λ_1^{il} which represent costs or biases on the adjustment of that component which must be explicitly modeled by the locally running algorithm algorithm since they are not inherent in the measurement data. These costs capture all the necessary information concerning the current state of the other network DMs which is necessary for a given processor to adjust its threshold to the optimal team value. These coefficients are a model or representation of the rest of the network which appropriately bias the local decision problem to account for the presence of the rest of the team.

The problem of solving the system of necessary conditions for the network problem looks, on a component by component basis, as if a general Bayes problem must be solved for each parameter, given that the other parameters are held fixed. We have a technique for solving the one-dimensional problem along each coordinate. When expressed in the form (5.20), it is evident that each component partial derivative is of the same form as the general Bayes problem described in Section 4.3.1, and therefore should be amenable to solution by window methods. However, the coupling coefficients are not exactly computable without knowledge of the functional form of the conditional probability densities. The key idea in developing training algorithms is that these quantities may be nonparametrically estimated. Each processor i may estimate the operating point corresponding to a given value of its threshold θ_i by executing a series of local decision trials using some number $N \geq 1$ local measurements $\{Y_{i(k)}, H^k; k = 1, \ldots, N\}$ and then computing the empirical relative frequencies

$$\hat{P}_F = \frac{N_F}{N_{H_0}}, \quad \hat{P}_D = \frac{N_D}{N_{H_1}} \tag{5.24}$$

where N_F denotes the number of times over the N measurements that $U_i = 1$ when $H = H_0$, N_D denotes the number of times over the N measurements that $U_i = 0$ when $H = H_1$, N_{H_0} is the number of occurrences of hypothesis H_0 in the string of N measurements, and N_{H_1} is the number of occurrences of hypothesis H_1. Properties of relative frequency estimates of fixed but unknown probabilities are discussed at length in Appendix A. Note that since several processors may be associated with a single DM, the same set of local measurements must be used to estimate each of that

DM's operating points. We may imagine that the set of measurements are stored and commonly available to those processors updating that DM's thresholds, or that the estimation phase is made sufficiently long so that the set of measurements in the estimation phase may be parsed, with a different subset being used to update each parameter. For the time being, how this is implemented is not really of consequence.

Estimates of the coupling costs can then be constructed by evaluating them at the estimated operating points. For example, for 2-Tand, the estimated costs would be computed as

$$\begin{aligned} \hat{\lambda}_0^A(\hat{P}_F^{B0}, \hat{P}_F^{B1}) &= [\hat{P}_F^{B1} - \hat{P}_F^{B0}] \\ \hat{\lambda}_1^A(\hat{P}_D^{B0}, \hat{P}_D^{B1}) &= [\hat{P}_D^{B1} - \hat{P}_D^{B0}] \end{aligned} \tag{5.25}$$

$$\hat{\lambda}_0^{B0}(\hat{P}_F^A) = [1 - \hat{P}_F^A], \quad \hat{\lambda}_1^{B0}(\hat{P}_D^A) = [1 - \hat{P}_D^A] \tag{5.26}$$

$$\hat{\lambda}_0^{B1}(\hat{P}_F^A) = \hat{P}_F^A, \quad \hat{\lambda}_1^{B1}(\hat{P}_D^A) = \hat{P}_D^A \tag{5.27}$$

It is evident that the analytic form of each network partial derivative must be known to the corresponding processor in order to structure the computation of the coupling costs. This is equivalent to saying that each DM must know how it is tied in structurally to the rest of the organization, but can be initially naive to the capabilities[3] of the other DMs since it can infer them during training. It is also clear that as the size of the networks grow, the complexity of computing the coupling coefficients grows as well. For 3-Vee, the required computations have already grown to

$$\begin{aligned} \hat{\lambda}_0^A &= [(1 - \hat{P}_F^B)(\hat{P}_F^{C(10)} - \hat{P}_F^{C(00)}) \\ &\quad + \hat{P}_F^B(\hat{P}_F^{C(11)} - \hat{P}_F^{C(01)})] \\ \hat{\lambda}_1^A &= [(1 - \hat{P}_D^B)(\hat{P}_D^{C(10)} - \hat{P}_D^{C(00)}) \\ &\quad + \hat{P}_D^B(\hat{P}_D^{C(11)} - \hat{P}_D^{C(01)})] \\ \hat{\lambda}_0^B &= [(1 - \hat{P}_F^A)(\hat{P}_F^{C(01)} - \hat{P}_F^{C(00)}) \end{aligned}$$

[3] As measured by the locus of attainable (P_F, P_D) values, i.e., its ROC curve.

$$+\hat{P}_F^A(\hat{P}_F^{C(11)} - \hat{P}_F^{C(10)})]$$

$$\hat{\lambda}_1^B = [(1 - \hat{P}_D^A)(\hat{P}_D^{C(01)} - \hat{P}_D^{C(00)}) + \hat{P}_D^A(\hat{P}_D^{C(11)} - \hat{P}_D^{C(10)})] \quad (5.28)$$

$$\hat{\lambda}_0^{C(00)} = [(1 - \hat{P}_F^A)(1 - \hat{P}_F^B)], \quad \hat{\lambda}_1^{C(00)} = [(1 - \hat{P}_D^A)(1 - \hat{P}_D^B)] \quad (5.29)$$

$$\hat{\lambda}_0^{C(01)} = [(1 - \hat{P}_F^A)\hat{P}_F^B], \quad \hat{\lambda}_1^{C(01)} = [(1 - \hat{P}_D^A)\hat{P}_D^B] \quad (5.30)$$

$$\hat{\lambda}_0^{C(10)} = [\hat{P}_F^A(1 - \hat{P}_F^B)], \quad \hat{\lambda}_1^{C(10)} = [\hat{P}_D^A(1 - \hat{P}_D^B)] \quad (5.31)$$

$$\hat{\lambda}_0^{C(11)} = [\hat{P}_F^A\hat{P}_F^B], \quad \hat{\lambda}_1^{C(11)} = [\hat{P}_D^A\hat{P}_D^B] \quad (5.32)$$

That the suggested estimation scheme is in fact sufficient to demonstrate convergence is proven in Chapter 6. As it turns out, all that is required of the estimated coupling costs is that they be *unbiased, bounded variance* estimates of the true costs. The above scheme yields unbiased estimates of the costs because the conditional probabilities which are multiplicatively combined when computing the cost are always generated based on conditionally independent observations. Thus, the assumptions of independent observations and tree-type topologies prove crucial to the success of the algorithm. That the estimates are also bounded variance follows from the properties of empirical relative frequency estimators.

One of the more surprising implications of this fact is that convergence is guaranteed even for the cases $N_{H_0} = N_{H_1} = 1$, i.e., even if the conditional probabilities associated with each operating point estimate are replaced with 0-1 estimates. This implies that the local model can be quite crude, so long as it is correct in an average sense. The fact that convergence can be guaranteed for a wide range of quality in the local models means indicates the potential for exploring many interesting issues, such as the trade-off between the quality of the local model and the overhead associated with the additional required estimation steps. Another interesting question concerns whether it benefits convergence of the team to have the operating points of certain

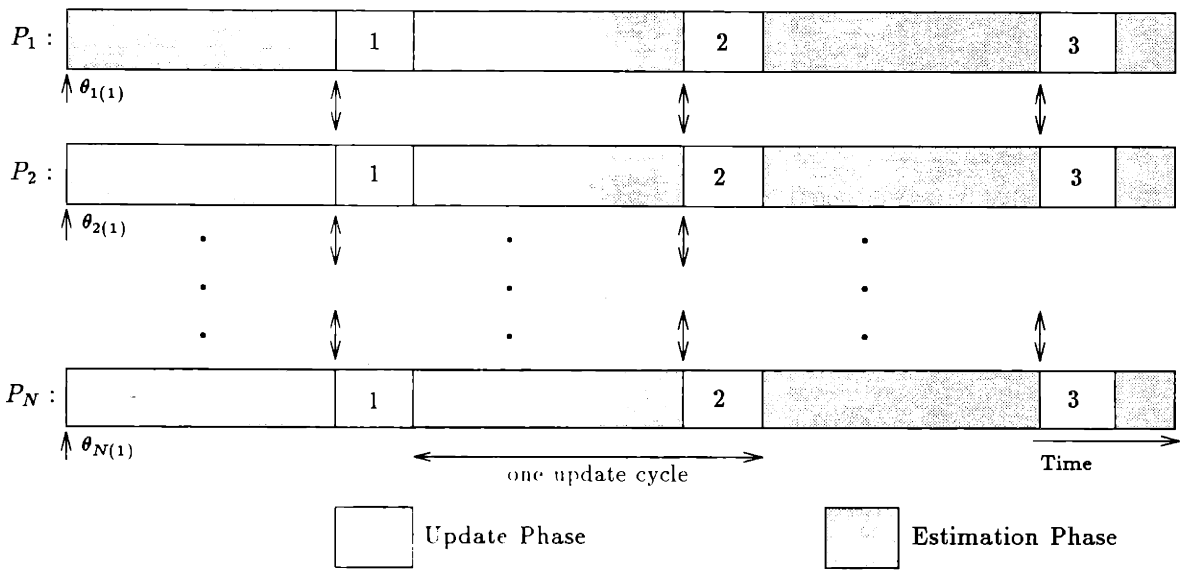


Figure 5-3: Timing diagram indicating alternating estimation and update phases. Two-sided vertical arrows represent communication and reception of estimated operating points between all DMs.

"key" DMs more accurately modeled than others, for example the primary DM. More generally, should DMs closer to the root node update with better local models since the sensitivity of the cost to their errors is correspondingly higher.

We now present three model-dependent algorithms based on the above discussion. The first algorithm, denoted WIN, is essentially a Jacobi iteration on the gradient equation. The WIN-GS algorithm is a Gauss-Seidel component solution variation, motivated by the discussion in Section 3.5. Finally, the WIN-BP algorithm utilizes the optimal control formulation of Section 3.2.3 to structure the computation of the coupling costs such that inter-DM communication is minimized.

5.3.1 Multivariable Window Algorithms (WIN)

The discussion of the previous section suggests a synchronous network training algorithm which comprises alternating estimation and update phases as illustrated by the timing diagram in Figure 5-3.

The algorithm consists of Jacobi-type iterations in which estimation and update phases are executed simultaneously at all processors. The algorithm is initialized with

arbitrary threshold settings $\theta_{1(1)}, \ldots, \theta_{N(1)}$. Each processor then estimates the operating point corresponding to its initial threshold setting. Once this state estimation process is complete, the estimated operating point of each processor is communicated to all processors requiring it to update. Upon receiving the operating point estimates, the processors combine them to compute the local estimated coupling costs. The processors then perform a simultaneous update, and the cycle begins again.

Each estimation phase requires execution of a number of local decision trials based on local measurements to compute the estimated operating points. Each update phase requires an additional local measurement to compute the step. The data processing dictated by the scheme is shown in Figure 5-4. Each parallel branch indicates the data processing by a single processor. Where processors i and j are associated with the same DM, the measurement sequences $\{X_{i(k)}\}$, $\{X_{j(k)}\}$ are the same for each. Equivalently, the sequences $\{NOISE_{i(k)}\}$ and $\{NOISE_{j(k)}\}$ are the same for each. The switches represent simultaneous transitions between estimation and update phases. Notice the parallel structure of the data processing, where the branches are cross-coupled through the operating point estimates.

Before characterizing the updates, a notational remark is in order. In the equations that follow, the variable k indexes only updates, and not measurements, a notation we adopt for simplicity. The complication arises because there is now a discrepancy between the measurement index and the update index due to the presence of the estimation phases, which also involve the processing of measurements. The equations which follow treat the estimation phases as if they take zero time (no iterations) to complete.

Concerning the updates, each processor iterates on a single component of the network parameter vector $\underline{\theta}_k$, where component θ_l associated with DM i is updated by processor l according to the local iteration

$$\theta_{l(k+1)} = \theta_{l(k)} - \rho_k Z_{l(k)}(X_{l(k)}, \hat{\lambda}^{il}_{0(k)}, \hat{\lambda}^{il}_{1(k)}, \theta_{l(k)}, \delta_k), \quad k = 1, 2, \ldots \tag{5.33}$$

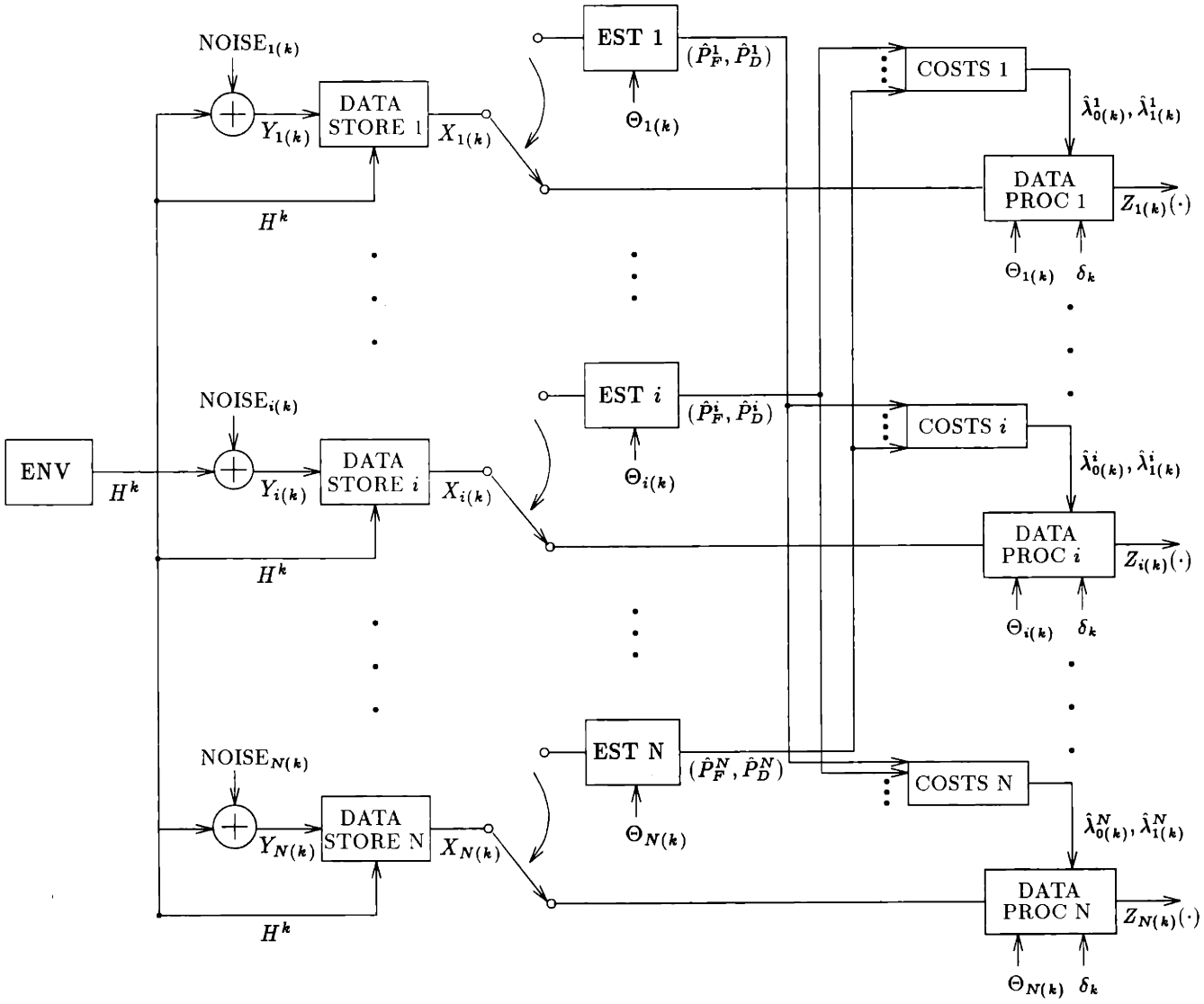


Figure 5-4: Data Processing, Team WIN Algorithm

For the unnormalized window variant the iteration takes the form

$$\theta_{l(k+1)} = \begin{cases} \theta_{l(k)} + \rho_k \hat{\lambda}^{il}_{0(k)} h(y_{i(k)} - \theta_{l(k)}, \delta_k) & \text{if } H^k = H_0 \\ \theta_{l(k)} - \rho_k \hat{\lambda}^{il}_{1(k)} h(y_{i(k)} - \theta_{l(k)}, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{5.34}$$

and for the normalized variant

$$\theta_{l(k+1)} = \begin{cases} \theta_{l(k)} + \rho_k \hat{\lambda}^{il}_{0(k)} 2\delta_k h(y_{i(k)} - \theta_{l(k)}, \delta_k) & \text{if } H^k = H_0 \\ \theta_{l(k)} - \rho_k \hat{\lambda}^{il}_{1(k)} 2\delta_k h(y_{i(k)} - \theta_{l(k)}, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{5.35}$$

As a specific example, for the 2-Tand network these computations take the following form. To begin the computation, the thresholds are initialized to arbitrary settings α_1, $\beta_{0(1)}$ and $\beta_{1(1)}$. Estimates of the operating points $(\hat{P}^{A}_{F(1)}, \hat{P}^{A}_{D(1)})$, $(\hat{P}^{B0}_{F(1)}, \hat{P}^{B0}_{D(1)})$, $(\hat{P}^{B1}_{F(1)}, \hat{P}^{B1}_{D(1)})$ corresponding to these initializations are obtained using local measurements. Each processor then communicates its estimated operating point to the other processors, and a single simultaneous update of each parameter is made according to the window-type iterations

$$\alpha_{k+1} = \begin{cases} \alpha_k + \rho_k \hat{\lambda}^{A}_{0(k)} 2\delta_k h(y_{A(k)} - \alpha_k, \delta_k) & \text{if } H^k = H_0 \\ \alpha_k - \rho_k \hat{\lambda}^{A}_{1(k)} 2\delta_k h(y_{A(k)} - \alpha_k, \delta_k) & \text{if } H^k = H_0 \end{cases} \tag{5.36}$$

$$\beta_{0(k+1)} = \begin{cases} \beta_{0(k)} + \rho_k \hat{\lambda}^{B0}_{0(k)} 2\delta_k h(y_{B(k)} - \beta_{0(k)}, \delta_k) & \text{if } H^k = H_0 \\ \beta_{0(k)} - \rho_k \hat{\lambda}^{B0}_{1(k)} 2\delta_k h(y_{B(k)} - \beta_{0(k)}, \delta_k) & \text{if } H^k = H_0 \end{cases} \tag{5.37}$$

$$\beta_{1(k+1)} = \begin{cases} \beta_{1(k)} + \rho_k \hat{\lambda}^{B1}_{0(k)} 2\delta_k h(y_{B(k)} - \beta_{1(k)}, \delta_k) & \text{if } H^k = H_0 \\ \beta_{1(k)} - \rho_k \hat{\lambda}^{B1}_{1(k)} 2\delta_k h(y_{B(k)} - \beta_{1(k)}, \delta_k) & \text{if } H^k = H_0 \end{cases} \tag{5.38}$$

which we have shown for the normalized variant, and where the estimated costs are as defined in (5.25) - (5.27). For the particular case when h is a rectangular window function, these iterations take the form

$$\alpha_{k+1} = \begin{cases} \alpha_k + \hat{\lambda}^{A}_{0(k)} \rho_k & \text{if } |\alpha_k - y_{A(k)}| \leq \delta_k,\ H^k = H_0 \\ \alpha_k & \text{if } |\alpha_k - y_{A(k)}| > \delta_k \\ \alpha_k - \hat{\lambda}^{A}_{1(k)} \rho_k & \text{if } |\alpha_k - y_{A(k)}| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{5.39}$$

$$\beta_{0(k+1)} = \begin{cases} \beta_{0(k)} + \hat{\lambda}^{B0}_{0(k)}\rho_k & \text{if } |\beta_{0(k)} - y_{B(k)}| \leq \delta_k,\ H^k = H_0 \\ \beta_{0(k)} & \text{if } |\beta_{0(k)} - y_{B(k)}| > \delta_k \\ \beta_{0(k)} - \hat{\lambda}^{B0}_{1(k)}\rho_k & \text{if } |\beta_{0(k)} - y_{B(k)}| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{5.40}$$

$$\beta_{1(k+1)} = \begin{cases} \beta_{1(k)} + \hat{\lambda}^{B1}_{0(k)}\rho_k & \text{if } |\beta_{1(k)} - y_{B(k)}| \leq \delta_k,\ H^k = H_0 \\ \beta_{1(k)} & \text{if } |\beta_{1(k)} - y_{B(k)}| > \delta_k \\ \beta_{1(k)} - \hat{\lambda}^{B1}_{1(k)}\rho_k & \text{if } |\beta_{1(k)} - y_{B(k)}| \leq \delta_k,\ H^k = H_1 \end{cases} \tag{5.41}$$

The algorithm operates in a similar fashion for any network. In practice, the estimated costs may be successfully normalized, although the analysis of such schemes is sufficiently complicated that we omit consideration of them when proving convergence in Chapter 6.

An obvious drawback of this scheme is that each network update may be quite expensive due to the (possibly many) estimation steps, and the extensive communication required to compute the resulting coupling cost estimates. This fact, in combination with the inter-parameter dependence indicated by the DAGs in Chapter 3, suggests that a Gauss-Seidel implementation, in which multiple updates of a single parameter are performed between each update, might have desirable properties.

5.3.2 Gauss-Seidel Implementation (WIN-GS)

A Gauss-Seidel implementation of the WIN scheme discussed above was introduced by Wissinger and Athans in [77]. A typical timing diagram for this algorithm is shown in Figure 5-5.

The algorithm requires the same computations as the WIN algorithm, but updates the coordinates one at a time, with multiple update steps being followed by a single estimation phase to estimate the new operating point. When all of the parameters have been updated, the update cycle is complete and begins again. Note that processors which are not updating or estimating remain idle[4], and only one processor is updating or estimating at a time. Furthermore, since communications occur only at

[4]Sitting idle does not preclude the reception of communicated messages.

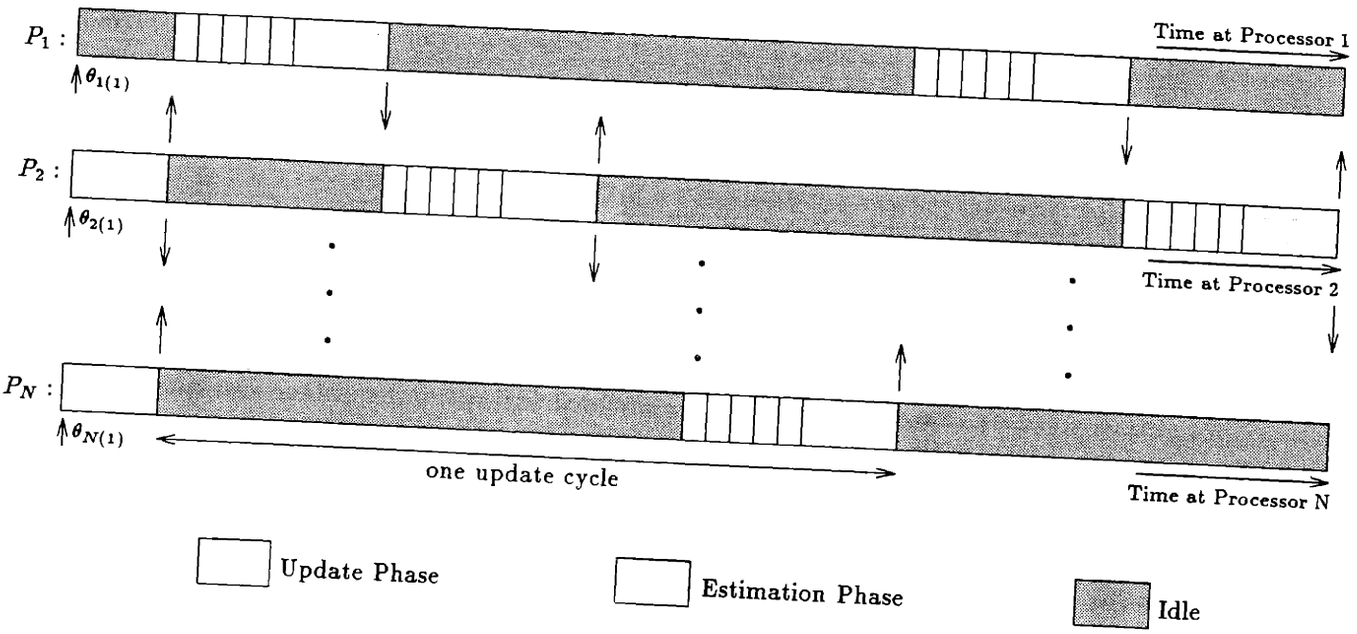


Figure 5-5: Timing diagram indicating cyclical updating scheme of Gauss-Seidel method. Vertical arrows indicate operating point information being transmitted to other processors.

the end of estimation phases, updates are much more frequent than communications in this scheme.

There are an infinite number of variations of this algorithm, which differ in the exact number of updates performed on each component during each update cycle of the algorithm. Because of this, these variants are more easily handled collectively as special cases of asynchronism in Chapter 7. In the limit, as the number of iterations on a component become large, the parameter converges[5] to an approximate person-by-person optimal value. The term approximate refers to the fact that the costs are only estimated, so that the true operating points of the other DMs remain unknown, and the subproblem is truncated after some finite number of iterations. To qualify, suppose that parameter θ_l of DM i is being updated and the other threshold parameters $\theta_j, j \neq l$ are held fixed. Then, in the limit the algorithm would determine a value of

[5] Assuming the conditional densities have a single point of intersection and certain conditions on the stepsizes and costs hold. This is covered in more detail in Chapter 6.

the parameter θ_l^* such that

$$\hat{\lambda}_0^{il}(\underline{\theta})p_0 p_{Y_i|H_0}(\theta_l^*|H_0) = \hat{\lambda}_1^{il}(\underline{\theta})p_1 p_{Y_i|H_1}(\theta_l^*|H_1) \tag{5.42}$$

where the costs $\hat{\lambda}_0^{il}$ and $\hat{\lambda}_1^{il}$ are computed based on estimates of the current operating points of the other DMs. Thus, the algorithm can be thought of as approximating the Gauss-Seidel component solution method described in Section 3.5 for the system of equations describing the necessary conditions for optimality. One interpretation is that the subproblem solved by each processor is to approximately determine the point of intersection of its scaled local densities, where the scaling is determined by estimates of the current values of the other network operating points. Note that this local problem is not guaranteed to be well-posed since it is conceivable that one of the estimated costs is negative while the other is positive, implying that no solution exists. What we find in Chapter 7, however, is that so long as we bound the number of iterations performed in each subproblem, that knowing the correct cost in an average sense is still sufficient to guarantee convergence. As a practical matter, however, the algorithm is observed to perform significantly better if the overhead is paid to obtain very good estimates of the network operating points. In numerical experiments the algorithm is found to provide good approximations to the paths followed by successive approximation on the necessary conditions (Section 5.3.4).

An equivalent interpretation of the action of the algorithm for large numbers of iterations per coordinate is that it is approximating cyclic coordinate descent, that is, it is approximately solving for each coordinate the one-dimensional minimization

$$\theta_{l(k+1)}^* = \arg\min_{\theta_{l(k)}} P_\epsilon^{Team}(\theta_{1(k+1)}, \ldots, \theta_{l-1(k+1)}, \theta_{l(k)}, \theta_{l+1(k)}, \ldots, \theta_{N(k)}) \tag{5.43}$$

using multiple iterations of a gradient technique along each coordinate.

Recall that in Section 3.5 it was argued that, due to the nature of the dependence of the network parameters on one another, the parameters at a particular DM could always be updated in parallel; hence the timing diagram of Figure 3-39. The same applies here as well, so that Figure 5-5 may be modified accordingly.

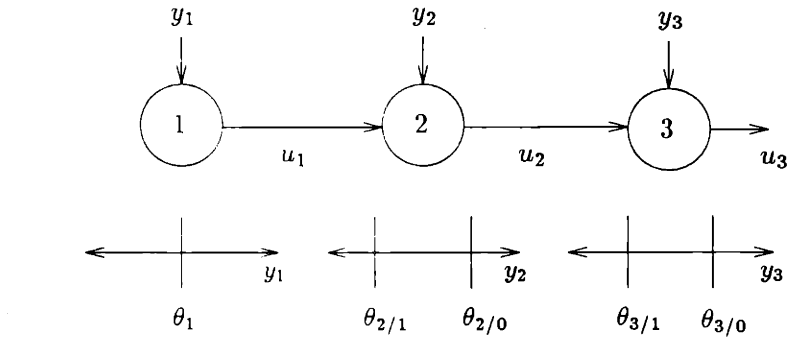
This algorithm still suffers from the large amount of communication it requires. Fortunately, the extensive communication required by each processor to compute its partial derivative may be neatly organized by exploiting the stagewise structure of the decentralized detection problem. In Section 3.2.3, the optimization of the decision rules was formulated as a deterministic optimal control problem. The result of this formulation was that the gradients were found to be computable using the adjoint method familiar from optimal control theory, which required that each node obtain only state and costate information from its immediate neighbors in order to compute its partial derivative. We now exploit this fact to show that the coupling costs may be computed with the aid of propagating state and costate equations, thus making the required communication strictly local. We will refer to this method as "back propagation" since it is directly analogous to the familiar back propagation training algorithm for perceptron neural networks [55], which itself can be shown to be equivalent to the adjoint method for a MSE terminal cost.

5.3.3 Back Propagation (WIN-BP) Implementation

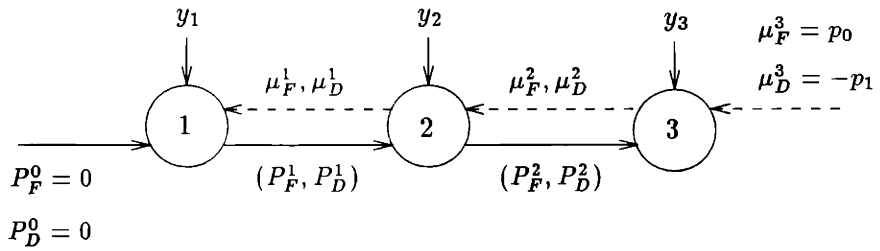
In this section we present a WIN-based nonparametric implementation of the adjoint method for computing gradients. We term this algorithm WIN-BP, for back-propagation with window evaluations. It is a scheme for structuring the coupling cost computations of the previous sections in order to minimize communication requirements. We establish that the λ_0, λ_1 cost coefficients required by each DM to update its threshold parameter(s) can be calculated using only state information from the DMs immediate predecessors and costate information from its immediate successors, where the state and costate are as defined as in Section 3.2.3. We consider only tandem topologies. Generalizations to trees follow in similar fashion from [62].

It is easiest to introduce the algorithm by means of an example. We illustrate using 3-Tand, which is depicted in Figure 5-6(a). Suppose we wish to optimize the threshold parameters of 3-Tand using a distributed implementation of the adjoint method discussed in Section 3.2.3.

In the distributed setting, the information and computation are parsed as shown



(a)



(b)

Figure 5-6: 3-Tand Tandem Topology (a) Threshold Parameters identified in terms of numerical notation. (b) Information requirements at each node to compute partial derivative.

	DM 1 (θ_1)	DM 2 ($\theta_{2/0}, \theta_{2/1}$)	DM 3 ($\theta_{3/0}, \theta_{3/1}$)
Obtains:	μ_F^1, μ_D^1	(P_F^1, P_D^1) μ_F^2, μ_D^2	(P_F^2, P_D^2)
Local Info:	(P_F^1, P_D^1)	$(P_F^{2/0}, P_D^{2/0})$ $(P_F^{2/1}, P_D^{2/1})$	p_0, p_1 $(P_F^{3/0}, P_D^{3/0})$ $(P_F^{3/1}, P_D^{3/1})$
Computes:	$P_F^1 = P_F^1$ $P_D^1 = P_D^1$ $\mu_D^1 = [P_D^{2/1} - P_D^{2/0}]\mu_D^2$	$P_F^2 = P_F^{2/0}(1-P_F^1) + P_F^{2/1} P_F^1$ $P_D^2 = P_D^{2/0}(1-P_D^1) + P_D^{2/1} P_D^1$ $\mu_F^1 = [P_F^{2/1} - P_F^{2/0}]\mu_F^2$ $\mu_D^2 = [P_D^{3/1} - P_D^{3/0}]\mu_D^3$; $\mu_D^3 = -p_1$	$P_F^3 = P_F^{3/0}(1-P_F^2) + P_F^{3/1} P_F^2$ $P_D^3 = P_D^{3/0}(1-P_D^2) + P_D^{3/1} P_D^2$ $\mu_F^2 = [P_F^{3/1} - P_F^{3/0}]\mu_F^3$; $\mu_F^3 = +p_0$
Partials:	$\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_1} =$ $-[\mu_F^1] p_{Y_1\|H_0}(\theta_1\|H_0)$ $+[-\mu_D^1] p_{Y_1\|H_1}(\theta_1\|H_1)$	$\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{2/0}} =$ $-[(1-P_F^1)\mu_F^2] p_{Y_2\|H_0}(\theta_{2/0}\|H_0)$ $+[-(1-P_D^1)\mu_D^2] p_{Y_2\|H_1}(\theta_{2/0}\|H_1)$ $\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{2/1}} =$ $-[P_F^1 \mu_F^2] p_{Y_2\|H_0}(\theta_{2/1}\|H_0)$ $+[-P_D^1 \mu_D^2] p_{Y_2\|H_1}(\theta_{2/1}\|H_1)$	$\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{3/0}} =$ $-[(1-P_F^2)\mu_F^3] p_{Y_2\|H_0}(\theta_{3/0}\|H_0)$ $+[-(1-P_D^2)\mu_D^3] p_{Y_2\|H_1}(\theta_{3/0}\|H_1)$ $\frac{\partial P_\epsilon^{3-Tand}}{\partial \theta_{3/1}} =$ $-[P_F^2 \mu_F^3] p_{Y_2\|H_0}(\theta_{3/1}\|H_0)$ $+[-P_D^2 \mu_D^3] p_{Y_2\|H_1}(\theta_{3/1}\|H_1)$

Table 5.1: Distribution of Information in Distributed Implementation of Optimal Control Formulation for 3-Tand

in Table 5.1.

The first line of the table specifies the information which must be obtained by each DM from the others. The communication of this information is depicted in in Figure 5-6(b). The second line specifies the information which must be estimated and maintained locally by each DM. The third line indicates the computations which must be performed by each DM, and the final line indicates the exact computation which must be performed by each DM in order to compute its partial derivative.

This scheme can be adapted for nonparametric optimization by propagating *estimated* states and costates, and then using locally running window algorithms to nonparametrically evaluate the partial derivatives. Since the window algorithms don't

require explicit modeling of the prior probabilities, the costate equation we desire is scaled by the priors. Equivalently, we can choose the alternative initial costate condition

$$\mu_F^3 = 1, \quad \mu_D^3 = -1 \tag{5.44}$$

The estimated costs for 3-Tand are given by

$$\hat{\lambda}_0^1 = \hat{\mu}_F^1, \quad \hat{\lambda}_1^1 = -\hat{\mu}_D^1 \tag{5.45}$$

$$\hat{\lambda}_0^{2/0} = (1 - \hat{P}_F^1)\hat{\mu}_F^2, \quad \hat{\lambda}_1^{2/0} = -(1 - \hat{P}_D^1)\hat{\mu}_D^2 \tag{5.46}$$

$$\hat{\lambda}_0^{2/1} = \hat{P}_F^1\hat{\mu}_F^2, \quad \hat{\lambda}_1^{2/1} = -\hat{P}_D^1\hat{\mu}_D^2 \tag{5.47}$$

$$\hat{\lambda}_0^{3/0} = (1 - \hat{P}_F^2)\mu_F^3, \quad \hat{\lambda}_1^{3/0} = -(1 - \hat{P}_D^2)\mu_D^3 \tag{5.48}$$

$$\hat{\lambda}_0^{3/1} = \hat{P}_F^2\mu_F^3, \quad \hat{\lambda}_1^{3/1} = -\hat{P}_D^2\mu_D^3 \tag{5.49}$$

The back propagation algorithm consists of the sequence of steps indicated in the flow diagram of Figure 5-7.

For a general tandem network the costs are computed as

$$\hat{\lambda}_0^1 = \hat{\mu}_F^1, \quad \hat{\lambda}_1^1 = -\hat{\mu}_D^1 \tag{5.50}$$

$$\hat{\lambda}_0^{i/0} = (1 - \hat{P}_F^{i-1})\hat{\mu}_F^i, \quad \hat{\lambda}_1^{i/0} = -(1 - \hat{P}_D^{i-1})\hat{\mu}_D^i \tag{5.51}$$

$$\hat{\lambda}_0^{i/1} = \hat{P}_F^{i-1}\hat{\mu}_F^i, \quad \hat{\lambda}_1^{i/1} = -\hat{P}_D^{i-1}\hat{\mu}_D^i \tag{5.52}$$

To prove convergence of the the WIN-BP algorithm, it has to be demonstrated that the conditions on the costs required for convergence still hold when the costs are calculated in this fashion. As it turns out, the desired properties are completely evident from this construction.

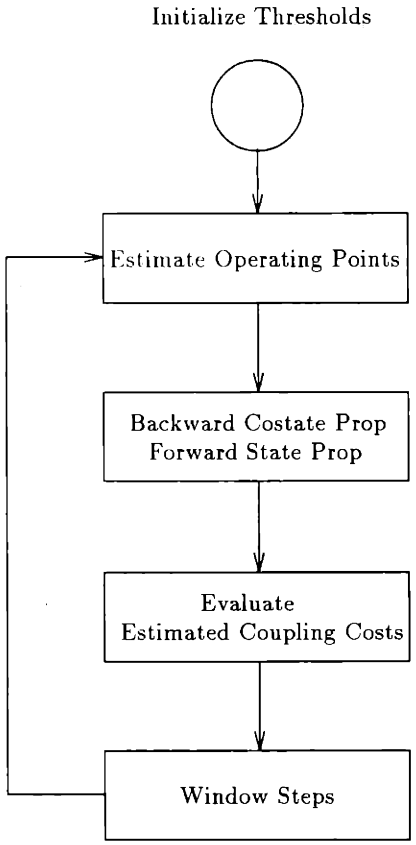


Figure 5-7: Flow chart depicting sequence of steps executed in back propagation implementation of WIN algorithm.

5.3.4 Numerical Experiments

For the team problem, a wide variety of numerical experiments can be performed by varying the prior probabilities, the variance of each DM, and the topology. In order to demonstrate all of the previously suggested algorithms, we limit the scope of the numerical experiments by considering two basic team Gaussian detection problems, one for 2-Tand and one for 3-Vee. The parameters of the underlying hypothesis test are held constant throughout in order to permit comparison. Our main purpose is to illustrate typical sample paths and the corresponding performance of each algorithm on some reasonable problem instances.

Example: 2-Tand

In this section, we consider the team Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3 \tag{5.53}$$

$$\sigma_A^2 = \sigma_B^2 = 1, \quad p_0 = 0.75 \tag{5.54}$$

The variances of the two DMs are equivalent, but there is significant prior bias in the data. The optimal[6] observation thresholds for this problem are $\alpha^* = 2.2209$, $\beta_0^* = 3.2521$, and $\beta_1^* = 1.5734$. The minimum probability of error is $P_\epsilon(\underline{\theta}^*) = .0794$.

All of the algorithms in this section are initialized to the threshold settings

$$\alpha_1 = 2.0, \quad \beta_{0(1)} = 2.5, \quad \beta_{1(1)} = 1.5 \tag{5.55}$$

for which the error probability is $P_\epsilon(\underline{\Theta}_1) = .1052$ As discussed in Chapter 4, it is not unreasonable to assume that the sample means of the data may be roughly determined, so that the algorithm may be initialized between the means, in this case at 2.0. The initial values of the thresholds $\beta_{0(1)}$ and $\beta_{1(1)}$ were obtained by small perturbations in the correct directions. It turns out that the algorithms are not too

[6]Computed previously in Chapter 3 using fixed point iteration, and checked by a variety of other numerical techniques.

sensitive to the initial parameter locations, so long as they are in a region of significant probability density.

WIN: In this section, we examine the performance of the WIN algorithm, with Jacobi-type iterations, for conditional probabilities estimated based on 10,000 trials, 50 trials, and 0-1 estimates. We use the normalized window variant, with a rectangular window, and the stepsize and window-width sequences

$$\rho_k = \frac{1}{\sqrt{k}}, \quad \delta_k = \frac{2.25}{\sqrt{k}} \tag{5.56}$$

to update all parameters. In chapter 6, we establish that this is a valid combination of step and window-width sequences. The gain parameters $\rho_1 = 1$ and $\delta_1 = 2.25$ were chosen based on a heuristic technique of Wassel and Sklansky [73].

In Figure 5-8, the paths are shown for the case of length 10,000 estimation phases. That is, each component conditional probability (operating point) is estimated based on 10,000 samples. For all intents and purposes, the operating points are computed exactly for this case; the costs are very accurately modeled. The sample paths are run out to 2000 iterations, meaning that 2000 *updates* of each parameter were performed. Estimation phases are not indicated, but including estimation phase measurements each sample path requires 20,000,000 measurements. The average sample path is computed by averaging 4 independent sample paths. We will see shortly that the number of estimation trials may be vastly curtailed without significantly impacting the average performance of the algorithm.

Notice that since both DMs have the same variance, approximately the same variance is evident in the individual sample paths.

The probability of error for several of the sample paths and the average sample path are shown in Figure 5-9. Notice that even the unaveraged paths have dipped below an error probability of .085 within 50 to 100 iterations.

We now examine the effect using estimated costs based on cruder computation of the operating points. Figure 5-10 illustrates some typical sample paths using estimation phases of 50 iterations. In comparison with the paths based on very

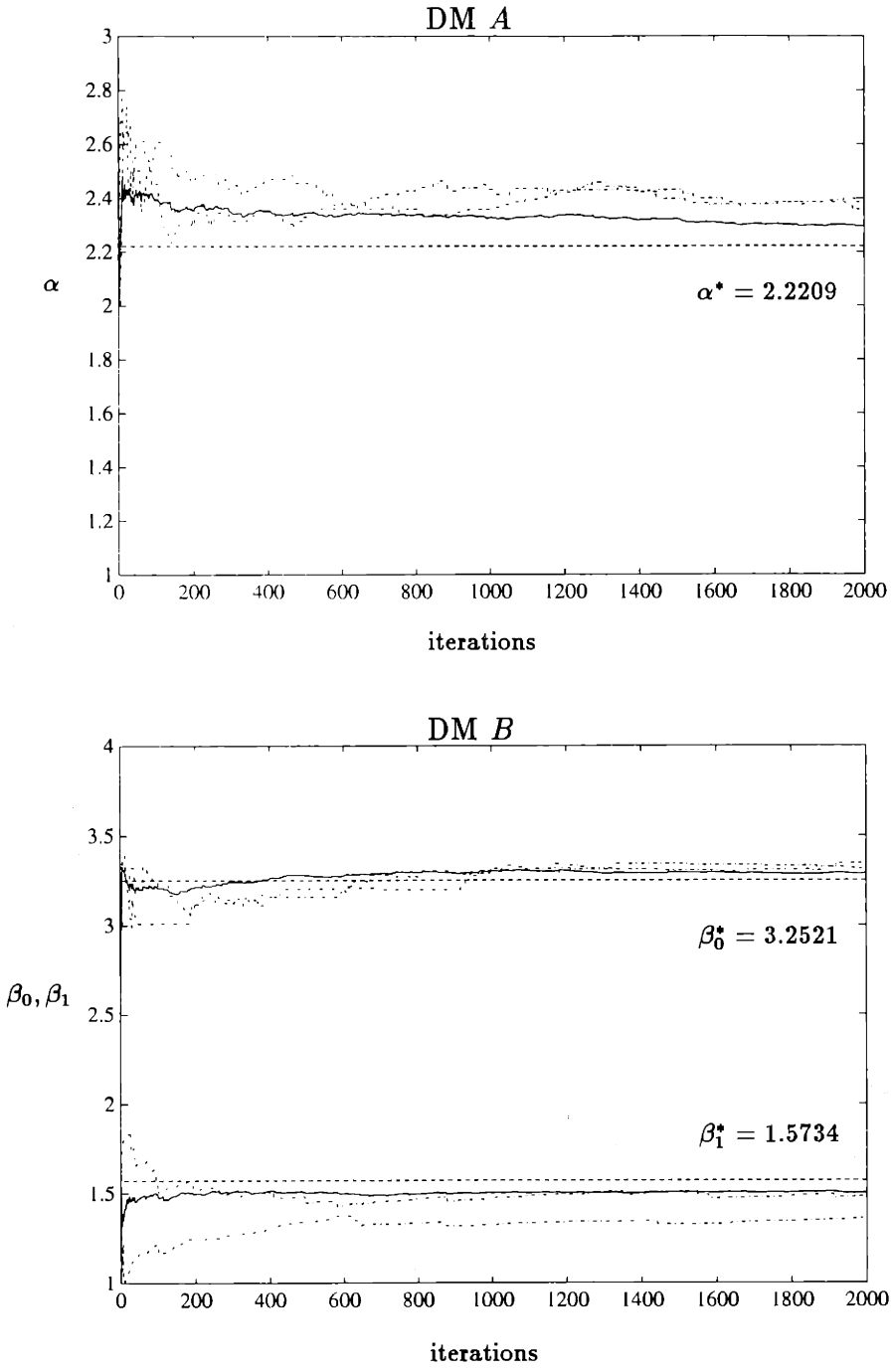


Figure 5-8: WIN, Jacobi iteration (10,000 estimation iterations) : Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 independent sample paths (solid), typical sample paths (dotted and dashed), and optimal threshold values (dashed). All sample paths initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

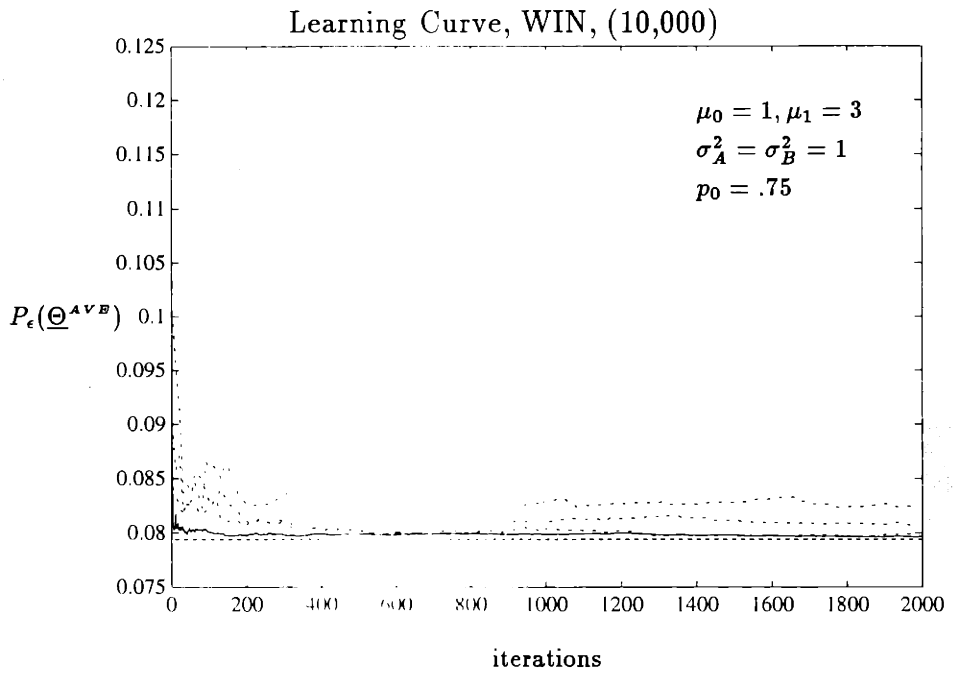


Figure 5-9: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$ (solid) and sample paths of the cost over the other runs (dotted and dashed). Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ (dashed). The initial error probability for all paths is is $P_\epsilon(\underline{\Theta}_1) = .1052$.

precise cost estimates, the individual sample paths display higher variance. This is most evident when comparing the sample paths for α between the two algorithms. However, the averages over 4 sample paths are very similar. Each sample path consists of 2000 updates, with 50 estimation trials corresponding to each update, so that each sample path is computed based on 100,000 measurements.

The corresponding average performance over 4 independent sample paths is quite good, although the error sample path exhibits slightly more error in the transient phase, and appears to be squeezing out the last bit of performance from 0.08 to 0.0794 extremely slowly.

Carrying this to the extreme, we consider sample paths computed based on 0-1 estimates of the component conditional probabilities. In Figure 5-12, even higher variance in the individual sample paths is evident; again this is most clear for the paths for α. It is difficult to compute exactly how many measurements are required by this algorithm, since the 0-1 conditional probability estimates require that at least

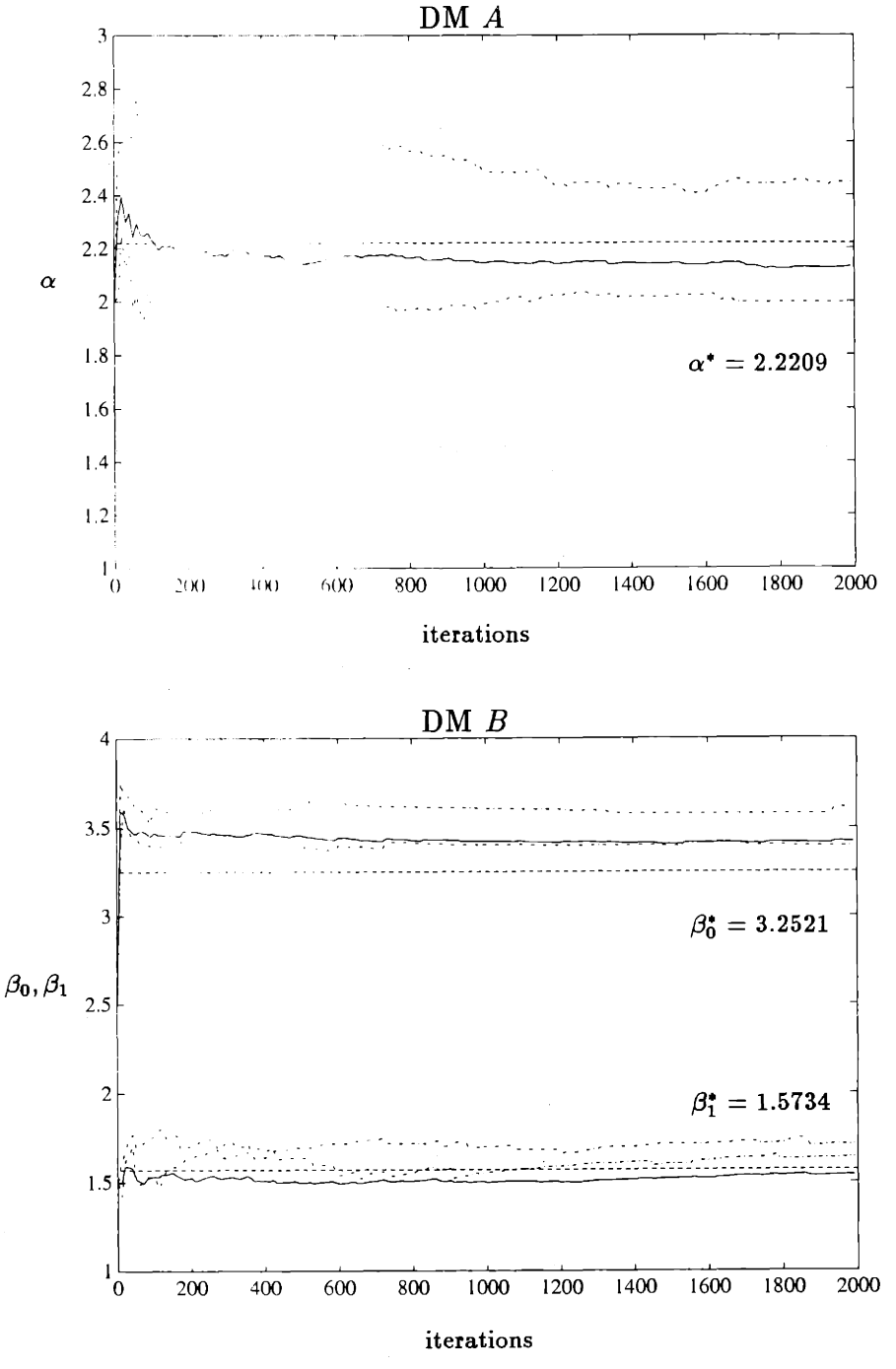


Figure 5-10: WIN, Jacobi iteration (50 estimation iterations) : Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). All sample paths initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

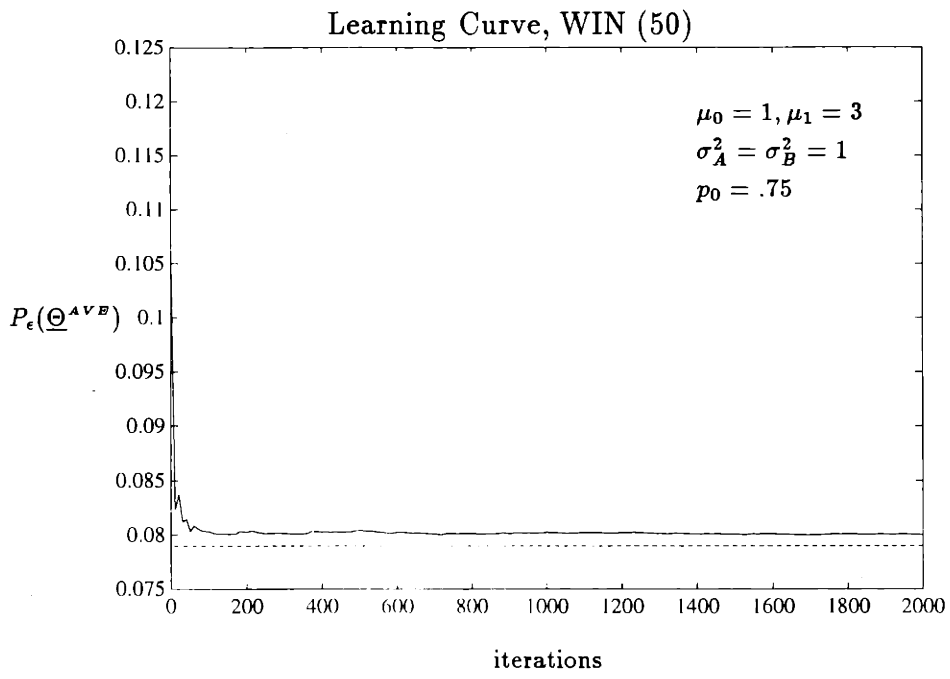


Figure 5-11: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ shown dashed. The initial error probability for all paths is is $P_\epsilon(\underline{\Theta}_1) = .1052$.

one case each of H_0 and H_1 be observed to compute the estimates. The number of trials required to obtain at least one realization of each hypothesis is a function of the prior probabilities. For this case, $p_0 = 0.75$ and $p_1 = 0.25$, so that after 11 estimation trials, the probability is 99.99% that an H_0 case has been observed and 95.77% that an H_1 case has been observed. Thus a very conservative estimate would be that 20,000 measurements are required.

The corresponding performance is shown in Figure 5-13. It is plainly visible that the error corresponding to the individual sample paths is higher, particularly in the transient portion of the curves. However, stochastic descent of the error is clearly evident, and reasonably good performance is achieved within 500 iterations. The performance of the average sample path is still quite good, although it displays slightly higher transient error, and again seems to be requiring many iterations to squeeze out the last bit of performance.

These experiments indicate that increased effort in modeling the costs results in lower variance of the individual sample paths, and more rapid convergence to the

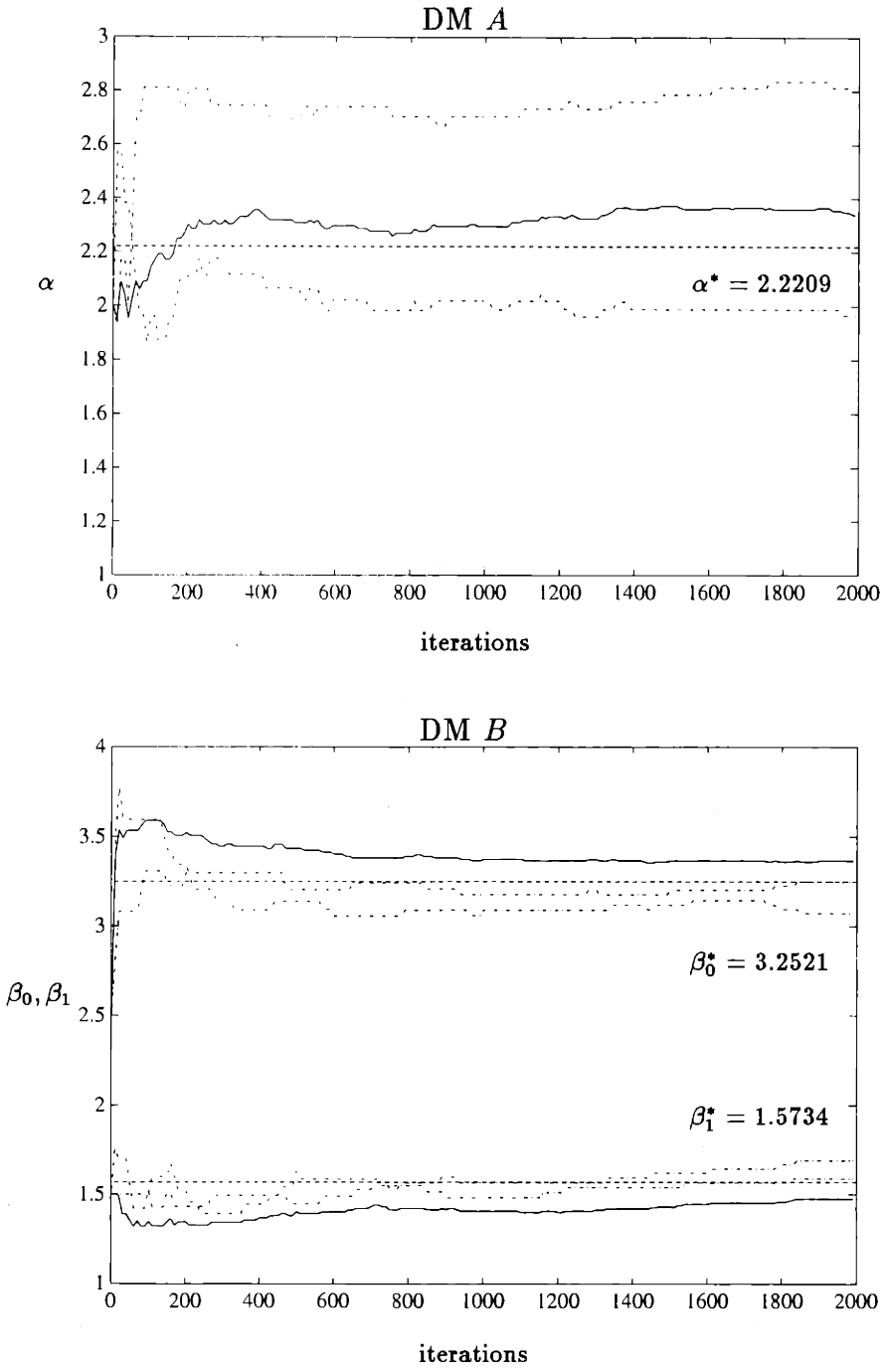


Figure 5-12: WIN, Jacobi (0-1 estimates): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). All sample paths initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

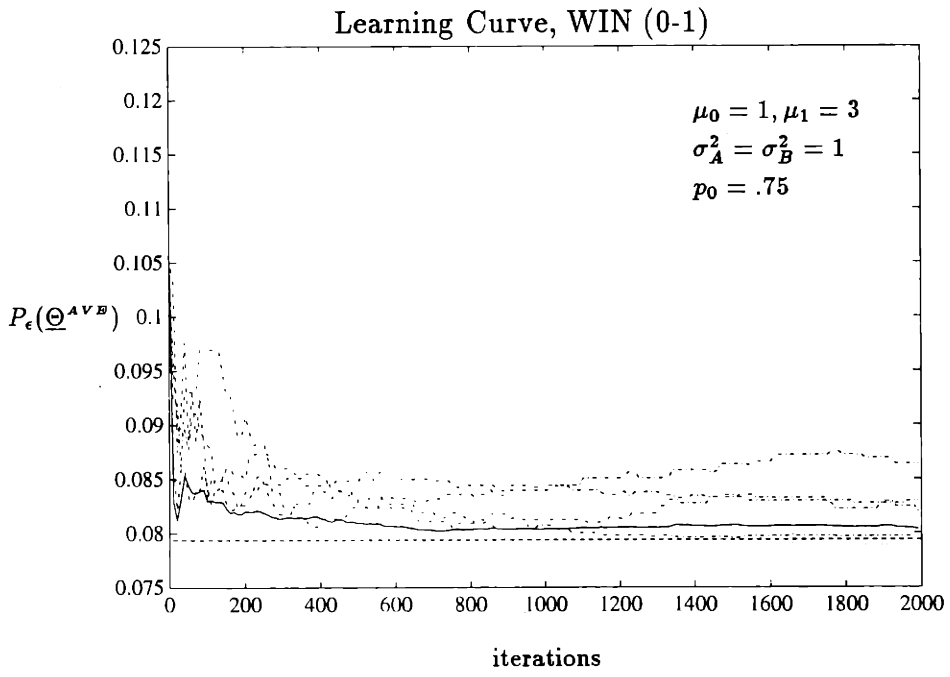


Figure 5-13: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$ (solid) and sample paths of the cost corresponding to each sample path (dotted-and-dashed). Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ (dashed). The initial error probability for all paths is is $P_\epsilon(\underline{\Theta}_1) = .1052$.

optimum value, although on the average, performance is quite good with crudely modeled costs.

WIN-GS: A Gauss-Seidel implementation of the previous algorithm was also suggested. In this technique, multiple updates of each component are performed, followed by an estimation phase. We use the same window algorithm, employing a normalized rectangular window, with the same update and window-width sequences

$$\rho_k = \frac{1}{\sqrt{k}}, \quad \delta_k = \frac{2.25}{\sqrt{k}} \tag{5.57}$$

As indicated in the timing diagram of Figure 5-5, the first action required by the algorithm is to obtain estimates of the operating points corresponding to the initializations $\beta_{0(1)}$ and $\beta_{1(1)}$. We use the initializations $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$ and $\beta_{1(1)} = 1.5$.

The algorithm of this section employs 700 consecutive updates of each parameter, followed by estimation phases of 1000 trials. A total of 15 complete update cycles

of the parameters are performed, leading to a total of 700(15) = 10,500 iterations per parameter sample path. Figure 5-14 shows average sample paths, in which each subproblem is averaged over 15 independent sample paths as the algorithm advances. Each of these paths therefore corresponds to 10,500(1000)(15) = 157,000,000 measurements. Note that the resulting convergence is steady and almost linear in appearance, but extremely slow.

The slow progress of this algorithm is evident in Figure 5-15, which illustrates the corresponding performance of the average sample path. Although descent is clearly visible, the error probability has still not dipped below 0.85, even after 15 update cycles have been completed.

The observed performance of this algorithm deteriorated notably when cruder estimates of the costs were used, since in this case, many steps in an incorrect direction could be taken, depleting the stepsize in the process, so that the algorithm had a difficult time recovering[7].

This observation motivates the use of an ad hoc technique in which the stepsizes used by the algorithm are restarted at the beginning of each subproblem. This algorithm was first suggested by Wissinger and Athans in [77]. By restarting the stepsize of each subproblem, consecutive subproblems for a particular parameter are decoupled. A new subproblem is re-solved on each cycle. In fact, it is not necessary, or even desirable, to initialize each subproblem to the endpoint of the previous one. It is only necessary that each subproblem be initialized in a region of high probability.

If the overhead is paid to obtain good estimates of the network operating points, then sample paths which closely approximate the exact Gauss-Seidel component fixed point iterations of Chapter 3 are obtained. Figure 5-16 illustrates the average sample paths of the thresholds obtained using 10 subproblems of 700 iterations each, with estimation trials of length 500 following each subproblem. The dotted paths are the exact successive approximation solution. Again, each subproblem is averaged 15 times before the algorithm advances. We initialize the algorithm with the values $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$ and $\beta_{1(1)} = 1.5$. Subsequently, the subproblems for each threshold are

[7]Although theoretically, asymptotic convergence of the algorithm is assured.

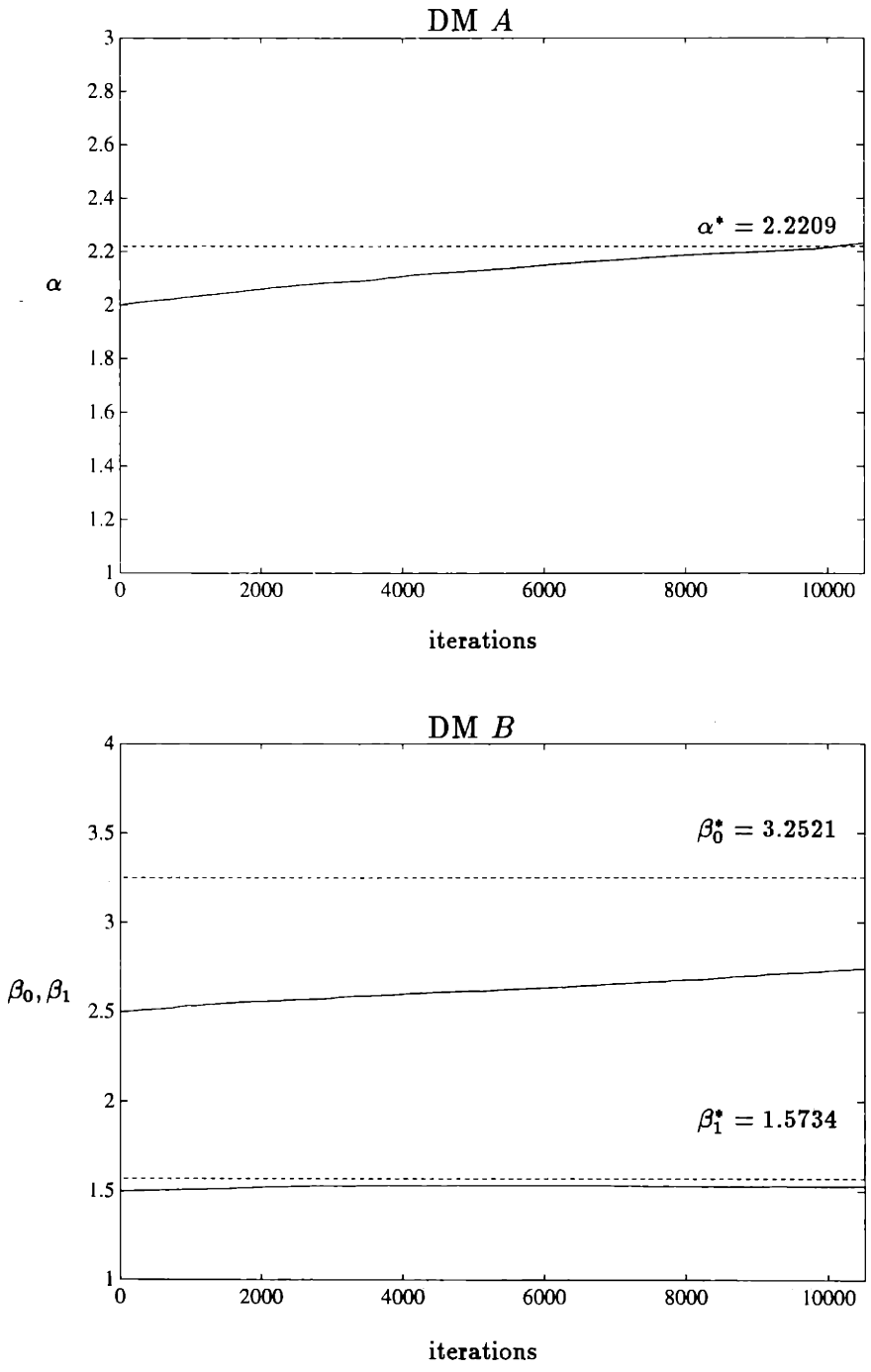


Figure 5-14: WIN-GS (1000 estimation trials): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 15 paths (solid), and optimal threshold values (dashed). All sample paths initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

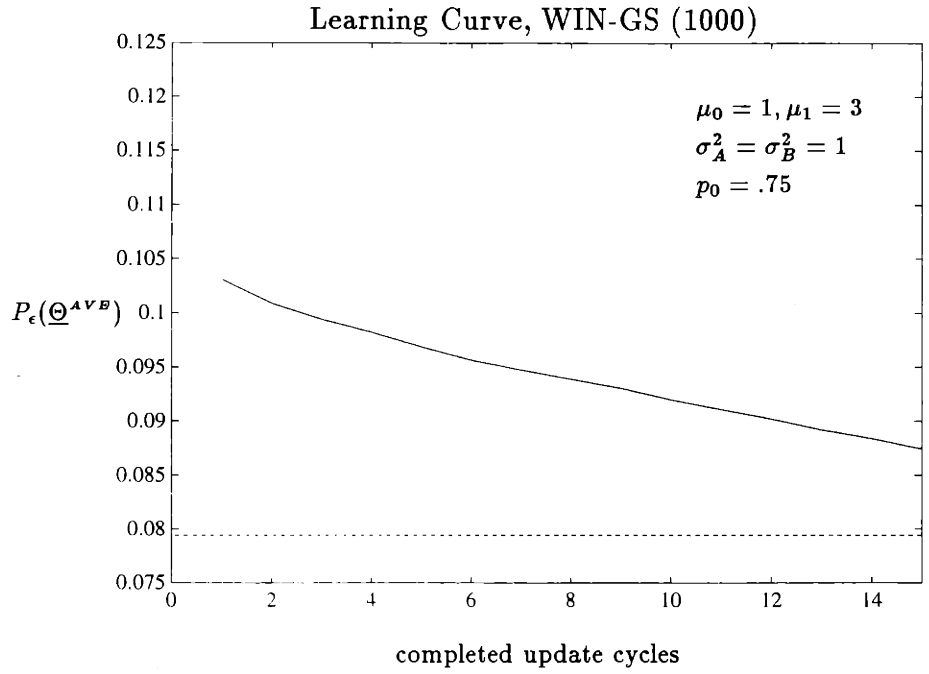


Figure 5-15: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVB})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = .0794$ shown dotted. The initial error probability is $P_\epsilon(\underline{\Theta}_1) = .1052$.

initialized to the value 2.0. The large initial steps give rise to the observed spikes.

If the subproblems were solved exactly, the sample paths would converge to the corresponding point on the exact successive approximation curve. Since the subproblems are truncated, this is not observed. However, it is generally true that the sample paths are moving toward the exact path when they are off of it.

The total amount of data required to generate a single average path in this example is (700)(15)(10) + (500)(10) = 110,000 measurements.

The performance of the algorithm is quite good, as evidenced by Figure 5-17, which illustrates the cost at the end of each completed update cycle. After a single update cycle, the cost is well below 0.085, and in fact never exceeds .082.

The total number of measurements required by the algorithm is a function of several things. The number of trials required to solve a given subproblem depends on the noisiness (variance) of the associated DM's observations, while the number of subproblems which must be solved is highly dependent on the degree of coupling between the DMs, the initial starting point, and the size of the network.

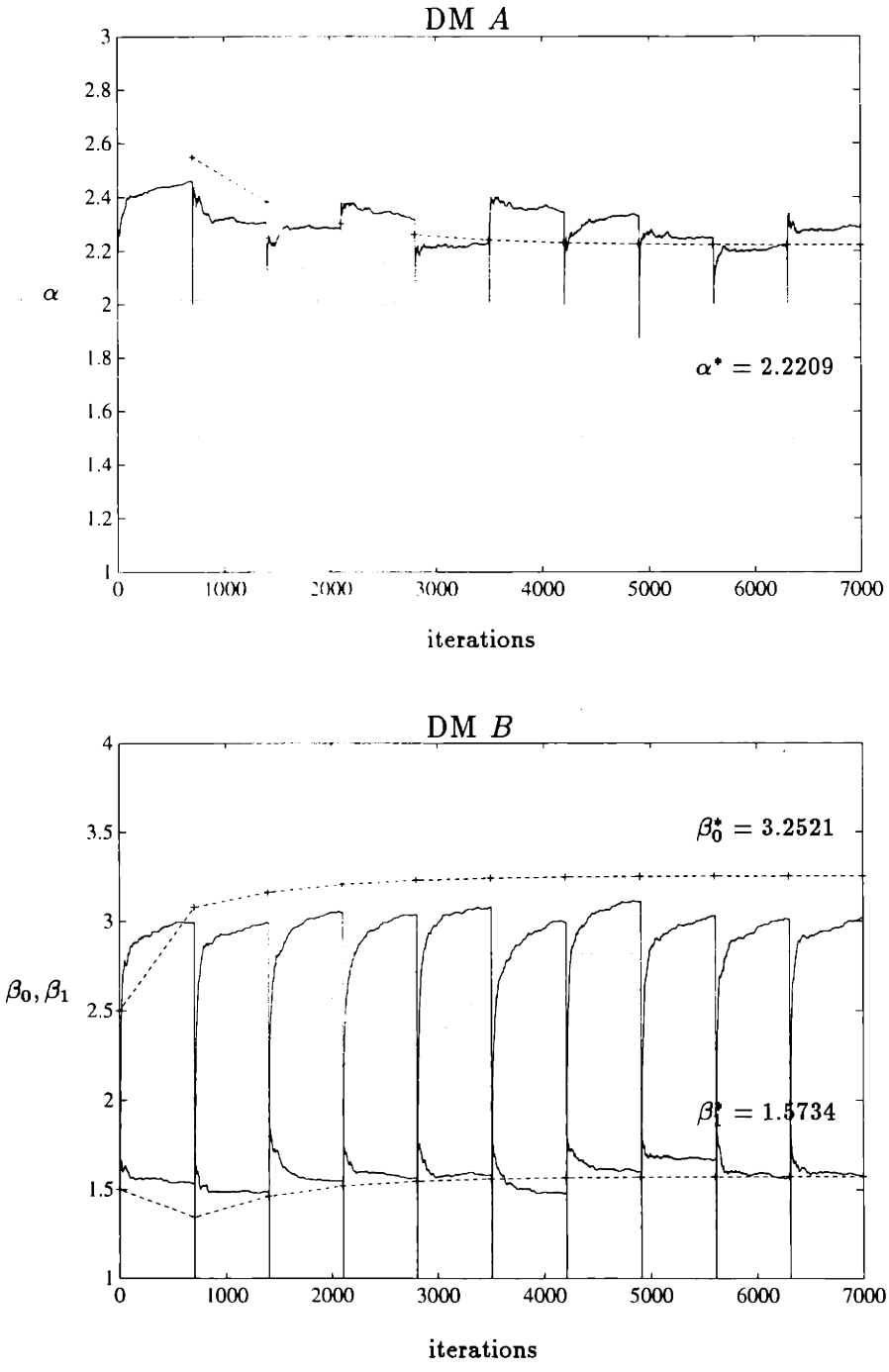


Figure 5-16: WIN-GS, ad hoc (500 estimation trials): Average sample path of $\{\underline{\Theta}_k\}$, each subproblem averaged over 15 paths as the algorithm advances (solid). The exact Gauss-Seidel successive approximation solution is shown (dashed). Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

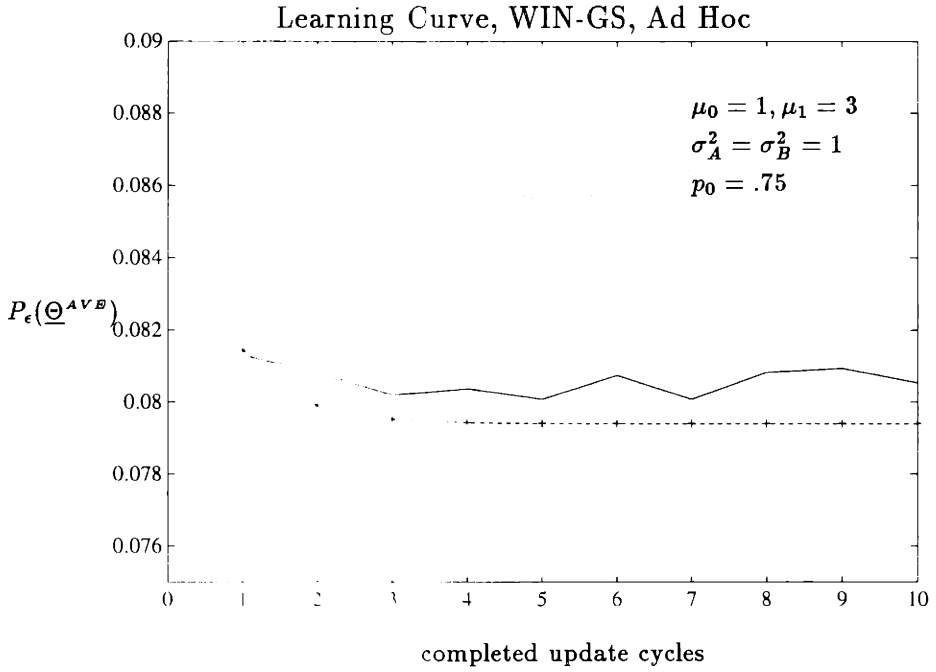


Figure 5-17: Samples of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$ plotted at the end of each update cycle (solid). Path corresponding to exact successive approximation shown (dashed).

Since the WIN-BP algorithm is simply a communication scheme for the WIN algorithm, and exhibits identical behavior, separate numerical experiments for it are not required.

Example: 3-Vee

In this section, we repeat several of the experiments of the previous section for the topology 3-Vee. We consider the team Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3 \tag{5.58}$$

$$\sigma_A^2 = 1.5, \quad \sigma_B^2 = 0.5, \quad \sigma_C^2 = 1.0, \quad p_0 = 0.75 \tag{5.59}$$

where now we have incorporated different variances at each DM, in addition to prior bias. The optimal thresholds for this problem are $\alpha^* = 2.0457$, $\beta^* = 2.1287$, $\xi_{00}^* = 4.2818$, $\xi_{01}^* = 1.8110$, $\xi_{10}^* = 2.9391$, and $\xi_{11}^* = 0.4683$. The optimal probability of error is $P_\epsilon(\underline{\theta}^*) = 0.0388$.

The same comments regarding initialization of the algorithm apply. We will choose the initialization $\alpha_1 = 2.0$, $\beta_1 = 2.0$, $\xi_{00} = 3.0$, $\xi_{01} = 2.0$, $\xi_{10} = 1.0$, and $\xi_{11} = 0.0$. Notice that the initializations of ξ_{01} and ξ_{10} require that these thresholds uncross to achieve their optimal values. The probability of error corresponding to this initial setting of the thresholds is $P_\epsilon(\underline{\theta}_1) = 0.1119$.

WIN: In this section we examine the behavior of the WIN algorithm with Jacobi iterations. We continue to use the normalized window algorithm, with rectangular windows, and the stepsize sequences $\rho_k = 1/\sqrt{k}$ and $\delta_k = 2.25/\sqrt{k}$.

Figures 5-18 and 5-19 illustrate some typical threshold sample paths. Each sample path was run out to 10,000 iterations (updates), with each update followed by an estimation phase of length 1000. Average paths were computed by averaging 3 independent sample paths. Thus, each sample path represents 3(10,000)(1000) = 30,000,000 measurements.

The effect of the increased variance at DM A is evident in the large variance of the sample paths. Due to the reduced sensitivity of the cost with respect to the threshold at DM A, the convergence at A is extremely slow. The lower variance and higher sensitivity at DM B, results in much tighter sample paths which rapidly converge to the vicinity of the optimal threshold. Similar behavior is observed for the thresholds of DM C in Figure 5-19. The typical sample paths for ξ_{00} and ξ_{11} very tightly follow the average paths, and are hardly visible. Typical sample paths for ξ_{01} and ξ_{10} are not shown, so as not to obscure the thresholds uncrossing. The corresponding probability of error indicates that the algorithm has rapidly reduced the cost from 0.1119 to below 0.05. Even though the algorithm is continually reducing the cost, it appears that convergence in the tail is very slow.

WIN-GS: The ad-hoc Gauss-Seidel technique, with restarted stepsizes, also performed well for 3-Vee, as indicated by the curves in Figures 5-20 and 5-21. The sample paths for each subproblem were computed based on 700 iterations, were averaged 15 times, and a total of 15 update cycles were performed. Estimates were computed based on 500 iterations. The algorithm was initialized with $\beta = 2.0$,

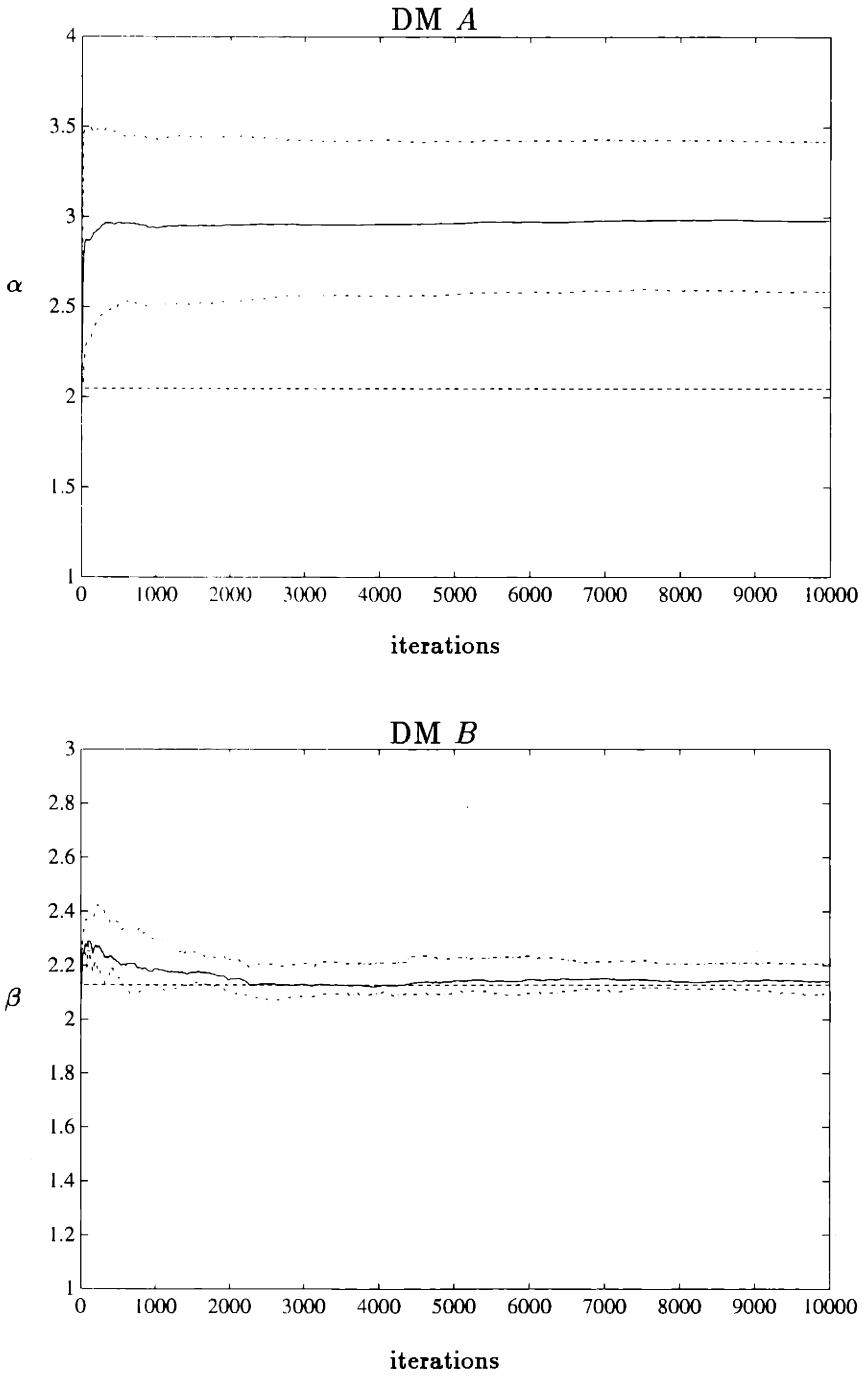
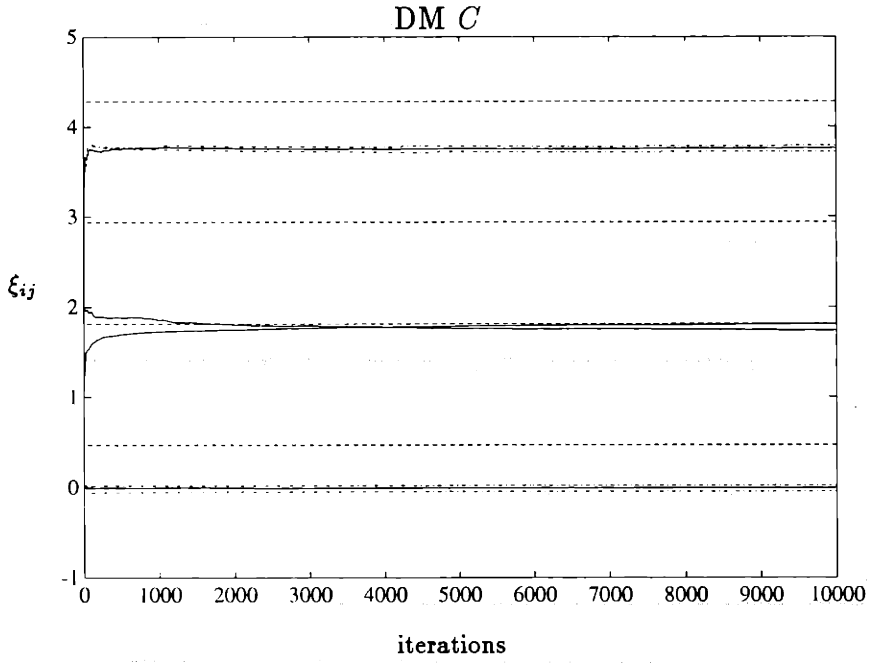


Figure 5-18: 3-Vee, WIN, (1000 estimation): Average sample path over 3 averages (solid), typical sample paths (dotted and dashed), and optimal threshold value (dashed). Gaussian case; $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = 1.5$, $\sigma_B^2 = 0.5$, $\sigma_C^2 = 1.0$, $p_0 = 0.75$.



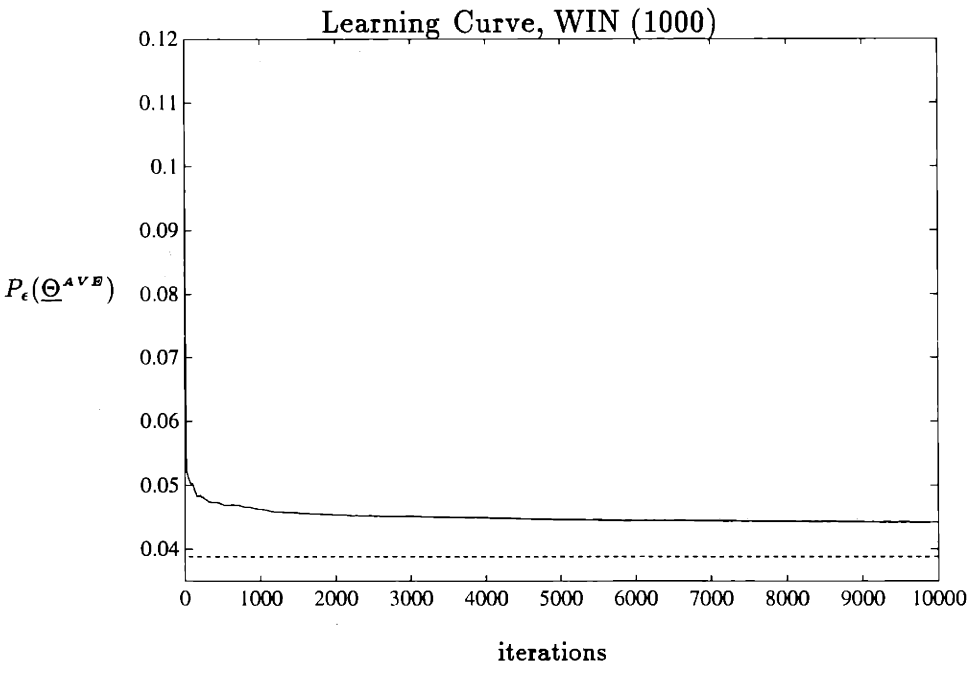


Figure 5-19: 3-Vee, WIN (cont'd): Lower curve; $P_\epsilon(\underline{\Theta}_k^{AVE})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) =$ 0.0388 shown dotted. The initial error probability is $P_\epsilon(\underline{\Theta}_1) = .1119$.

$\xi_{00} = 3$, $\xi_{01} = 2.0$, $\xi_{10} = 1.0$ and $\xi_{00} = 0.0$. Subsequent subproblems were always initialized to 2.0. The dashed curves indicate the exact fixed point iterations for this system as computed in Chapter 3.

It is perhaps more clearly evident in these figures than those displayed for 2-Tand that the subproblems are approximately converging to the corresponding points on the exact successive approximation curves. Note that thresholds ξ_{01} and ξ_{10} uncross almost immediately.

The probability of error for the algorithm shows excellent performance, with the cost after one completed update cycle already reduced from 0.1119 to less than 0.044.

In this section we have merely indicated the type of behavior one might expect to obtain from distributed window algorithms, at least for normalized rectangular windows on Gaussian detection problems. Extensive numerical study is warranted to determine good choices of stepsize and window sequences and gain parameters, as well as to better identify the tradeoffs involved between the number of trials that should be devoted to estimation versus updates.

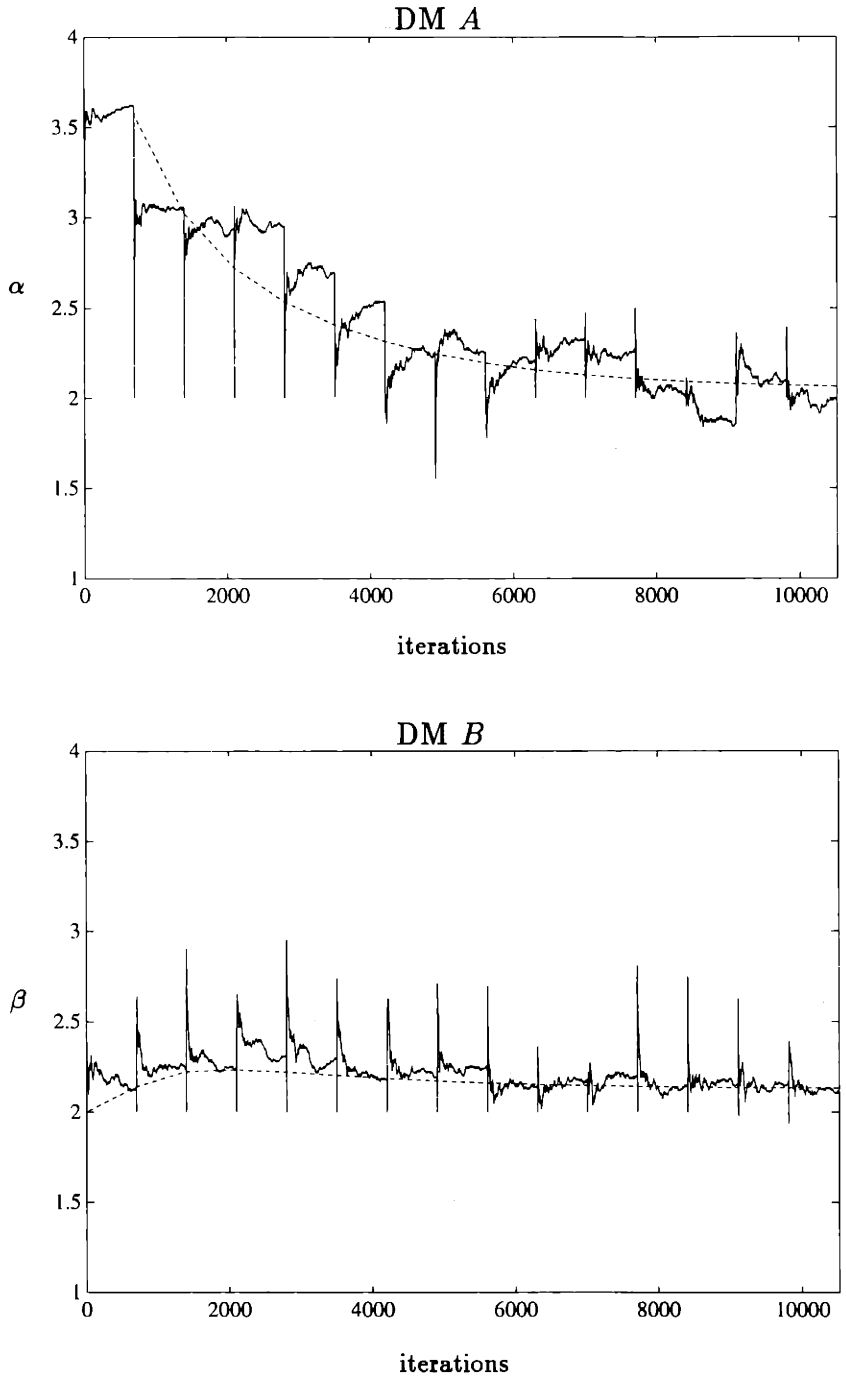


Figure 5-20: 3-Vee, WIN-GS ad hoc, (500 estimation): Averaged sample paths (solid) and the exact successive approximation solution (dashed). $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = 1.5$, $\sigma_B^2 = 0.5$, $\sigma_C^2 = 1.0$, $p_0 = 0.75$.

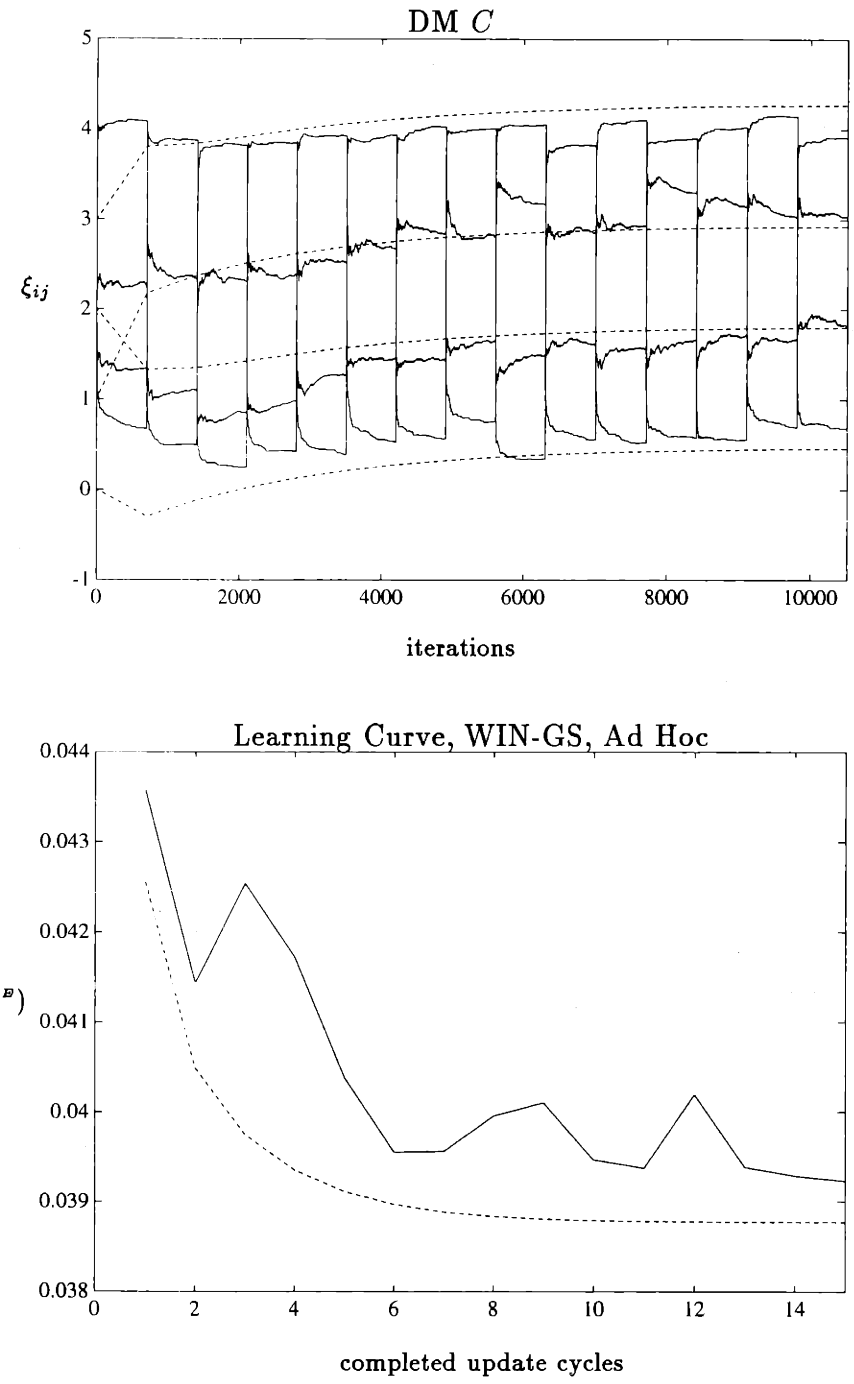


Figure 5-21: 3-Vee, WIN-GS ad hoc(cont'd): Lower curve; Samples of $\{P_\epsilon(\underline{\Theta}_k^{AVB})\}$ plotted at the end of each update cycle (solid). Path corresponding to exact successive approximation shown (dashed).

5.4 KW-Type Training Algorithms

As discussed in the chapter introduction, the KW-type algorithms we investigate for the network training problem are model-free, in the sense that no DM maintains an internal representation of the network. Thus, there is no communication aspect to these algorithms. Updates are computed by sampling the error surface directly, and require no derivative information. The price for this is that these algorithms must utilize global feedback from the network output rather than simply local feedback. In particular, we must make the following additional assumption in the KW setting.

Assumption 5.3 (Observability of the Output)
The team decision (decision by the primary DM) is available to every DM at each time k, i.e., available to DM i at time k is a realization of the information set $\{Y_{i(k)}, H^k, U_{Team(k)}\}$.

This assumption ensures that feedback from the network output is available to every DM (processor). This is required so that each processor may assess the impact of perturbations of its parameter on the team P_ϵ, measured at the output of the network. In terms of the notation previously established in Section 4.4, each DM must be able to obtain a realization of a random variable Q^{Team} which satisfies

$$E_{\underline{X}}\{Q^{Team}(\underline{X},\underline{\Theta})|\underline{\Theta}=\underline{\theta}\} = P_\epsilon^{Team}(\underline{\theta}),\ \ \forall \underline{\theta} \in \Re^N \tag{5.60}$$

where $\underline{X}$ denotes a network measurement. The indicator function of a team error is given by

$$Q^{Team}(\underline{X},\underline{\Theta}) = \begin{cases} 1 & \text{if } U_{Team}=1 \text{ and } H=H_0 \\ 1 & \text{if } U_{Team}=0 \text{ and } H=H_1 \\ 0 & \text{else} \end{cases} \tag{5.61}$$

satisfies equation (5.60).

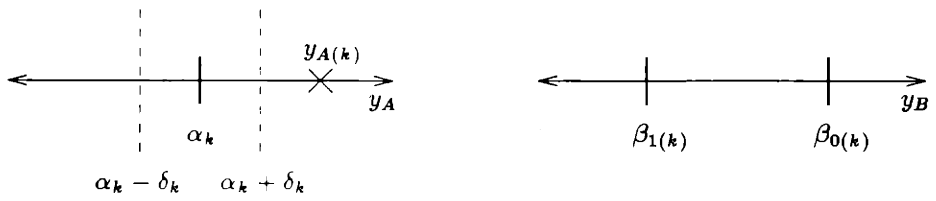


Figure 5-22: No Observability of β_0.

Of course, each realization of Q_k^{Team} depends on the entire network measurement vector $\underline{X}_k$ and parameter vector $\underline{\Theta}_k$, and in the distributed setting these are unavailable to each processor. However, knowledge of the team decision $U_{Team(k)}$ is sufficient for a processor l to obtain a realization of Q_k^{Team} since the acting hypothesis is known to it through its local measurement vector $X_{l(k)}$. It is useful to think of U_{Team} as an additional measurement received by each processor, which, in contrast to the external measurement X, is generated internal to the network. This is analogous to the situation for the coupling costs in the WIN-Type Algorithms. In the same way that $\lambda_{0(k)}^{il}$ and $\lambda_{1(k)}^{il}$ captured the dependence of the local update on $\underline{X}_k$ and $\underline{\Theta}_k$, so too does $U_{Team(k)}$. This is intuitive since updates of a component cannot be made in the absence of information concerning the current values of the other components. So, from a local point of view $Q_k^{Team}(\underline{X}_k, \underline{\Theta}_k)$ may be equivalently viewed as a function with dependence $Q_k^{Team}(X_{l(k)}, U_{Team(k)}, \Theta_{l(k)})$.

An entire team decision process must be executed in order to sample the team error surface. An apparent complication which arises in conjunction with this sampling process is that only a subset of the total parameter set is active in the production of any particular sample. Indeed, the set of parameters which is active on each trial is random, and depends directly on the network measurement vector $\underline{X}_k$ as well as the value of the current iterate $\underline{\Theta}_k$. For example, consider the situation depicted in Figure 5-22 for 2-Tand, for which the relative location of $y_{A(k)}$ to α has rendered parameter β_0 unobservable at the output.

It might appear that this property could interfere with convergence of the algorithm, since on any given trial some subset of the total parameter set is unobservable at the output by virtue of having no impact on the decision process. However, all we

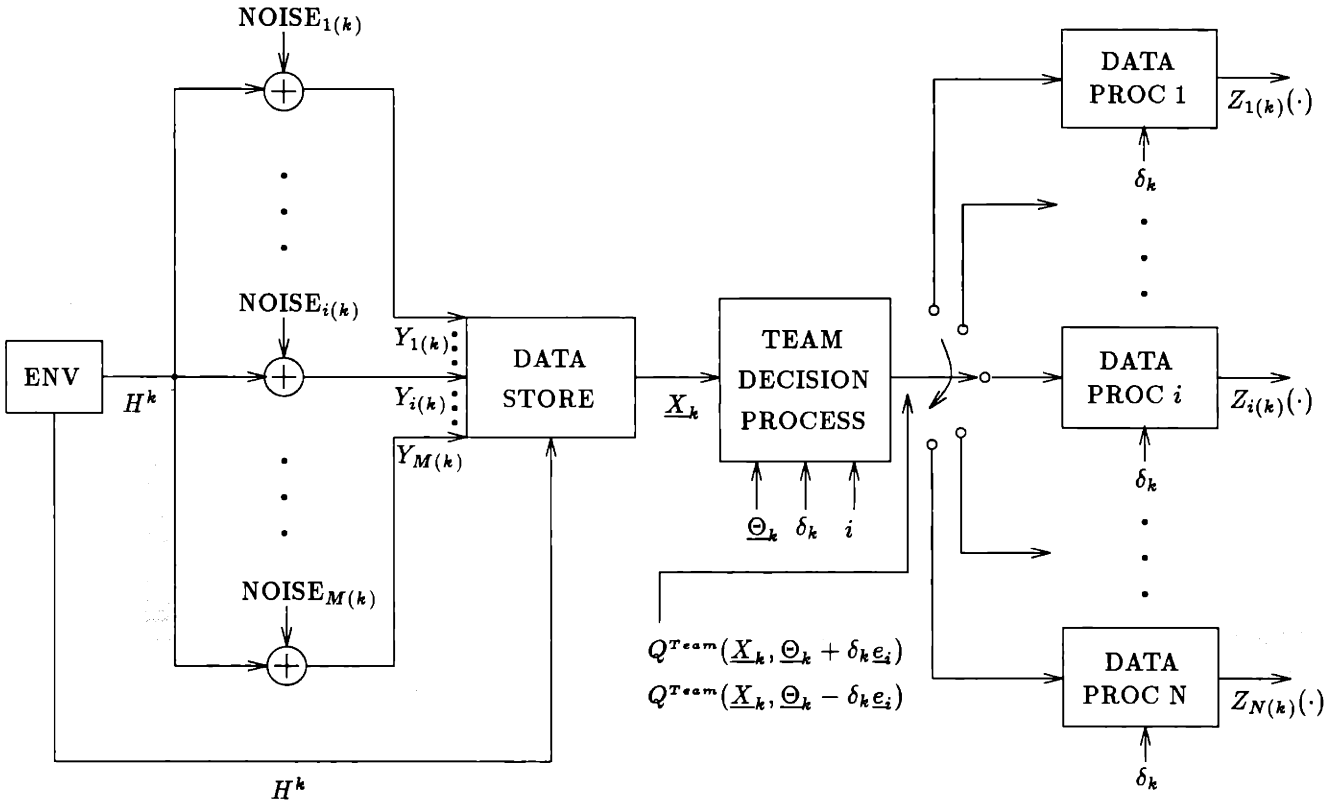


Figure 5-23: Data Processing, Team KW Setting

require to demonstrate convergence is that Q obeys (5.60). If so, then on the average the effect of each threshold parameter is observable.

The overall network data processing required for KW-Type algorithms is shown in Figure 5-23. Contrast the information fan-in through the Team Decision Process block with the parallel nature of the data processing that was depicted in Figure 5-4. This is the price paid the KW methods; a sequence of team decision processes must be executed for every network update.

5.4.1 Multivariable KW (KW)

The multivariable KW algorithm is the natural extension of the single parameter algorithm. The gradient is estimated by constructing estimates of each partial derivative using separate finite-difference approximations along each coordinate. Either one-sided or two-sided finite differences may be used.

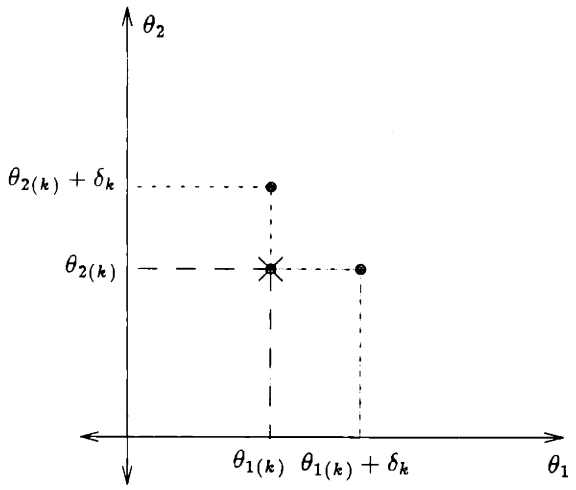


Figure 5-24: Sampling Pattern of One-Sided KW method for the case $N = 2$. The $\times$ indicates the location of the current iterate $\underline{\theta}_k$. The dots indicate locations of samples. Since the one-sided technique uses a single sample at the current iterate, a total of $N + 1 = 2 + 1 = 3$ samples are required for each update of $\underline{\theta}_k$ in this example.

One-Sided Variant

When each partial derivative is estimated using the one-sided variant, the function is sampled in the pattern illustrated in Figure 5-24, which illustrates parameter space for the case $N = 2$.

The relative timing of the events at each processor is shown in the timing diagram of Figure 5-25. Each update cycle is initiated by a simultaneous sampling of the cost corresponding to no perturbation of any component of the parameter vector. That is, a decision process is executed by the team, with all threshold parameters held fixed at their current values, and the resulting decision $U_{Team(k)}$ is observed by each processor, so that local realizations of $Q_k^{Team}(\underline{X}_k, \underline{\Theta}_k)$ may be computed by each. Then, one-by-one, the processors cyclically perturb their thresholds up by an amount δ_k and obtain a second sample, where each perturbation up is followed by the execution of a decision process by the team, and the resulting U_{Team} fed back to the updating processor. It is necessary that the parameters be perturbed one at a time, so that the effect of each perturbation may be independently observed at the output. Note that for an N-dimensional parameter vector, this scheme requires that $N + 1$

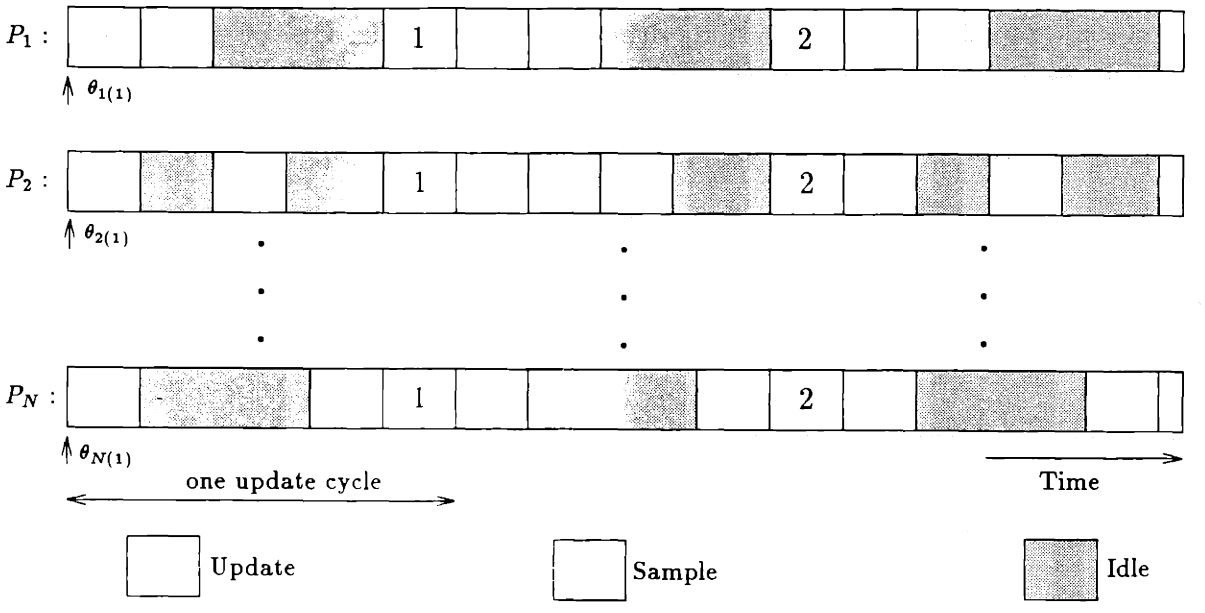


Figure 5-25: Timing diagram for multivariable KW with one-sided approximations of each partial derivative. Each sample block indicates that a realization of the cost function is observed by the processor.

different samples of P_ϵ^{Team} be taken, each of which requires a team decision process. We should point out that, as discussed in Chapter 4, it is not critical that each sample be taken with a different network measurement, although this will certainly work. It is satisfactory for each processor l corresponding to DM i to simply store its measurement $X_{l(k)} = \{Y_{i(k)}, H^k\}$ at each time k, and use it throughout the entire update cycle.

In order to make the previous discussion more precise, we need to address some notational issues. First, with regard to the index variable k, we again use it to index updates rather than measurements. If we adopt the implementation in which a single measurement $\underline{X}_k$ is used throughout the entire kth update cycle, then there is no difficulty. If each sample uses a different measurement, then the indexing of measurements and updates would no longer correspond. A second point of notation is that we subscript realizations of Q with the processor at which they are used. For example, if at time k processor l perturbs threshold $\theta_{l(k)}$ and obtains a sample of the cost by observing $U_{Team(k)}$, the sample is denoted $Q_{l(k)}^{Team}(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_l)$. If no subscript is indicated, then the sample was obtained by all processors. Each processor l must

nent of the perturbation vector from being chosen uniformly or normally. Analysis by Spall [58] indicates that the algorithm is more efficient in its use of measurements than KW or KW-RD on large dimensional problems.

5.4.5 Numerical Experiments

In this section we provide illustrative examples of the behavior of the KW-type training algorithms KW (one-sided, two-sided), KW-GS, KW-RD, and KW-SP. As for the window algorithms, our intention is simply to illustrate the typical behavior of each algorithm on a reasonable problem instance.

Example: 2-Tand Topology

We again consider the team Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3 \tag{5.82}$$

$$\sigma_A^2 = \sigma_B^2 = 1, \quad p_0 = 0.75 \tag{5.83}$$

The optimal observation thresholds for this problem are $\alpha^* = 2.2209$, $\beta_0^* = 3.2521$, and $\beta_1^* = 1.5734$. The minimum probability of error is $P_\epsilon(\underline{\theta}) = .0794$. Throughout this section, the thresholds are again initialized to $\alpha = 2.0$, $\beta_0 = 2.5$, and $\beta_1 = 1.5$. The probability of error corresponding to this initialization is $P_\epsilon(\underline{\Theta}_1) = .1052$.

KW, Two-Sided: In this section we investigate the convergence behavior of the two-sided KW implementation. The stepsize and perturbation sequences are taken to be

$$\rho_k = \frac{1}{k}, \quad \delta_k = \frac{2.25}{(k)^{1/6}} \tag{5.84}$$

Figure 5-33 illustrates some typical threshold sample paths for the two-sided KW training algorithm. The axes are scaled as they were for the window algorithms in order to facilitate comparison. As expected from the results on the scalar parameter problem in Chapter 4, the sample paths are visibly smoother than the WIN paths, and transition more slowly. For this case, the thresholds converge relatively quickly

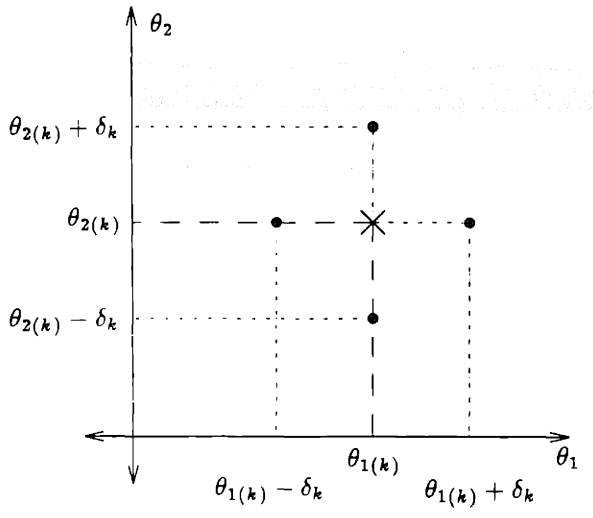


Figure 5-26: Sampling Pattern of Two-Sided KW method for the case $N = 2$. The $\times$ indicates the location of the current iterate $\underline{\theta}_k$. The dots indicate locations of samples. The two-sided technique uses $2N = 4$ samples for each update of $\underline{\theta}_k$ in this example.

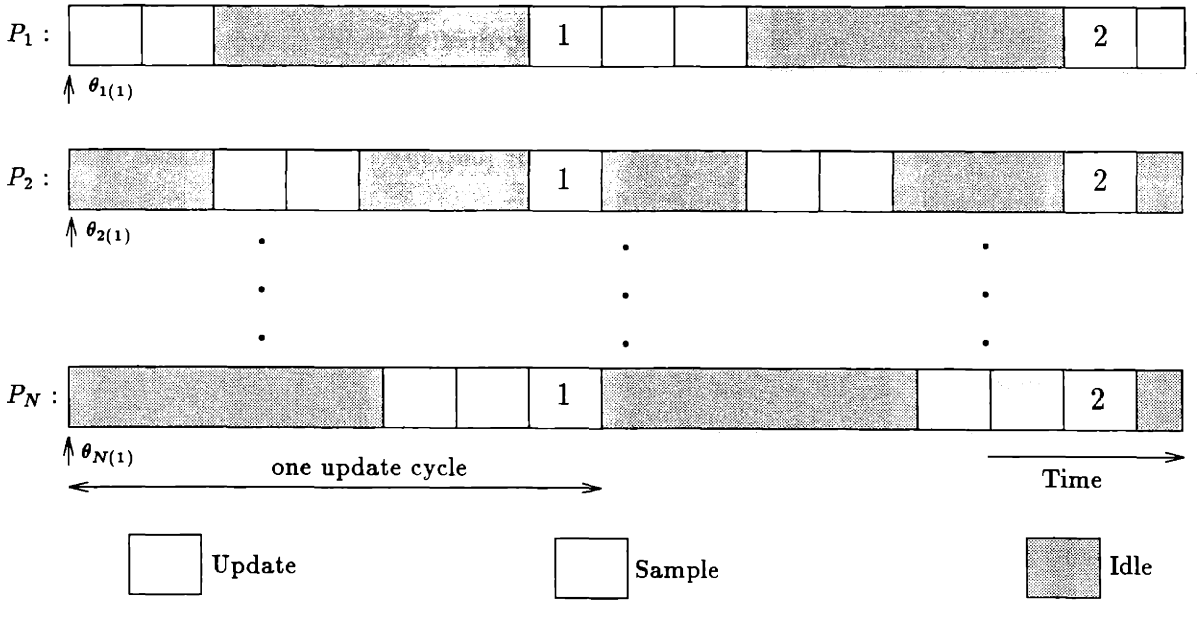


Figure 5-27: Timing Diagram for Multivariable KW with Two-Sided Approximations.

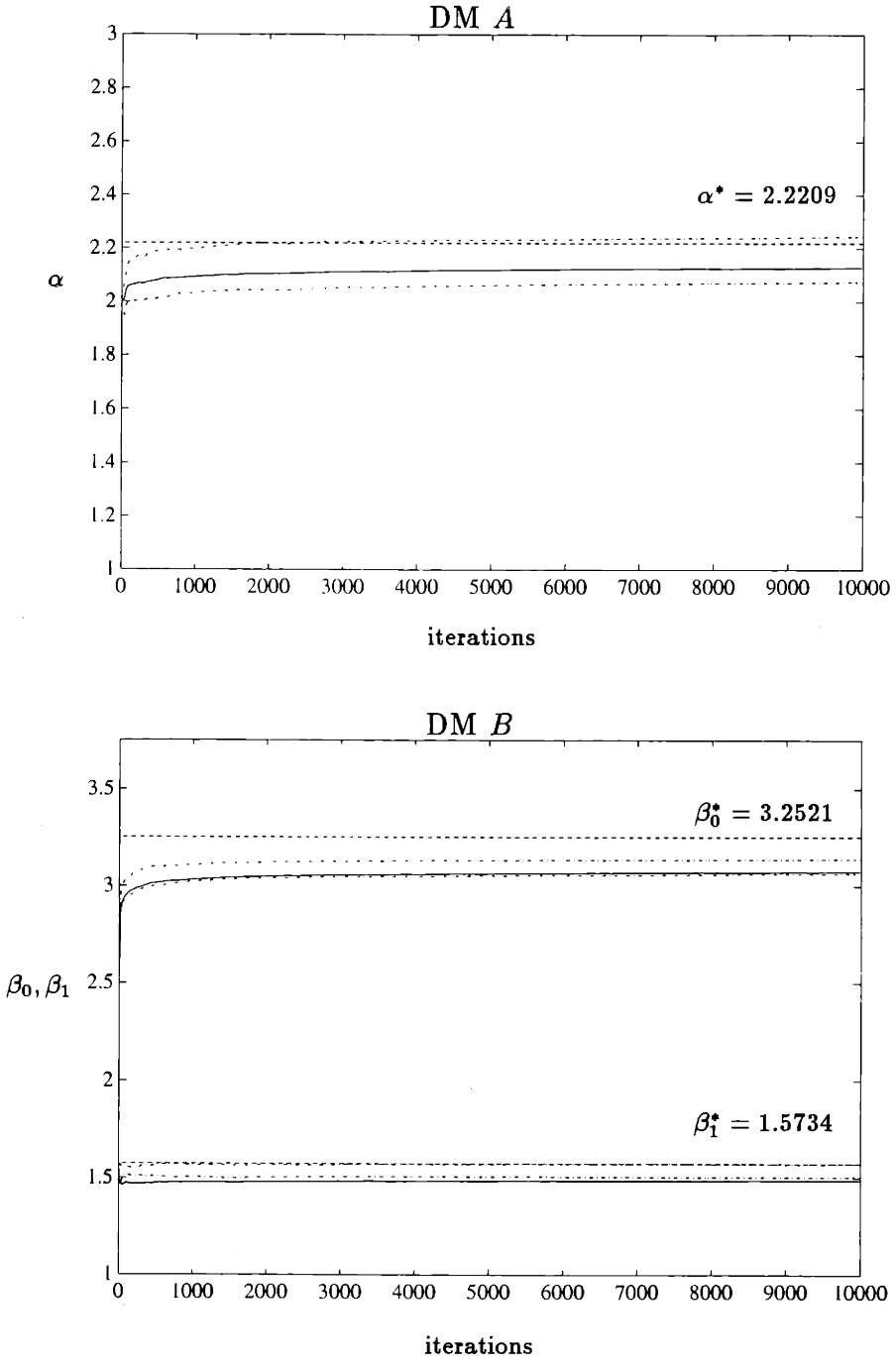


Figure 5-33: KW (two-sided): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

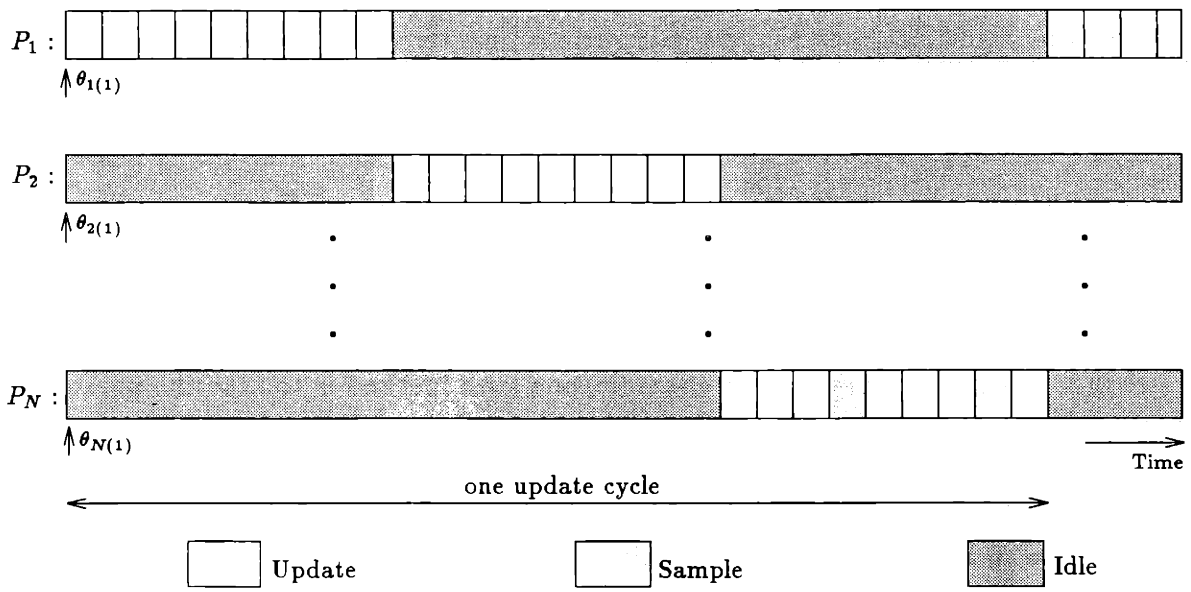


Figure 5-28: Typical Timing Diagram for Gauss-Seidel Implementation of Multivariable KW.

two samples. The figure indicates three updates of the component for each one-dimensional subproblem, but the number may be as small as one, or as large as approaching infinity.

As the number of updates in each one-dimensional subproblem becomes large, the algorithm approximates nonlinear Gauss-Seidel iterations, or cyclic coordinate descent. In this case, each one dimensional search is approximately solved to obtain a person-by-person optimal solution, i.e., a stationary point along that coordinate given that the other parameters are held fixed. In particular, if a large number of updates of coordinate θ_l corresponding to DM i are performed by processor l, then the result is approximate solution of the one-dimensional subproblem

$$\min_{\theta_{l(k)}} P_\epsilon^{Team}(\theta_{1(k+1)}, \ldots, \theta_{l-1(k+1)}, \theta_{l(k)}, \theta_{l+1(k)}, \ldots, \theta_{N(k)}) \tag{5.70}$$

where the other parameters are held fixed at their current values. Equivalently, minimization of the function $P_\epsilon^{Team}(\theta_l|\theta_j, j \neq l)$ is performed. Of course the solution is approximate because the exact solution is obtained only in the limit as the number of iterations goes to infinity. The difference between this algorithm and WIN-GS is

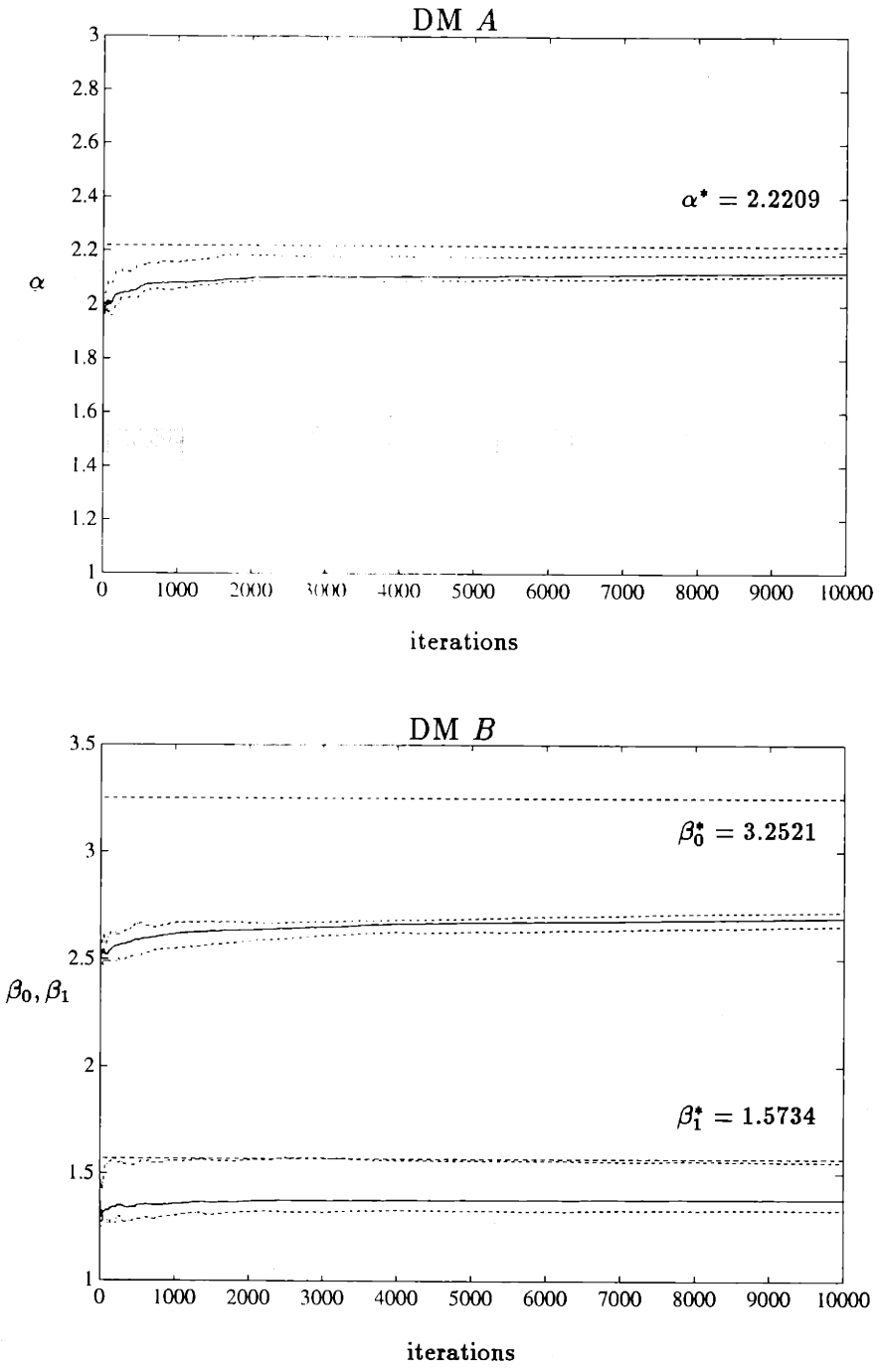


Figure 5-35: KW (one-sided): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

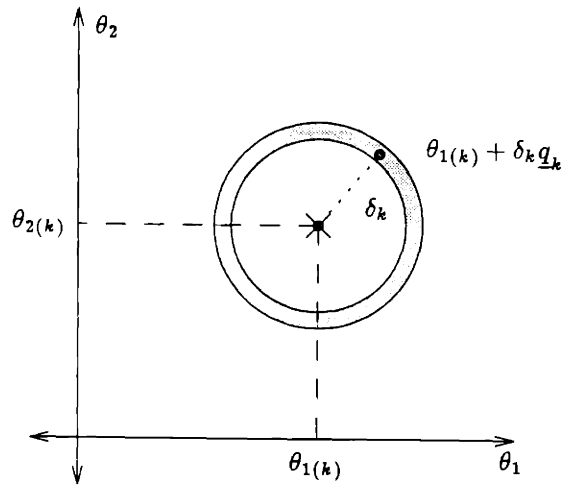


Figure 5-29: Sampling Pattern of One-Sided Random Directions implementation of the KW method for case $N = 2$. The $\times$ indicates the location of the current iterate $\underline{\theta}_k$. The shaded annulus represents the locus of points from which the second sample point is chosen. A typical point is illustrated by the dot. The technique uses 2 samples for each update, for *any* value of N.

coordinate axes. However, consider the one-sided random directions sampling pattern illustrated in Figure 5-29 for the case $N = 2$. One sample of the function is taken at the current iterate, while a second location is randomly chosen from the spherical locus of points indicated in the figure.

In general, at each time k a vector $\underline{q}_k$ is chosen at random from the set

$$S_N = \{\underline{X} \in \Re^N | \, \|\underline{X}\| = 1\} \tag{5.71}$$

which is the unit sphere around the origin[8]. The entire network parameter vector is then updated according to

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k, \quad k = 1, 2, \ldots \tag{5.72}$$

where the vector step $\underline{Z}_k$ is in direction $\underline{q}_k$, and is given by

$$\underline{Z}_k(\underline{X}_k, \underline{\Theta}_k, \underline{q}_k, \delta_k) = [Q_k^{Team}(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{q}_k) - Q_k^{Team}(\underline{X}_k, \underline{\Theta}_k)]\underline{q}_k/\delta_k \tag{5.73}$$

[8]Note that $\underline{q}_k$ denotes a random vector despite being represented with lower case.

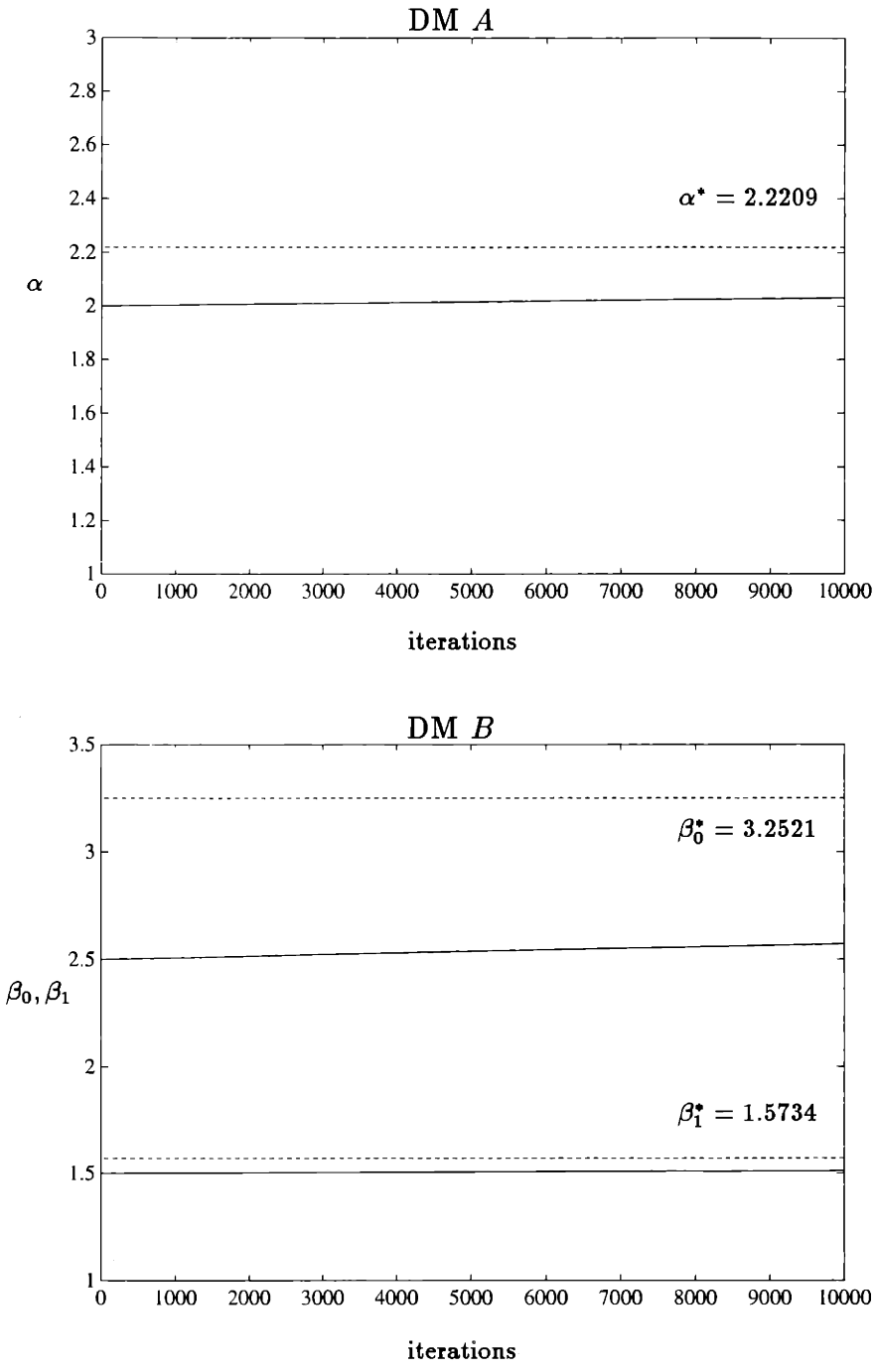


Figure 5-37: KW-GS (two-sided, no restarts): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

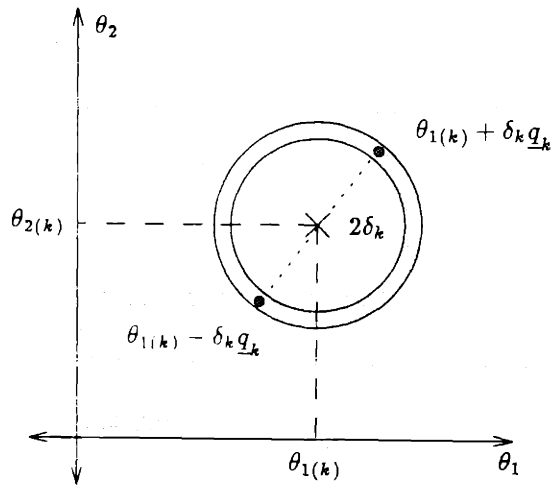


Figure 5-31: Sampling Pattern of Two-Sided Random Directions implementation of the KW method for case $N = 2$. The $\times$ indicates the location of the current iterate $\underline{\theta}_k$. The shaded annulus represents the set of points from which the sample points are taken. The first is selected randomly, with the second taken as the negative of the first. Typical points are illustrated by the dots. The technique also uses 2 samples for each update, for any value of N.

parameters.

A two-sided variant also exists, with the sampling pattern shown in Figure 5-31. Steps for the two-sided variant are given by

$$\underline{Z}_k(\underline{X}_k, \underline{\Theta}_k, \underline{q}_k, \delta_k) = [Q_k^{Team}(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{q}_k) - Q_k^{Team}(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{q}_k)]\underline{q}_k / 2\delta_k \quad (5.75)$$

where $\underline{q}_k \in \mathcal{S}_N$. The local step computed by processor l updating component θ_l corresponding to DM i is given by

$$Z_{l(k)}(X_{l(k)}, U^+_{Team(k)}, U^-_{Team(k)}, \Theta_{l(k)}, q_{l(k)}, \delta_k) = \quad (5.76)$$

$$\frac{[Q_k^{Team}(X_{l(k)}, U^+_{Team(k)}) - Q_k^{Team}(X_{l(k)}, U^-_{Team(k)})]q_{l(k)}}{2\delta_k} \quad (5.77)$$

The obvious difficulty of this approach is generating, in distributed fashion, the perturbation vector $\underline{q}_k$ which is uniformly distributed on the sphere. In particular, each local update of $\theta_{l(k)}$ requires the component $q_{l(k)}$.

5.4.4 Simultaneous Perturbation (KW-SP)

A technique having similar benefits in terms of low numbers of required samples, but which is better suited to distributed implementation, has been suggested by Spall [58]. Motivation for developing the algorithm came from large-scale systems and neural networks, both of which represent settings in which the number of parameters to be updated is sufficiently large to make the updates very expensive. Application of the algorithm to the problem of training neural networks is similar in spirit to what was proposed by Dembo and Kailath in [13].

The technique is referred to Simultaneous Perturbation (SP) and also involves sampling the function at two randomly generated locations. However, Spall suggests generating the locations using N-dimensional random vectors with *independent* components. Such a technique is much better suited to implementation in a distributed setting because each component of the search vector may be generated without knowledge of the others.

To qualify, the KW-SP algorithm is of the form

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k(\underline{X}_k, \underline{\Theta}_k, \underline{\Delta}_k, \delta_k), \quad k = 1, 2, \ldots \tag{5.78}$$

where

$$\underline{\Delta}_k = \left[\Delta_{1(k)}, \Delta_{2(k)}, \ldots, \Delta_{N(k)}\right]^T \tag{5.79}$$

is a vector of mutually independent zero-mean random variables. The overall network-wide step computed by the processors at time k is given by

$$\underline{Z}_k(\underline{X}_k, \underline{\Theta}_k, \underline{\Delta}_k, \delta_k) = \sum_{l=1}^{N} \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{\Delta}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{\Delta}_k)]\underline{e}_l}{2\delta_k \Delta_{l(k)}} \tag{5.80}$$

Note that this technique allows for estimation of each partial derivative without independently observing the effect of perturbations of the parameter at the output. In the distributed implementation, component θ_l corresponding to DM i is updated at

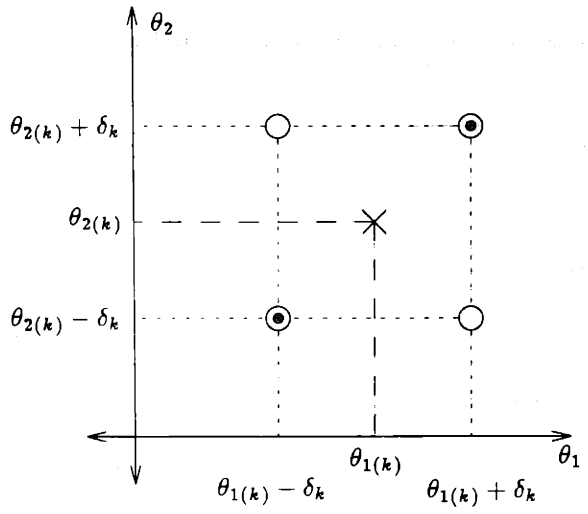


Figure 5-32: Sampling Pattern of Simultaneous Perturbation Technique for the case $N = 2$. The $\times$ indicates the location of the current iterate $\underline{\theta}_k$. The shaded circles illustrate the locus of possible sample locations. The dark dots indicate typical samples. The technique uses 2 samples for each update, for any value of N.

time k using the local step

$$Z_{l(k)}(X_{l(k)}, U^+_{Team(k)}, U^-_{Team(k)}, \Theta_{l(k)}, \Delta_{l(k)}, \delta_k) = \frac{[Q_k^{Team}(X_{l(k)}, U^+_{Team(k)}) - Q_k^{Team}(X_{l(k)}, U^-_{Team(k)})]}{2\delta_k \Delta_{l(k)}} \tag{5.81}$$

The author of [58] suggests choosing the $\Delta_{l(k)}$, $l = 1, \ldots, N$ to be symmetric Bernoulli random variables taking on the values ± 1 with equal probability. This results in the sampling pattern shown in Figure 5-32. Note that generation of the perturbation vector in this fashion is easily accomplished in distributed fashion, and that each processor only requires its locally generated component to perform its update.

Comparing equations (5.81) and (5.77), it is clear that the Simultaneous Perturbation technique is not a special case of the two-sided random directions technique since the corresponding component of the random vector enters the computation of the local step in the numerator for two-sided KW-RD and in the denominator for KW-SP. In Chapter 6 it is shown that different conditions are required on $\underline{q}_k$ than on $\underline{\Delta}_k$. In particular, a boundedness condition on $E\{|\Delta^{-1}_{l(k)}|\}$ precludes each compo-

nent of the perturbation vector from being chosen uniformly or normally. Analysis by Spall [58] indicates that the algorithm is more efficient in its use of measurements than KW or KW-RD on large dimensional problems.

5.4.5 Numerical Experiments

In this section we provide illustrative examples of the behavior of the KW-type training algorithms KW (one-sided, two-sided), KW-GS, KW-RD, and KW-SP. As for the window algorithms, our intention is simply to illustrate the typical behavior of each algorithm on a reasonable problem instance.

Example: 2-Tand Topology

We again consider the team Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3 \tag{5.82}$$

$$\sigma_A^2 = \sigma_B^2 = 1, \quad p_0 = 0.75 \tag{5.83}$$

The optimal observation thresholds for this problem are $\alpha^* = 2.2209$, $\beta_0^* = 3.2521$, and $\beta_1^* = 1.5734$. The minimum probability of error is $P_\epsilon(\underline{\theta}) = .0794$. Throughout this section, the thresholds are again initialized to $\alpha = 2.0$, $\beta_0 = 2.5$, and $\beta_1 = 1.5$. The probability of error corresponding to this initialization is $P_\epsilon(\underline{\Theta}_1) = .1052$.

KW, Two-Sided: In this section we investigate the convergence behavior of the two-sided KW implementation. The stepsize and perturbation sequences are taken to be

$$\rho_k = \frac{1}{k}, \quad \delta_k = \frac{2.25}{(k)^{1/6}} \tag{5.84}$$

Figure 5-33 illustrates some typical threshold sample paths for the two-sided KW training algorithm. The axes are scaled as they were for the window algorithms in order to facilitate comparison. As expected from the results on the scalar parameter problem in Chapter 4, the sample paths are visibly smoother than the WIN paths, and transition more slowly. For this case, the thresholds converge relatively quickly

to the vicinity of the optimal thresholds. Each sample path is run out for 10,000 iterations, and the average path represents an average over 4 independent sample paths. Since each parameter update in the two-sided technique requires two samples of the cost, a total of 20,000 network measurements (and associated team decisions) were required to generate each sample path for a single parameter. Thus, 80,000 measurements were required to generate each averaged sample path, and a total of $3(80{,}000) = 240{,}000$ network measurements were required to create all of the averaged paths,

The performance on the average sample path is depicted in Figure 5-34. The probability of error has dipped below 0.085 within 500 iterations, although after this point continued decreases in the cost come very slowly. In comparison with the WIN algorithm, the convergence rate can be seen to be clearly inferior; in particular, updates of the KW algorithm are less effective. However, as a function of total measurements required, the algorithms may be comparable, since the KW methods require no estimation phases.

KW, One-Sided: In this section we examine the behavior of the one-sided KW technique. Stepsizes for this algorithm are taken to be

$$\rho_k = \frac{1}{k}, \quad \delta_k = \frac{2.25}{(k)^{1/4}} \tag{5.85}$$

Figure 5-35 illustrates several typical sample paths. These paths were again run out to 10,000 iterations, with the average paths computed over 4 independent sample paths. Each sample path requires 20,000 network measurements, so that a total of 80,000 are required to generate the averaged path. Since one measurement is common to the update of every parameter, the total number of network measurements required to generate all 3 averaged sample paths is $4(20{,}000 + 10{,}000 + 10{,}000) = 160{,}000$. In comparison with the two-sided technique, convergence of the thresholds is visibly slower; this is most obvious for $\{\beta_{0(k)}\}$ and $\{\beta_{1(k)}\}$.

The corresponding error probability for the average sample path is illustrated in Figure 5-36. Convergence is significantly slower than that obtained for the two-sided

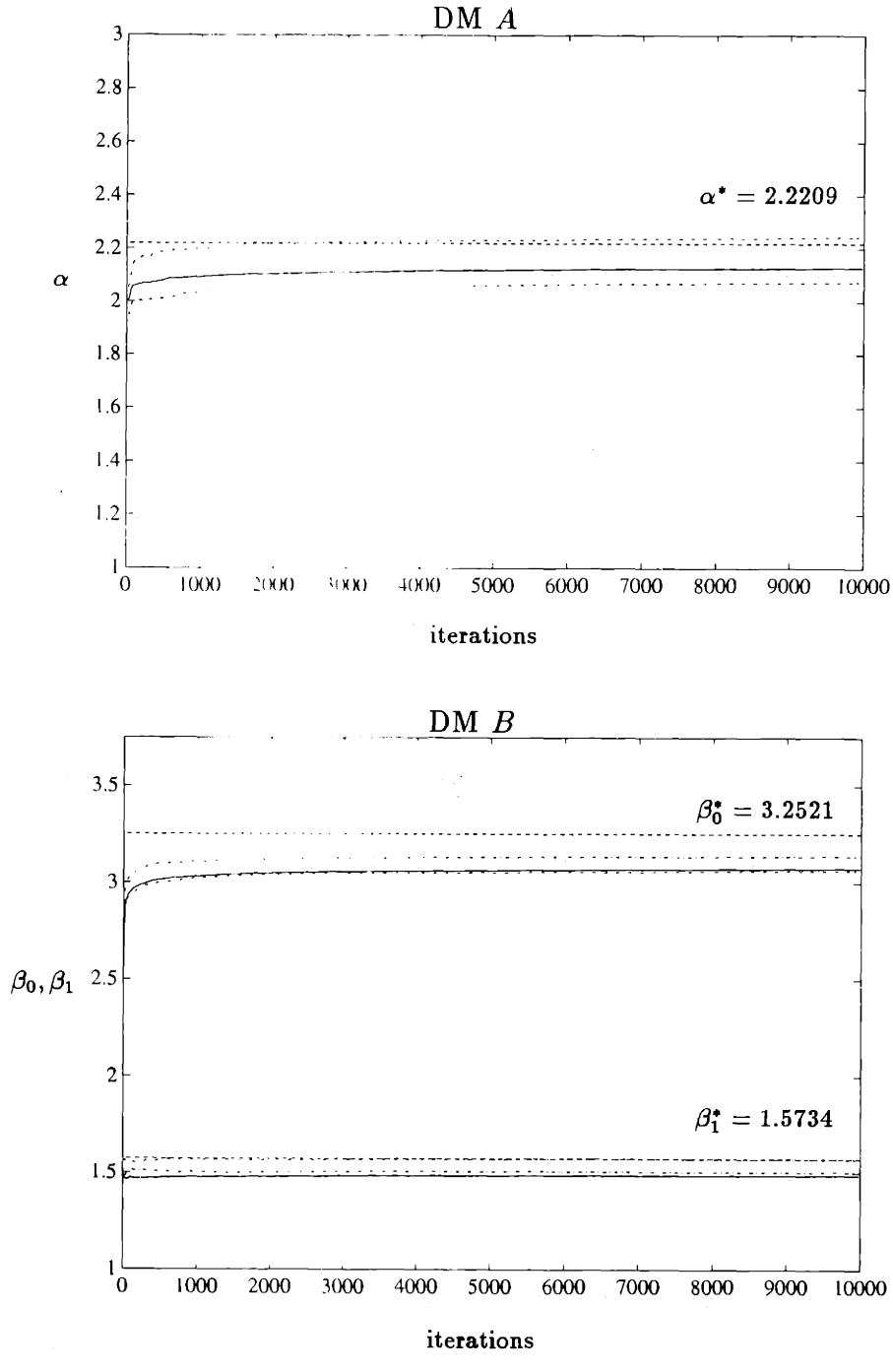


Figure 5-33: KW (two-sided): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

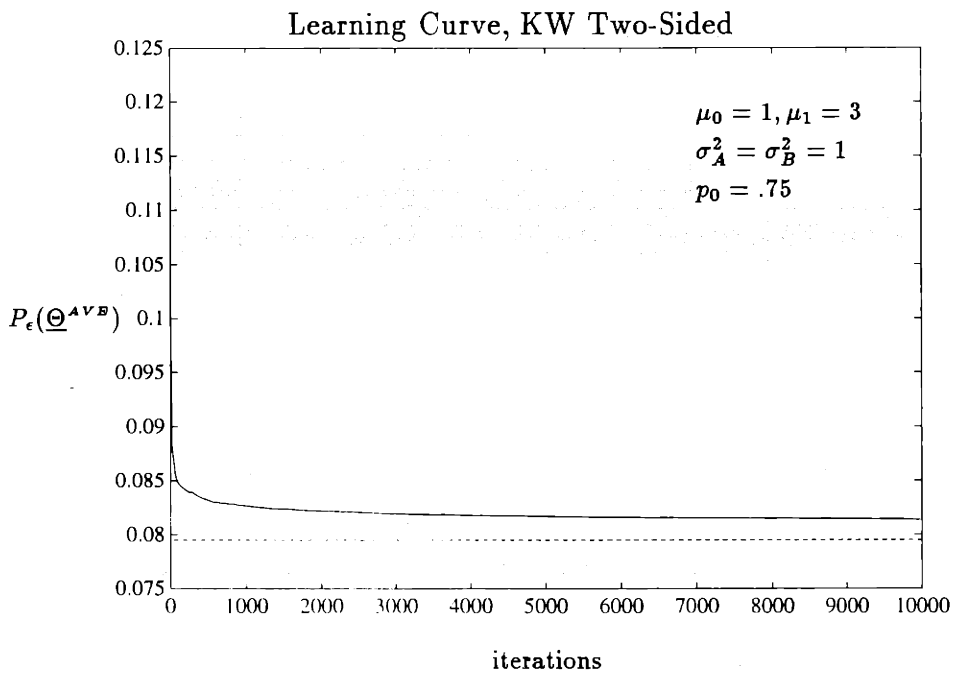


Figure 5-34: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVB})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ (dashed). Initial error probability is $P_\epsilon(\underline{\Theta}_1) = 0.1052$.

technique, although the algorithm is clearly reducing the cost.

KW-GS: In this section we investigate a Gauss-Seidel implementation of the two-sided KW technique. As with the WIN algorithms, we first investigate the standard algorithm, and then attempt an ad hoc modification involving restarting the stepsize at the beginning of each subproblem. We again take the stepsize and perturbation sequences to be $\rho_k = 1/k$ and $\delta_k = 2.25/(k)^{1/6}$.

Sample paths for the thresholds are shown in Figure 5-37. Each solid path in the figure is composed of 10 subproblems of 1000 iterations, each averaged 4 times as the algorithm advances. As was observed for the WIN-GS algorithm with no restarts, the convergence of the algorithm, while steady and almost linear in appearance, is extremely slow.

The performance of the KW-GS algorithm is indicated in Figure 5-38.

The incredibly slow convergence rate of this algorithm again motivates an ad hoc method in which a sequence of one-dimensional searches are implemented in order to better approximate cyclic coordinate descent on the cost surface. Figure 5-39

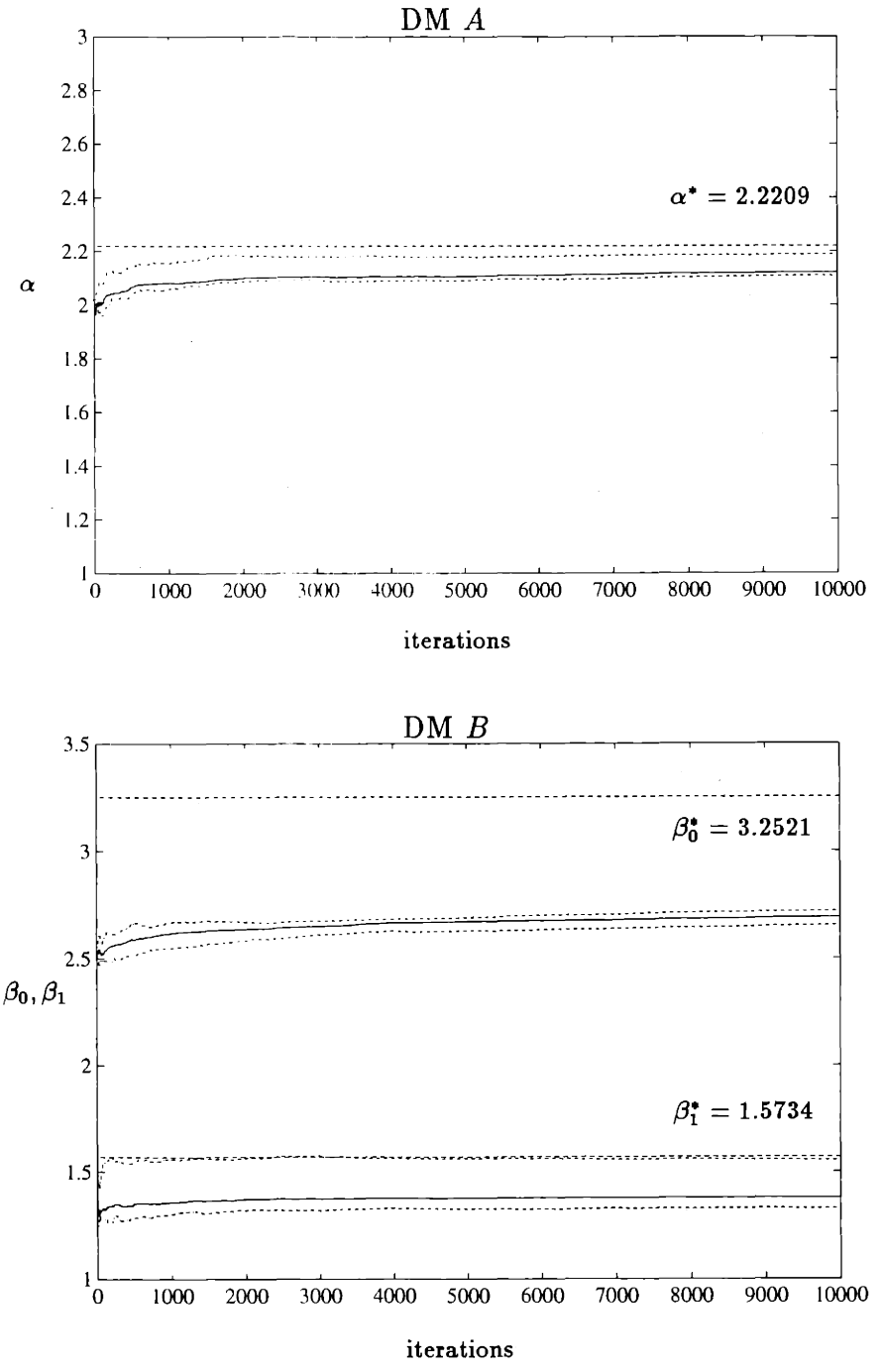


Figure 5-35: KW (one-sided): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

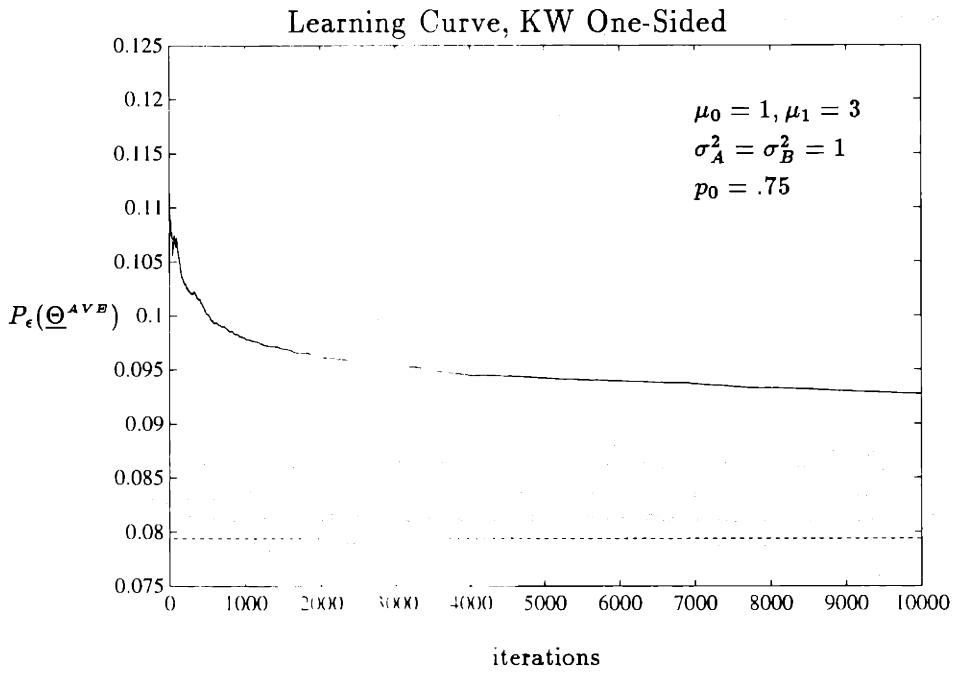


Figure 5-36: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ shown dashed. Initial error probability is $P_\epsilon(\underline{\Theta}_1) = 0.1052$.

illustrates average sample paths for the thresholds for a Gauss-Seidel implementation in which each subproblem is solved out to 5000 iterations, and averaged 4 times. A total of 4 update cycles of this type were executed. The stepsize was restarted to begin each subproblem, and in contrast to the ad hoc WIN-GS algorithm, each succeeding subproblem was initialized to the final value of the previous one. This was observed to provide better performance since the stepsize decreases quite rapidly.

The corresponding performance of the algorithm is shown in Figure 5-40. The error is reduced dramatically after a single update cycle.

KW-RD, Two-Sided: In this section, we investigate the behavior of the two-sided random directions technique. In this technique, only two network measurements are used to compute every network update. Each sample path is run out to 10,000 iterations (updates), and the average paths represent an average over 4 independent sample paths. Thus, the total number of network measurements required to generate the averaged sample paths for the network is 4(20,000) = 80,000. Stepsize and perturbation sequences are chosen as for the two-sided KW technique.

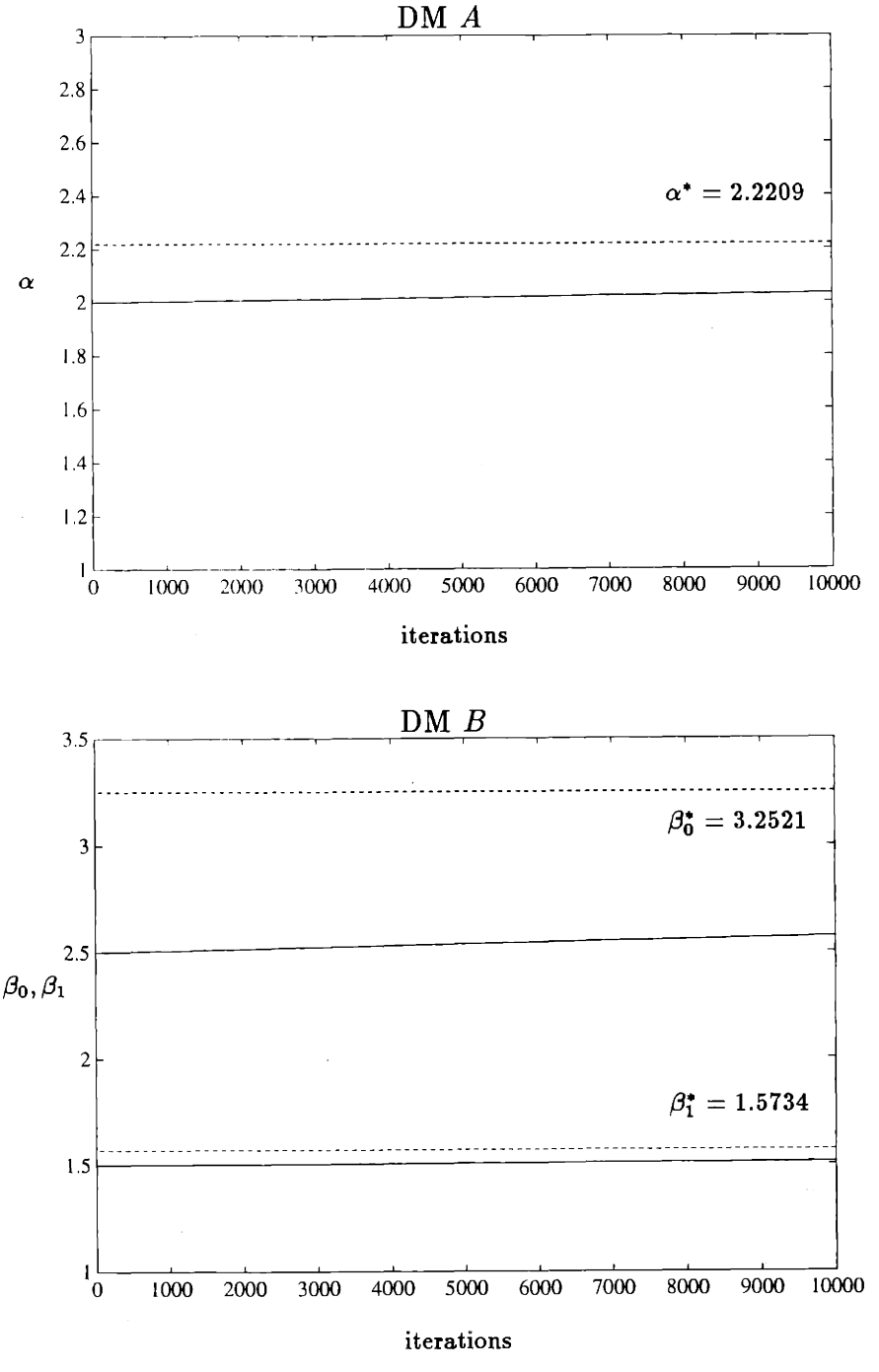


Figure 5-37: KW-GS (two-sided, no restarts): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

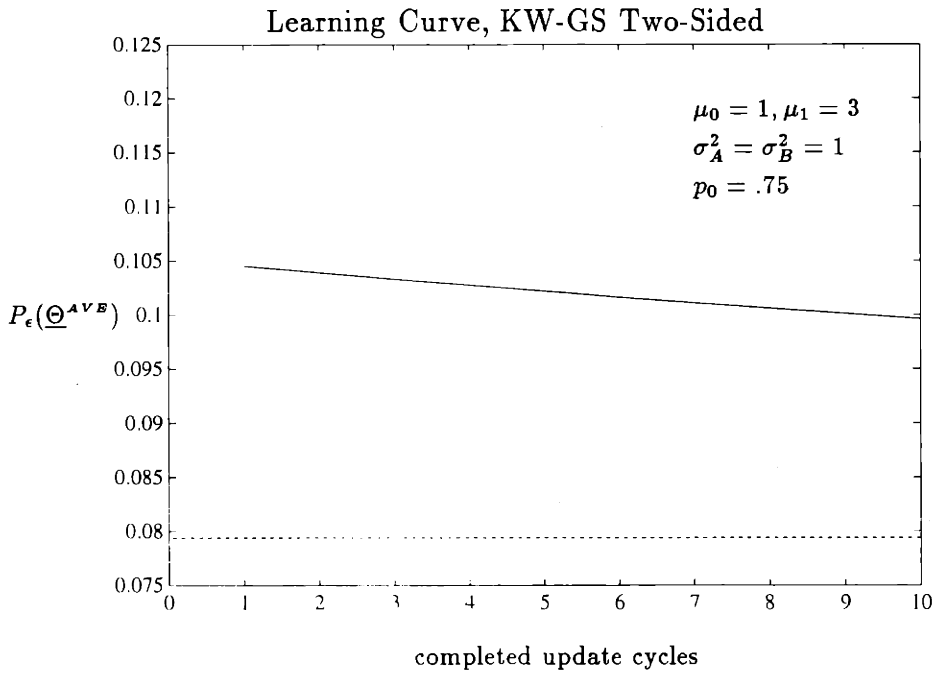


Figure 5-38: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ shown dashed. Initial error probability is $P_\epsilon(\underline{\Theta}_1) = 0.1052$.

Typical sample paths for the random directions variant are shown in Figure 5-41 and the corresponding performance is shown in Figure 5-42. The performance of the algorithm is only sightly worse that that of the one-sided KW technique, a fact that is perhaps surprising in view of the significantly fewer network measurements the algorithm employs.

KW-SP: In this section we try the simultaneous perturbation technique of Spall. It also requires only two network measurements per network update, and thus has the same data usage as the random directions method.

Figures 5-43 and 5-44 indicate that the algorithm performs significantly better than either the random directions method or the one-sided KW method.

Example: 3-Vee Topology

In this section. we illustrate the best of the previous algorithms, the two-sided KW method, on the 3-Vee topology. We again consider the team Gaussian detection

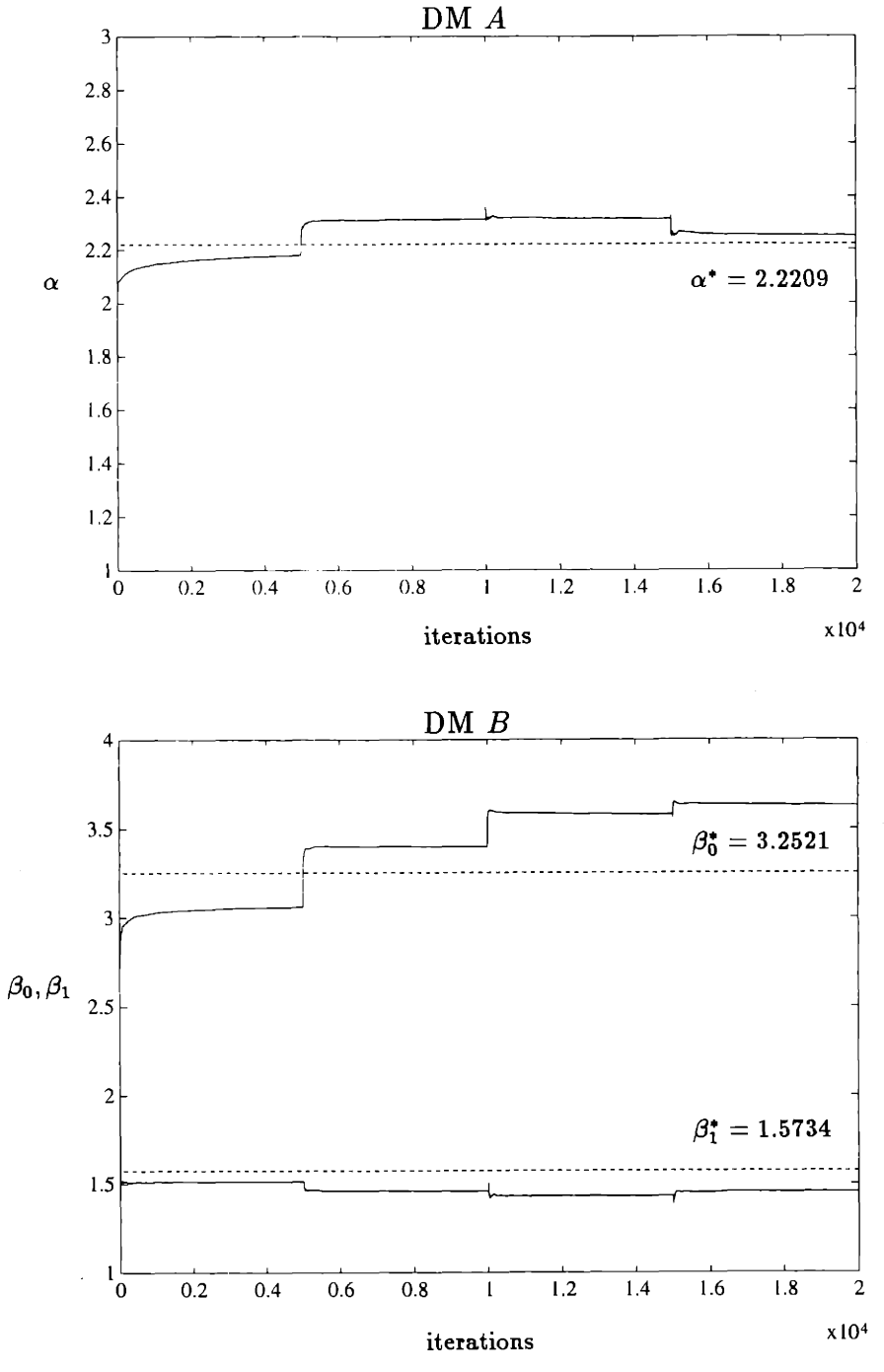


Figure 5-39: KW-GS ad hoc (two-sided): Average sample paths of $\{\ \underline{\Theta}_k\ \}$ during training averaged over 4 paths (solid), and optimal threshold values (dashed). Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

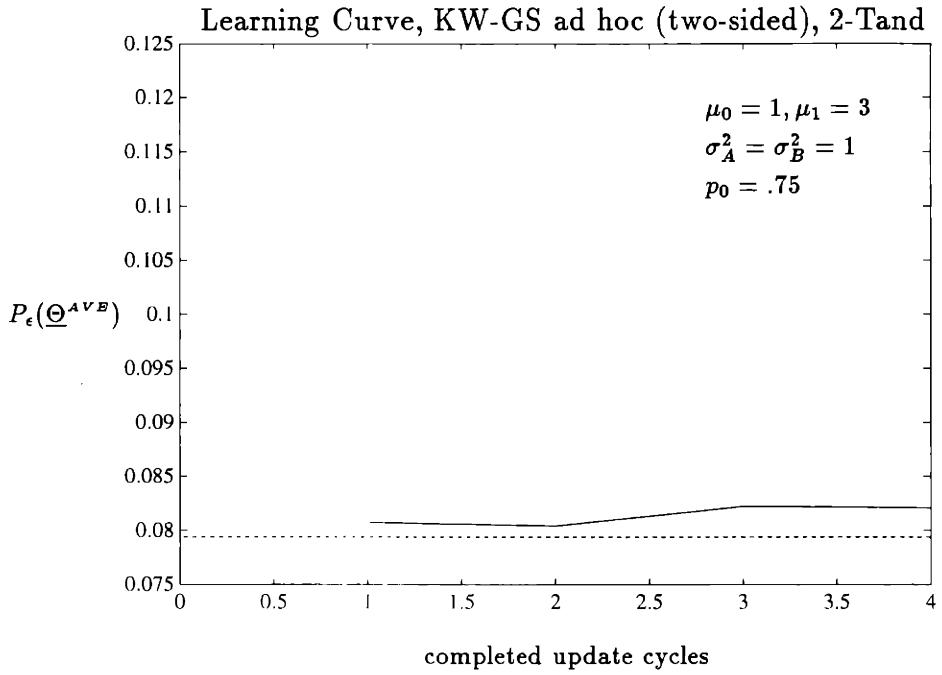


Figure 5-40: Sample Path of $P_\epsilon(\underline{\Theta}_k^{AVE})$ at end of completed update cycles (solid). Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ (dashed).

problem

$$\mu_0 = 1, \quad \mu_1 = 3 \tag{5.86}$$

$$\sigma_A^2 = 1.5, \quad \sigma_B^2 = 0.5, \quad \sigma_C^2 = 1.0, \quad p_0 = 0.75 \tag{5.87}$$

The optimal thresholds for this problem are $\alpha^* = 2.0457$, $\beta^* = 2.1287$, $\xi_{00}^* = 4.2818$, $\xi_{01}^* = 1.8110$, $\xi_{10}^* = 2.9391$, and $\xi_{11}^* = 0.4683$. The optimal probability of error is $P_\epsilon(\underline{\theta}^*) = 0.0388$.

The same comments regarding initialization of the algorithm apply. We will choose the initialization $\alpha_1 = 2.0$, $\beta_1 = 2.0$, $\xi_{00} = 3.0$, $\xi_{01} = 2.0$, $\xi_{10} = 1.0$, and $\xi_{11} = 0.0$. Notice that the initializations of ξ_{01} and ξ_{10} require that these thresholds uncross to achieve their optimal values. The probability of error corresponding to this initial setting of the thresholds is $P_\epsilon(\underline{\theta}_1) = 0.1119$.

KW, Two-Sided: The stepsize and perturbation sequences are again taken to be $\rho_k = 1/k$ and $\delta_k = 2.25/(k)^{1/6}$.

In Figures 5-45, 5-46 the sample paths and performance of the KW algorithm are

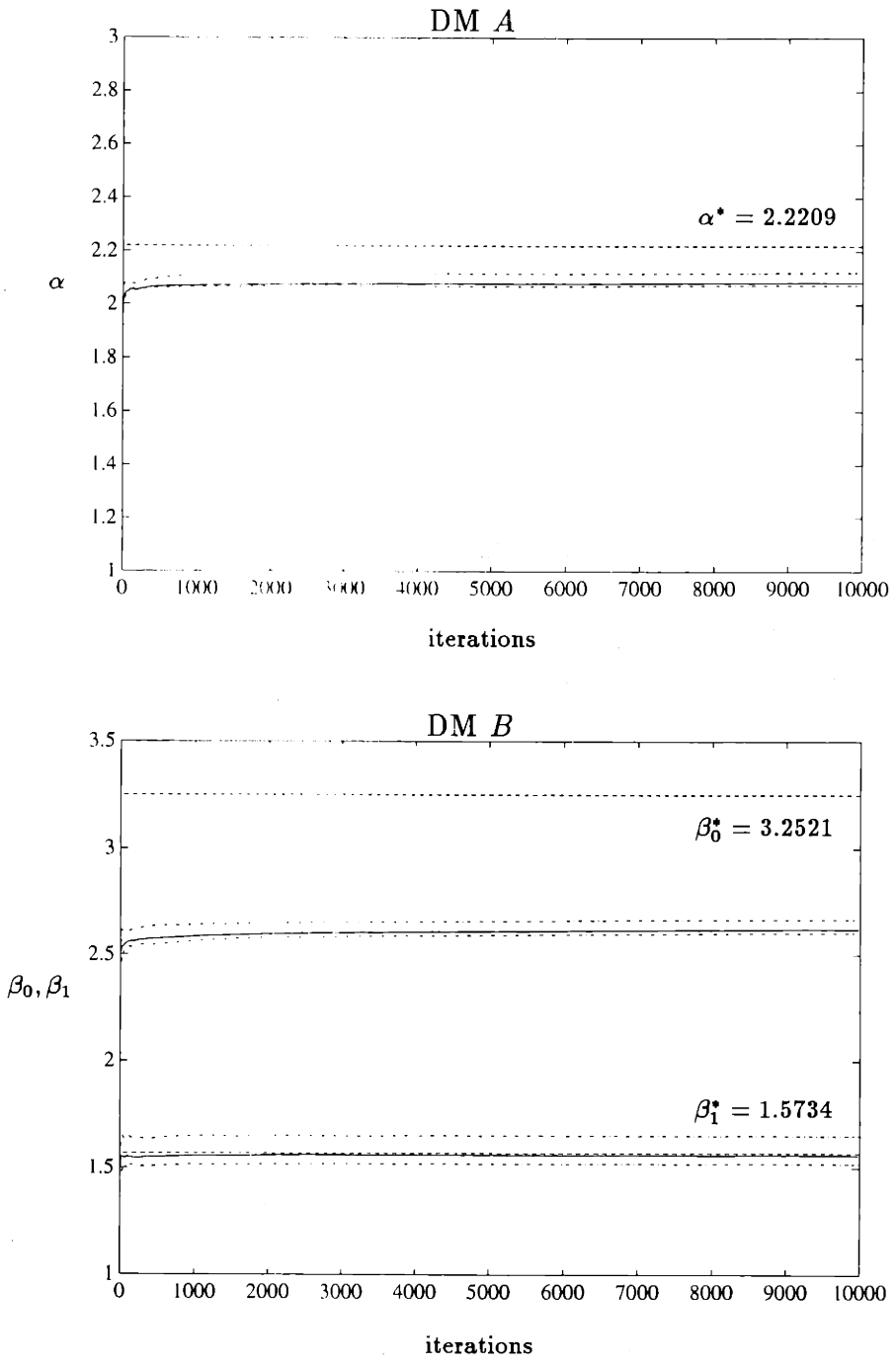


Figure 5-41: KW-RD (two-sided): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

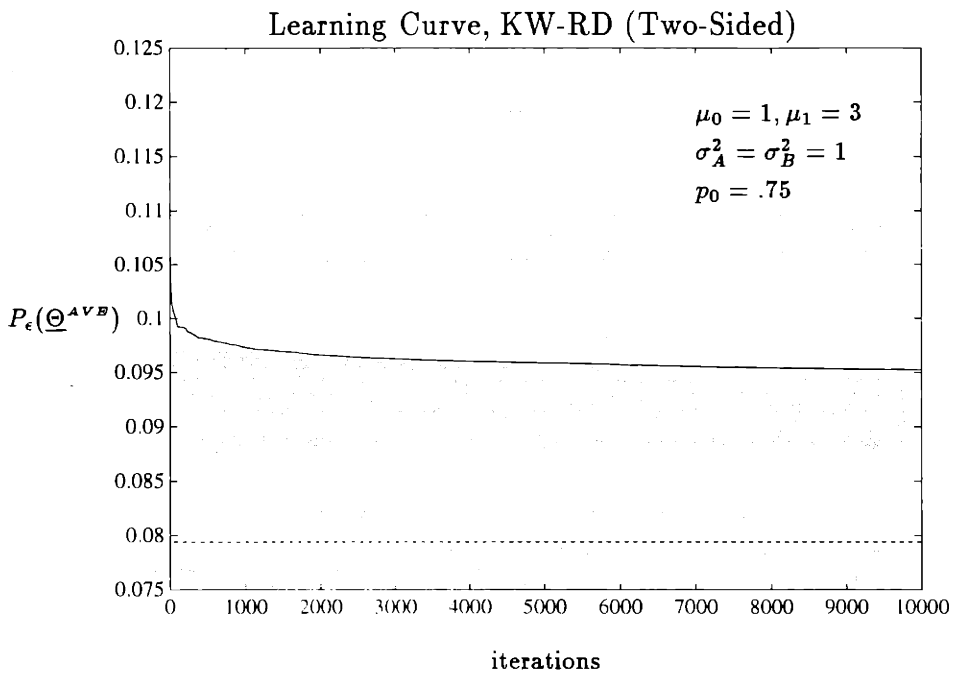


Figure 5-42: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVB})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ shown dashed. Initial error probability is $P_\epsilon(\underline{\Theta}_1) = 0.1052$.

shown. Each sample path was run out to 15,000 iterations. Average paths are the result of averaging 3 independent sample paths.

In comparison to the WIN algorithms on 3-Vee, convergence of this algorithm is very slow. The thresholds show only slight progress from their initial positions, and ξ_{01} and ξ_{10} are not close to uncrossing, even after 15,000 iterations. However, reduction of the cost is clear.

This section has served to indicate that the KW algorithms we propose exhibit reasonable behavior. In particular, all variants clearly resulted in descent on the cost surface. The major difficulty appears to be the huge number of network measurements and team decision processes which must be executed to adapt the thresholds. However, it is certainly possible that with more numerical experience, the convergence rate of the algorithms might be significantly improved.

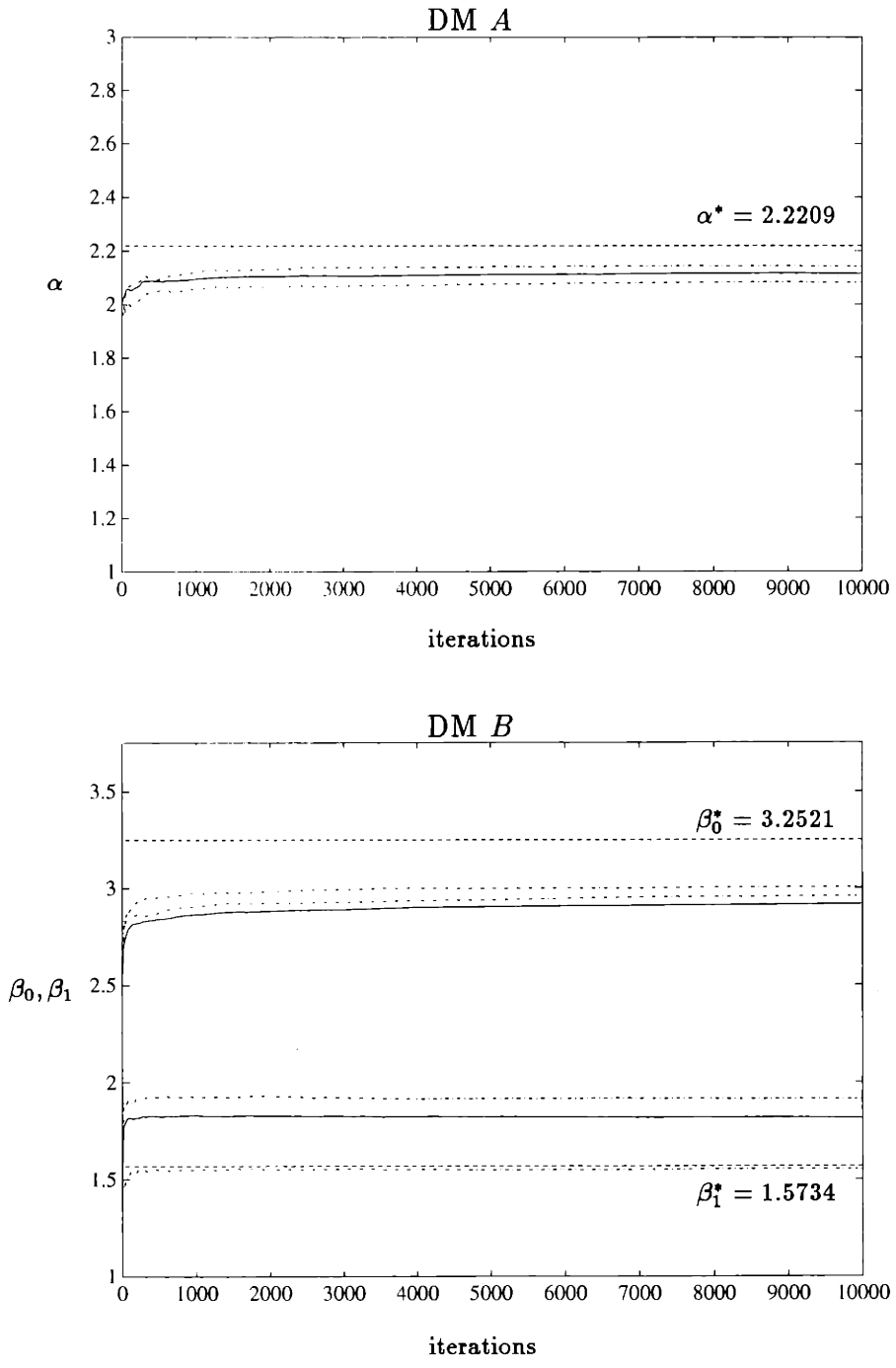


Figure 5-43: KW-SP: Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 4 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Thresholds initialized at $\alpha_1 = 2.0$, $\beta_{0(1)} = 2.5$, and $\beta_{1(1)} = 1.5$. Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$.

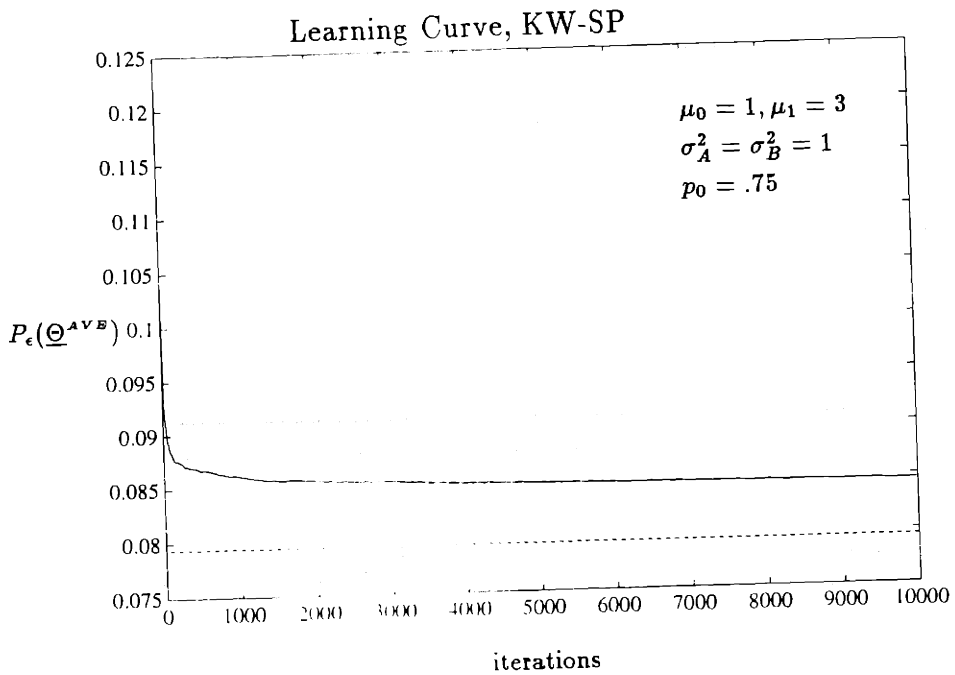


Figure 5-44: Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$. Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$ shown dashed. Initial error probability is $P_\epsilon(\underline{\Theta}_1) = 0.1052$.

5.5 Chapter Conclusions

We have presented a variety of synchronous distributed nonparametric training algorithms that may be grouped into two distinct classes, the so-called WIN-Type algorithms and the KW-Type algorithms. All of the techniques embody schemes for estimating the gradient in distributed fashion. The primary problem that had to be addressed was how each processor could infer enough information concerning the values of the other network parameters and measurements so that its partial derivative estimate could be locally computed. Discussion of the WIN algorithms centered around computation of the coupling costs, while the discussion of the KW algorithms focused on sampling techniques.

The WIN-Type algorithms model the analytic form of the partial derivatives of the cost with respect to the parameters explicitly. In particular, at each processor estimates of the operating points corresponding to the current values of the other network threshold parameters are obtained through communication, and locally com-

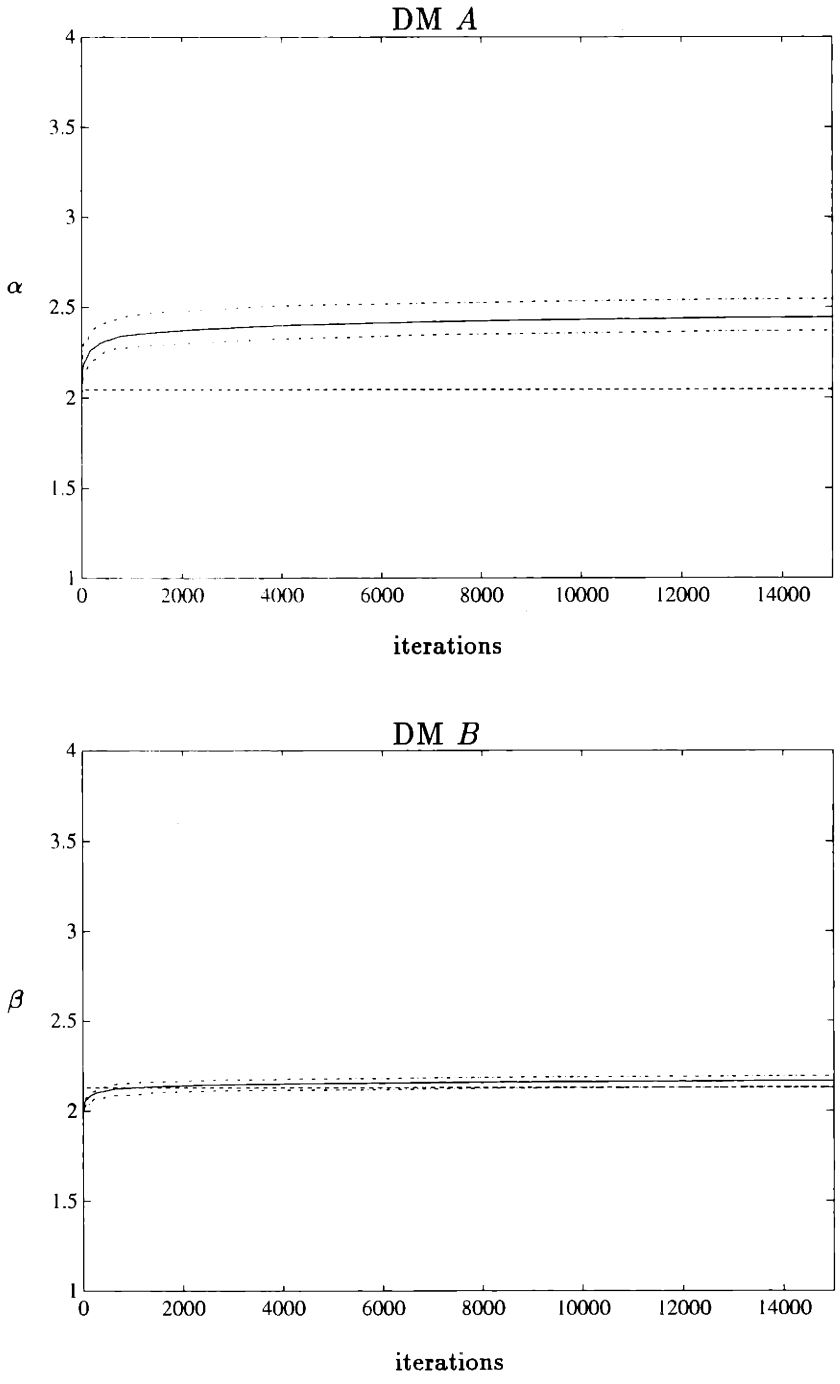
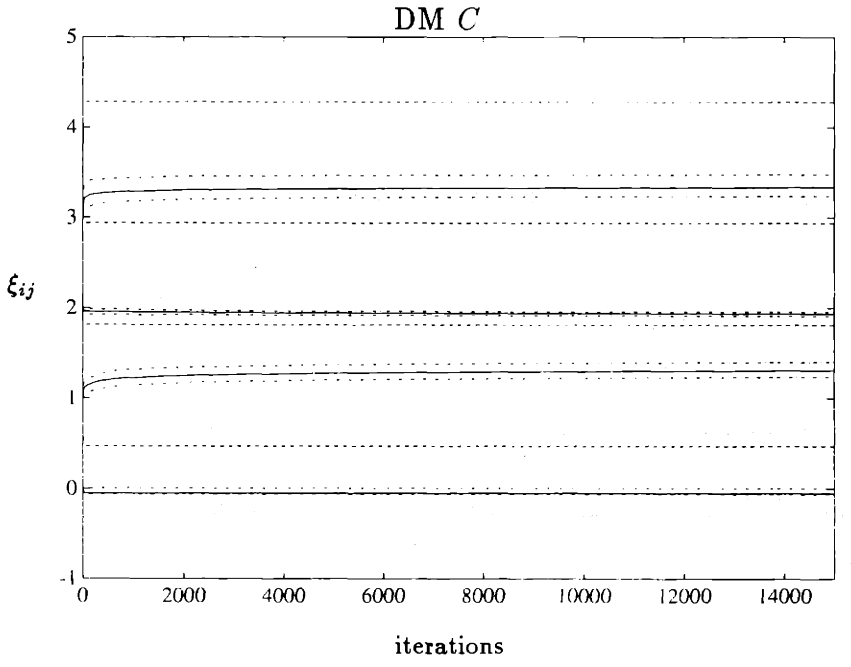
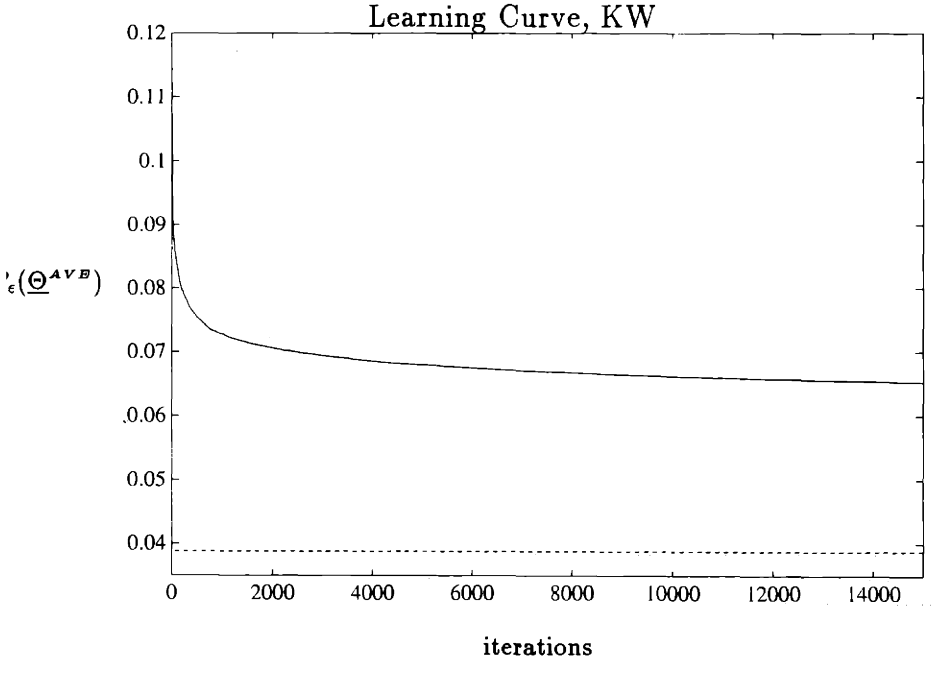


Figure 5-45: 3-Vee, KW (two-sided): Sample paths of $\{\underline{\Theta}_k\}$ during training; average over 3 paths (solid), some typical sample paths (dotted and dashed), and optimal threshold values (dashed). Gaussian Case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = 1.5$, $\sigma_B^2 = 0.5$, $\sigma_C^2 = 1.0$, $p_0 = 0.75$.





5-46: 3-Vee, KW (cont'd): Lower curve; Sample Path of $\{P_\epsilon(\underline{\Theta}_k^{AVE})\}$. Optimal $P_\epsilon(\underline{\Theta}^*) = 0.0388$ shown dashed. Initial error probability is $P_\epsilon(\underline{\Theta}_1) = 0.1119$.

bined to form an estimate of the coupling costs. Since the dependence of the local parameter update on the other processors is completely captured by these costs, the update may be computed based on local feedback only. The analytic form of the coupling costs must be known a priori, and is dictated by the network topology as described in Chapter 3. Thus, each DM must essentially know the overall network topology and how it is tied in, in order to rely purely on local feedback to perform its updates. Furthermore, this information must be continually updated, using inter-DM communication, in order to maintain the validity of these representations as the state of the network evolves.

The following WIN variants were suggested:

WIN: The basic Jacobi-type iteration, corresponding to multivariable gradient descent, in which the estimation and update phases are synchronized across all network DMs.

WIN-GS: A Gauss-Seidel implementation in which approximate person-by-person optimal solutions are achieved by having the processors cycle through solving sequences of one-dimensional problems.

WIN-BP: A back-propagation implementation of WIN employing a reduced communication scheme based on the optimal control formulation. In particular, using a forward-propagating state equation in conjunction with a backward-propagating costate equation, the coupling costs may be computed at each processor using only communication from the associated DMs' immediate predecessors and successors.

In contrast to the WIN algorithms, KW-Type algorithms do not model the functional form of the partial derivatives at all, but rather infer their values directly using finite difference approximations of the team cost function. In order to do this, a team decision process must be executed to generate every sample of the team cost. Since no local models are maintained, no inter-processor communication is required. Furthermore, it is not necessary for a DM (its processors) to know anything about the topology of which it is a part.

KW: The standard KW technique which corresponds to gradient descent with each component of the gradient replaced with a finite difference approximation along that coordinate. Both one-sided and two-sided variants were discussed, where the one-sided technique required $N+1$ samples to perform each network update, and the two-sided required $2N$.

KW-GS: A Gauss-Seidel implementation corresponding to approximate nonlinear Gauss-Seidel iterations on the cost surface.

KW-RD: Search scheme requiring only two samples of the function to perform each network update. For the one-sided variant, one sample of the function is taken randomly from a sphere of radius δ_k around the current iterate and the other at the location of the iterate, while in the two-sided implementation the first is randomly chosen from the sphere, and the second is taken to lie directly opposite it on the sphere.

KW-SP: An alternative search scheme, better suited to distributed implementation, that requires only two samples of the function for each network update, but where each component of the perturbation vector is independently generated by each processor.

In comparing the two classes of algorithms, certain tradeoffs are clearly evident. The WIN algorithms have the benefit of relying purely on local feedback to perform updates, but require a good deal of computational overhead in estimating and communicating operating points as well as computing the coupling costs. In addition, their information requirements are higher, as each processor must have prior knowledge of the topology to compute its update. The KW algorithms require no models and no communications, but must rely on an expensive network-wide decision process to sample the cost and perform updates. From the point of view of establishing a distributed computation-based modeling framework for exploring organizational issues, the WIN algorithms are more interesting, as they incorporate the notions of local models and communication.

Numerical experiments indicated that both classes of algorithms performed reasonably. Average descent of the cost was observed for all of the algorithms, although several appeared to be extremely slow and required many network measurements. The best performance[9] was achieved for WIN using large numbers of estimation trials in computing the cost estimates, for the two-sided KW technique, and for the Gauss-Seidel variants WIN-GS and KW-GS. Updates for the WIN algorithms were observed to be significantly more effective than updates for the KW algorithms.

[9]fastest observed rate of convergence as a function of number of updates

Chapter 6

Convergence Analysis

In this chapter, we present proofs of asymptotic convergence for all of the one-dimensional iterative stochastic optimization algorithms of Chapter 4, as well as the distributed multivariable network algorithms of Chapter 5, with the exception of the Gauss-Seidel variants WIN-GS and KW-GS, which are more easily handled as special cases of the asynchronous algorithms covered in Chapter 7. Specifically, the development here covers WIN, WIN-BP, KW, KW-RD, and KW-SP. We assume that the network algorithms obey all the communication and synchronization assumptions introduced at the beginning of Chapter 5, which effectively masked the distributed nature of the updates. We analyze them here as being mathematically equivalent to their centralized versions.

For the development of this chapter we will denote the cost with the general notation J, intended to represent either the probability of error or the Bayes risk. We are concerned with the convergence of the sequence of random parameter vectors $\{\underline{\Theta}_k\}$ as well as the associated sequence of random scalar costs $\{J(\underline{\Theta}_k)\}$. We must accordingly adopt a probabilistic notion of convergence[1]. In this report, we focus on demonstrating convergence with probability one (w.p.1), also referred to as almost sure convergence (a.s.) or strong convergence. The results we prove are asymptotic, meaning that the convergence is guaranteed only in the limit as infinitely many iter-

[1] A discussion of the various notions of probabilistic convergence may be found in Appendix A, Section A.2.

ations are performed.

For the single DM training algorithms, we will be able to specify reasonable conditions which are sufficient to guarantee (a.s.) convergence of both the sequence of costs $\{J(\underline{\Theta}_k)\}$ and the sequence of parameters $\underline{\Theta}_k$ to the unique global minimum for a large class of conditional densities, namely conditional densities with a single point of intersection.

For the team problem, things are more complicated. It is relatively straightforward to demonstrate convergence of the sequence of costs. However, in the face of our inability to guarantee unimodality of the team cost, the best we can hope for is a result which indicates that every limit point of the sequence $\{\underline{\Theta}_k\}$ is a stationary point (a.s.). This statement parallels the type of results which may be obtained for deterministic gradient techniques [51], [5]. Notice that this statement does not imply convergence of the sequence of parameters, and does not guarantee that all limit points are minima. The weakness of the statement is not a result of any deficiency in our method of proof, but rather represents a standard difficulty with gradient-based optimization in general, both deterministic and stochastic. However, if there is assumed to be a unique stationary point of the team cost corresponding to a global minimum, then under mild conditions we again obtain probability one convergence to the global minimum. Thus, the training algorithms perform well on "nice" problems. Numerical experiments have suggested that the team Gaussian detection may possess unimodal cost, although this fact is elusive to prove. Thus, in practice we may be observing convergence to the optimum thresholds for this case.

We should emphasize that simply establishing the fact of convergence of an algorithm is no assurance that the algorithm will prove practically useful in solving a given problem. The convergence of the algorithm may be impractically slow. It is nevertheless useful to demonstrate convergence, because if this fact cannot be established, then the utility of the algorithm must be seriously questioned.

6.1 Philosophy of Proof Method

The method of proof we adopt is referred to as the descent approach and corresponds to Lyapunov's Second (Direct) Method. It is well known in the fields of iterative nonlinear and stochastic optimization [6], [11], [14], [40], [51], [50], [49]. The essential idea is to monitor the sequence of costs $\{J(\underline{\Theta}_k)\}$ on the iterates $\{\underline{\Theta}_k\}$. Roughly speaking, if the values of the cost on the iterates can be shown to decrease (in some suitable sense) monotonically, and if the cost is known to be bounded from below, then under certain assumptions it can be argued that the sequences of costs converge. Additional assumptions on the cost can then guarantee convergence of the sequence of parameters. Of course, these notions must be suitably adapted to handle the probabilistic nature of our setting. The primary tool for handling "monotonic" sequences of random variables is the theory of martingales and their various convergence theorems, on which we rely heavily in our proofs [15], [51], [38].

It is important to point out that we provide *sufficient* conditions for the convergence of the algorithms. We do not claim that the conditions are necessary, or are the least restrictive possible. The types of assumptions we require to prove convergence using the descent approach fall into the following categories, which we comment on briefly.

Smoothness conditions on cost function: These conditions include the restrictions that the function be continuously differentiable and bounded from below, and that its derivative obey a Lipschitz continuity condition, Since we are dealing with probability functions in this report, we also take the function itself to be bounded where convenient. We find that our proofs are made substantially simpler if we also exploit some additional properties of the cost available in our setting, namely boundedness of the first derivative (magnitude of the gradient) and second derivative (Hessian bounded as described in Proposition 3.8). These assumptions may be viewed as restrictive in terms of the generality of our results, but they are satisfied for a wide number of problems of interest in our application.

Markov property: The approach we take assumes that the process under consideration is defined by the iteration

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k, \quad k = 1, 2, \ldots \tag{6.1}$$

and is Markov, meaning that the distribution of the step $\underline{Z}_k$ depends only on $\underline{\Theta}_k$ and k. Equivalently, if we assume a deterministic initial condition $\underline{\Theta}_1 = \underline{\theta}_1$, and define the set of random variables

$$\mathcal{F}_k = \{\underline{Z}_\kappa | \kappa = 1, 2, \ldots, k-1\} \tag{6.2}$$

then the set $\mathcal{F}_k$ contains all the randomness present in the algorithm up to time k. Since this knowledge is sufficient to reconstruct the entire sample path of the algorithm up to and including the value of $\underline{\Theta}_k$, we may view the set $\mathcal{F}_k$ as a representation of the entire past history of the algorithm up to the moment that the update $\underline{Z}_k$ is to be generated. The Markov assumption then implies the statement

$$E\{\underline{Z}_k | \mathcal{F}_k\} = E\{\underline{Z}_k | \underline{\Theta}_k\} \tag{6.3}$$

In our setting, the validity of this assumption is assured by the IID measurement sequence. If the measurement sequence contains correlation across time, the Markov assumption no longer holds. This assumption is sufficient for us to invoke the machinery on martingales that we require.

Step direction and magnitude: The conditions we impose on the update steps will be the central focus of our arguments.

Descent conditions are familiar from deterministic nonlinear programming [51]. For an iteration of the form

$$\underline{\theta}_{k+1} = \underline{\theta}_k - \rho_k \underline{s}_k, \; k = 1, 2, \ldots \tag{6.4}$$

$\underline{s}_k$ is a descent step if for $\nabla J(\underline{\theta}_k) \neq 0$ it holds that

$$\nabla J(\underline{\theta}_k)^T \underline{s}_k > 0 \tag{6.5}$$

Intuitively, the condition means that the vector $\underline{s}_k$ makes a strictly acute angle with the gradient (i.e., strictly within $\pi/2$), which because of the nonnegativity of the sequence $\{\rho_k\}$ and the minus sign in algorithm (6.4), means that the parameter vector $\underline{\theta}_k$ is updated in the direction of the negative gradient. Since by Taylor's theorem

$$J(\underline{\theta}_{k+1}) = J(\underline{\theta}_k - \rho_k \underline{s}_k) = J(\underline{\theta}_k) - \rho_k \nabla J(\underline{\theta}_k)^T \underline{s}_k + o(\|\rho_k \underline{s}_k\|^2) \tag{6.6}$$

for a choice of ρ_k sufficiently small it follows that

$$J(\underline{\theta}_{k+1}) < J(\underline{\theta}_k) \tag{6.7}$$

Hence the term descent. If the cost is known to be bounded below, we can hope to reduce it all the way to its minimum value.

In the stochastic setting, where the $\underline{\Theta}_k$ are random vectors evolving according to stochastic difference equations, these ideas must be suitably modified. The analogous condition on the step direction takes the form of a bound involving the first moment of the step, conditioned on the past, and ensures that the steps are stochastic descent steps, or that the expected step direction (conditioned on the past of the algorithm) is in the direction of the negative gradient. The first such condition was introduced by Polyak and Tsypkin in [50], with steps obeying the condition termed *pseudogradient* steps. Specifically, an algorithm of the form

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k, \quad k = 1, 2, \ldots \tag{6.8}$$

was termed a pseudogradient algorithm if the steps $\underline{Z}_k$ obeyed the condition

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} \geq 0 \ \ (a.s.) \tag{6.9}$$

The condition means that the *average* step direction is a descent direction. Using this condition and the previously mentioned Markov assumption, the cost can be shown to obey

$$E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\} \leq J(\underline{\Theta}_k) \tag{6.10}$$

In other words, the cost is a nonnegative[2] supermartingale, the convergence of which can be argued as in Appendix A. The algorithms considered in this report do *not* obey this condition, but a related less restrictive variant. We will refer to this condition as *generalized stochastic descent*, to be defined momentarily. The generalized stochastic descent condition results in a modified form of (6.10) which can be argued to converge by a suitable extension of the supermartingale convergence theorem (Appendix A).

A companion condition on the step magnitude is also required, expressed in the form of a bound on the second moment of the steps, conditioned on the past. This condition bounds the growth of the step variance. For example, the bound originally presented by Polyak and Tsypkin [50] was

$$E\{\|\underline{Z}_k\|^2|\underline{\Theta}_k\} \leq \nu_k + K_1 J(\underline{\Theta}_k) + K_2 \nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k|\underline{\Theta}_k\} \tag{6.11}$$

where K_1, K_2 are nonnegative constants and ν_k is a positive real-valued sequence which may grow to infinity. The condition means that $E\{\|\underline{Z}_k\|^2|\underline{\Theta}_k\}$ as a function of the iteration number k does not grow faster that some sequence $\{\nu_k\}$, and as a function of the parameter vector $\underline{\Theta}_k$ does not increase more rapidly than $J(\underline{\Theta}_k)$ or $\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k|\underline{\Theta}_k\}$. The steplength condition we adopt will be similar in form to this one.

It may be necessary to impose assumptions on particular methods of step generation in order to ensure that these conditions hold. For example, appropriate conditions on the window functions are required to prove convergence of WIN algorithms using this approach.

Sequences $\{\rho_k\}$, $\{\delta_k\}$: We will require certain conditions on the real-valued step-

[2]by virtue of being a probability function

size sequence $\{\rho_k\}$, as well as the perturbation/window-width sequence $\{\delta_k\}$ and the sequence $\{\nu_k\}$ introduced in the previous paragraph. For example, convergence of the pseudogradient algorithm described above requires the set of auxiliary stepsize conditions

$$\begin{aligned} \rho_k \geq 0, \qquad & \sum_{k=1}^{\infty} \rho_k = \infty, \\ \sum_{k=1}^{\infty} \rho_k^2 < \infty, \qquad & \sum_{k=1}^{\infty} \rho_k^2 \nu_k < \infty \end{aligned} \tag{6.12}$$

These conditions are familiar from classical stochastic approximation; the stepsize sequence must have infinite sum to be able to reach any value, but must also be made to go to zero so that the effects of noise eventually become negligible. A frequent choice is $\rho_k = 1/k$, although whether this is sufficient for convergence depends on the sequence $\{\nu_k\}$ as well.

The descent method of proof has value in several respects. In the first place, in this framework we may provide a single proof encompassing most of the algorithms of Chapters 4 and 5 [3]. Specifically, using Proposition 6.1 which follows, the proof of convergence of each algorithm is reduced to verification that the necessary assumptions on the algorithm steps hold. Such an approach was presented in [50] to establish the convergence of a wide range of iterative stochastic algorithms. However, the approach adopted here differs in that convergence of algorithms obeying a generalized descent property is directly established using an alternative variant of the martingale convergence theorem[4]. A clear advantage of the approach is that it provides a unifying conceptual framework in which to view the operation of all of the algorithms.

Secondly, this framework is well-suited to making the extensions required to demonstrate conditions under which several of these algorithms admit asynchronous implementations. Thus, this chapter also serves to lay some of the technical ground-

[3] As commented earlier we actually prefer to handle the GS versions in Chapter 7.

[4] We should point out that the idea to directly establish convergence of algorithms obeying more general step conditions is not new, and has been investigated in [49], although the development here is somewhat different.

work to be invoked in Chapter 7.

The descent approach is not the only approach to such problems. Dvoretzky [18] adopts an approach based on contraction mappings, but his results require unique stationary points and thus cannot be used here. Kushner and Clark [32], and Ljung [36] utilize the so-called ODE approach, in which the behavior of the stochastic difference equation (6.1) is approximated by the solution of a deterministic differential equation. The essence of the approach lies in the observation that the associated ODE defines the "asymptotic paths" of the stochastic recursive scheme, and if the recursive scheme converges it must be to a stable equilibrium point of the associated ODE. The approach requires the machinery of weak convergence theory. We have adopted the descent approach because we believed that it offered the most direct route to developing asynchronous results.

6.2 Main Proof: Convergence of Generalized Stochastic Descent Iterations

In this section we present the general proof of convergence encompassing our algorithms. We present two versions of the proposition as follows.

We first present the result assuming that the same time-dependent stepsize and window-width/perturbation sequences are used by every processor in updating its component. The stepsize conditions are multivariable, so that the proof applies in the case where the overall action of the algorithm is generalized stochastic descent, but where this property does not necessarily hold along each coordinate.

We then present a second version of the proof in which the time-dependent sequences are allowed to be different, and for which the generalized stochastic descent conditions are assumed to hold on a coordinate-by-coordinate basis. This is a stronger condition than that assumed to hold in the first version of the proof, so that any algorithm satisfying the conditions of this proposition also satisfies the conditions of the first. In particular, coordinate-by-coordinate GSD conditions guarantee that the overall step of the algorithm is GSD, but not vice-versa. The conditions of this propo-

sition hold for many (but not all) of our algorithms, are better suited to the analysis of distributed algorithms, and are the conditions which are necessary to prove asynchronous convergence as discussed in Chapter 7. However, the algebra in the proof is more cumbersome, and obscures some of the structure of the proof. For these reasons, we present the results separately.

In the subsequent sections of this chapter, each particular training algorithm is shown to obey these assumptions of one of the propositions of this section.

We assume that unconstrained optimization of a scalar-valued performance measure $J(\underline{\theta}) : \Re^N \mapsto \Re$ is performed using an algorithm of the form

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k, \quad k = 1, 2, \ldots \tag{6.13}$$

We make the following assumptions per the preceding discussion.

Assumption 6.1 (Cost Function)

(a) *There holds* $0 \leq J(\underline{\theta}) \leq D < \infty$ *for some* $D > 0$, *and for all* $\underline{\theta} \in \Re^N$

(b) *The function* $J(\underline{\theta})$ *is twice continuously differentiable*

(c) *The gradient of* J *is has bounded magnitude and is Lipschitz continuous. In other words, there exist constants* $0 < K_5 < \infty$ *such that* $\|\nabla J(\underline{\theta})\| < K_5$ *for all* $\underline{\theta} \in \Re^N$, *and* $L > 0$ *s.t.*

$$\|\nabla J(\underline{\theta}_1) - \nabla J(\underline{\theta}_2)\| \leq L \|\underline{\theta}_1 - \underline{\theta}_2\| \tag{6.14}$$

for all $\underline{\theta}_1, \underline{\theta}_2 \in \Re^N$

Again these are strong assumptions on the cost, excluding even quadratic functions such as $J(\underline{\theta}) = \underline{\theta}^T Q \underline{\theta}$. The assumptions are tailored to the fact that the cost being optimized is a probability function. The simplicity in our method of proof lies in its exploitation of this fact. The Lipschitz continuity of the gradient is required to invoke the descent lemma of Appendix B. The additional conditions on the Hessian referred to earlier are actually required only in the context of two-sided finite difference methods, and so will be introduced at the point they become necessary.

Assumption 6.2 (Process)

We assume iterations of the form

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k, \quad k = 1, 2, \ldots \tag{6.15}$$

with deterministic starting value $\underline{\Theta}_1 = \underline{\theta}_1$, such that the process is Markov, i.e., that the distribution of $\underline{Z}_k$ depends only on $\underline{\Theta}_k$ and k, and $\underline{Z}_k$ is independent of the previous steps $\underline{Z}_1, \ldots, \underline{Z}_{k-1}$

Verification of the following assumption represents the only place in which the properties of the training algorithms enter the discussion of convergence.

Assumption 6.3 (Generalized Stochastic Descent - GSD)

There exists a positive constant K_1, nonnegative constants K_2, K_3 and K_4, and nonnegative sequences $\{\alpha_k\}, \{\beta_k\}, \{\nu_k\}$ such that

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} \geq K_1 \alpha_k \|\nabla J(\underline{\Theta}_k)\|^2 - K_2 \beta_k \|\nabla J(\underline{\Theta}_k)\|, \quad \alpha_k, \beta_k \geq 0 \tag{6.16}$$

$$E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} \leq K_3 \|\nabla J(\underline{\Theta}_k)\|^2 + K_4 \nu_k, \quad \nu_k \geq 0 \tag{6.17}$$

hold w.p.1 for all k.

The presence of the sequences $\{\beta_k\}$, and $\{\nu_k\}$ represents the major point of departure of our assumptions with the standard ones [50],[51]. The sequence $\{\beta_k\}$ represents bias in the estimate of the gradient which arises as a result of noninfinitesimal finite difference approximations, and smoothing by window functions. For our algorithms, the bias will be a positive sequence which may be made to decay to zero at a suitable rate. The sequence $\{\nu_k\}$ will generally be a sequence going to infinity, so that we do not assume that the conditional variance remains bounded.

These conditions are very weak, in the sense that it is no longer clear that the

parameter vector will be updated in a direction of descent. In particular, if we rewrite (6.16) in the form

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k|\underline{\Theta}_k\} \geq \|\nabla J(\underline{\Theta}_k)\|(K_1\alpha_k\|\nabla J(\underline{\Theta}_k)\| - K_2\beta_k) \tag{6.18}$$

we see that unless $K_2\beta_k$ is small compared with $K_1\alpha_k\|\nabla J(\underline{\Theta}_k)\|$ we are not guaranteed to be making pseudogradient steps.

Condition (6.16) can be used to establish that

$$E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\} \leq J(\underline{\Theta}_k) + V_k \tag{6.19}$$

where V_k is a nonnegative random variable, so that J possesses an "almost supermartingale" property. Provided that V_k satisfies certain conditions, convergence can still be guaranteed.

We impose the following conditions on the update and bounding sequences, which ensure that the bias and step variance are controlled by the stepsize sequence $\{\rho_k\}$.

Assumption 6.4 (Stepsizes)

The real-valued sequences $\{\rho_k\}$, $\{\alpha_k\}$, $\{\beta_k\}$, *and* $\{\nu_k\}$ *are such that*

$$\rho_k \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k = \infty, \qquad \sum_{k=1}^{\infty} \rho_k^2 < \infty \tag{6.20}$$

$$\alpha_k \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k\alpha_k = \infty \tag{6.21}$$

$$\beta_k \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k\beta_k < \infty \tag{6.22}$$

$$\nu_k \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k^2\nu_k < \infty \tag{6.23}$$

We are now prepared to present the central proposition of this chapter, the proof

of which is modeled after the argument of Do-Tu and Installe in [14].

Proposition 6.1 (Convergence of GSD Iterations)

Let Assumptions 6.1-6.4 hold. Then, for any initial fixed starting value $\underline{\theta}_1$, the iteration

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k, \quad k = 1, 2, \ldots \tag{6.24}$$

is such that

1. $\lim_{k\to\infty} J(\underline{\Theta}_k) = J$ *(a.s.), for some random variable* J
2. $\liminf_{k\to\infty} \|\nabla J(\underline{\Theta}_k)\| = 0$ *(a.s.)*

Proof. We use (6.24) and the first-order descent lemma (Appendix B) to obtain

$$J(\underline{\Theta}_{k+1}) \leq J(\underline{\Theta}_k) - \rho_k \nabla J(\underline{\Theta}_k)^T \underline{Z}_k + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \tag{6.25}$$

Taking conditional expectations of both sides of this inequality, conditioned on $\underline{\Theta}_k$, we obtain

$$\begin{aligned} E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\} &\leq J(\underline{\Theta}_k) - \rho_k \nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k|\underline{\Theta}_k\} \\ &+ (L/2)\rho_k^2 E\{\|\underline{Z}_k\|^2|\underline{\Theta}_k\} \ (a.s.) \end{aligned} \tag{6.26}$$

Using Assumption 6.3 we obtain

$$\begin{aligned} E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\} &\leq J(\underline{\Theta}_k) - \rho_k \left(K_1\alpha_k\|\nabla J(\underline{\Theta}_k)\|^2 - K_2\beta_k\|\nabla J(\underline{\Theta}_k)\|\right) \\ &\quad +(L/2)\rho_k^2(K_3\|\nabla J(\underline{\Theta}_k)\|^2 + K_4\nu_k) \qquad (6.27) \\ &= J(\underline{\Theta}_k) - \rho_k\alpha_k K_1\|\nabla J(\underline{\Theta}_k)\|^2 + \rho_k\beta_k K_2\|\nabla J(\underline{\Theta}_k)\| \\ &\quad +\rho_k^2(L/2)K_3\|\nabla J(\underline{\Theta}_k)\|^2 + \rho_k^2\nu_k(L/2)K_4 \qquad (6.28) \\ &\leq J(\underline{\Theta}_k) + \rho_k\beta_k K_2\|\nabla J(\underline{\Theta}_k)\| + \rho_k^2(L/2)K_3\|\nabla J(\underline{\Theta}_k)\|^2 \\ &\quad +\rho_k^2\nu_k(L/2)K_4 \ (a.s.) \qquad (6.29) \end{aligned}$$

By Assumption 6.1, there exists K_5 such that

$$\|\nabla J(\underline{\Theta}_k)\| \leq K_5, \quad \forall \underline{\Theta}_k \in \Re^N \tag{6.30}$$

so that we may write

$$E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\} \leq J(\underline{\Theta}_k) + \rho_k\beta_k K_6 + \rho_k^2 K_7 + \rho_k^2\nu_k K_8 \ \ (a.s.) \tag{6.31}$$

where $K_6 = K_2K_5$, $K_7 = (L/2)K_3K_5^2$, and $K_8 = (L/2)K_4$.

Let us now define the quantities

$$X_k \triangleq J(\underline{\Theta}_k) \tag{6.32}$$

$$V_k \triangleq \rho_k\beta_k K_6 + \rho_k^2 K_7 + \rho_k^2\nu_k K_8 \tag{6.33}$$

$$\mathcal{F}_k \triangleq \{\underline{Z}_\kappa | \kappa = 1, 2, \ldots, k-1\} \tag{6.34}$$

The Markov nature of iteration (6.24) implies that $E\{J(\underline{\Theta}_k)|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\} = E\{J(\underline{\Theta}_k)|\Theta_k\}$, so that these associations allow us to express (6.31) in the form

$$E\{X_{k+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\} \leq X_k + V_k \ \ (a.s.) \tag{6.35}$$

Taking into consideration the boundedness of the cost J and the conditions of Assumption 6.4, we may invoke the following variant of the martingale convergence theorem due to MacQueen [38]. The proof may be found in Appendix A, Section A.5, and relies on the boundedness of X_k.

Lemma 6.1 (Extended Supermartingale Convergence; MacQueen)
Let $\{X_k\}$ and $\{V_k\}$ be given sequences of random variables and for each $k \geq 1$ let X_k and V_k be measurable with respect to $\mathcal{F}_k$, where $\mathcal{F}_1 \subset \mathcal{F}_2 \cdots$ is a monotonically increasing sequence of Borel Fields. Suppose each of the following conditions holds with probability one and for all k.

1. $|X_k| \leq D < \infty$ *for some* $D > 0$.

2. $V_k \geq 0$ *and* $\sum_{k=1}^{\infty} V_k < \infty$.

3. $E\{X_{k+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\} \leq X_k + V_k$

Then, the sequences of random variables $\{X_k\}$ and $\{R_k\}$, where $R_0 \stackrel{\triangle}{=} 0$ and

$$R_k \stackrel{\triangle}{=} \sum_{j=1}^{k} (X_j - E\{X_{j+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_j\}), \quad k \geq 1 \tag{6.36}$$

both converge to random variables with probability one.

Note that the V_k specified by (6.33) are not random variables after we have bounded the magnitude gradient terms, but the lemma still applies and indicates that the sequence of costs $\{J(\underline{\Theta}_k)\}$ converges with probability one; the first assertion of the proposition is thus proved.

Now, reordering (6.28) and summing both sides we obtain

$$\begin{aligned} K_1 \sum_{k=1}^{\infty} \rho_k \alpha_k \|\nabla J(\underline{\Theta}_k)\|^2 \;\leq\; & \sum_{k=1}^{\infty} (J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\}) \\ & + \; K_6 \sum_{k=1}^{\infty} \rho_k \beta_k + K_7 \sum_{k=1}^{\infty} \rho_k^2 + K_8 \sum_{k=1}^{\infty} \rho_k^2 \nu_k \;\; (a.s.) \end{aligned} \tag{6.37}$$

Invoking the MacQueen Lemma once again, the term

$$\sum_{k=1}^{\infty} (J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1})|\underline{\Theta}_k\}) \tag{6.38}$$

is bounded w.p.1, so that it follows that

$$\sum_{k=1}^{\infty} \rho_k \alpha_k \|\nabla J(\underline{\Theta}_k)\|^2 < \infty \quad (a.s.) \tag{6.39}$$

Since, by Assumption 6.4, we have that

$$\sum_{k=1}^{\infty} \rho_k \alpha_k = \infty \tag{6.40}$$

we conclude that

$$\liminf_{k \to \infty} \|\nabla J(\underline{\Theta}_k)\|^2 = 0 \quad (a.s.) \tag{6.41}$$

and the second assertion of the proposition is proved.

■

Similar analyses in the literature [14] have incorrectly concluded that

$$\lim_{k \to \infty} \|\nabla J(\underline{\Theta}_k)\|^2 = 0 \quad (a.s.) \tag{6.42}$$

or that every limit point of the sequence $\{\underline{\Theta}_k\}$ is a stationary point (w.p.1). However, under the assumptions of the proposition, it is not possible to obtain a stronger statement than liminf, as the following simple counterexample demonstrates. Assume the product

$$\rho_k \alpha_k = 1/k, \quad k = 1, 2, \ldots \tag{6.43}$$

so that condition (6.40) holds, and assume that the sequence $\|\nabla J(\underline{\Theta}_k)\|^2$ is such that

$$\|\nabla J(\underline{\Theta}_k)\|^2 = \begin{cases} 1 & \text{if } k \in \{j^2 | j = 1, 2, \ldots\} \\ 0 & \text{else} \end{cases} \tag{6.44}$$

Then clearly (6.39) holds, but the sequence $\{\|\nabla J(\underline{\Theta}_k)\|\}$ is not converging to zero with probability one[5]. The difficulty here centers on the fact that the sequence $\rho_k \alpha_k$ may be going to zero. There would be no difficulty if for some $\epsilon > 0$ it held that

[5]Although it does happen to be converging to zero in probability as discussed in Appendix A

$\rho_k \alpha_k \geq \epsilon$, $\forall k$, but unfortunately this assumption may not be made. We address these issues further momentarily.

The conclusions of the proposition are weak, mainly because of our very weak stepsize assumptions. In particular, while the proposition guarantees convergence of the sequence of cost realizations $\{J(\underline{\Theta}_k)\}$ to some finite random variable J it does *not* say that J is a minimum cost, or even that the sequences of gradient realizations or parameters converge at all.

Since we may not strengthen our stepsize conditions, we must settle for rectifying the situation through an additional boundedness assumption. If we impose the a posteriori condition that the sequence of parameters remains bounded with probability one, i.e., that

$$\|\underline{\Theta}_k\| < \infty \ (a.s.), \ \forall k \tag{6.45}$$

then progress can be made. Conditions of this kind are normally dangerous, but Kushner and Clark [32] discuss why this condition is not restrictive and can be expected to hold in most applications. For example, if it is known that the solutions of interest lie in a bounded set, then the assumption can be made to hold by modifying the original algorithm to project the iterates back into this set. Boundedness conditions of this variety are often guaranteed through bounded level set assumptions on the cost, which in this case are not applicable because the cost is bounded.

We now present two corollaries, strengthening the conclusions of Proposition 6.1. The first states that under the assumption that the sequence of parameters remains bounded with probability one, and that the function possesses a unique minimizing stationary point, convergence of both the sequence of parameters and costs to their globally optimal values is assured. The proof is based on a result of Bertsekas and Tsitsiklis [6].

Corollary 6.1 *Assume that in addition to the conditions of Proposition 6.1 it also holds that*

$$\|\underline{\Theta}_k\| < \infty \ (a.s.), \ \forall k \tag{6.46}$$

that there exists a unique vector $\underline{\Theta}^*$ *at which* J *is minimized, and that this is the unique vector at which* ∇J *vanishes. Then* $\{\underline{\Theta}_k\}$ *converges to* $\underline{\Theta}^*$ *with probability one.*

Proof. Since, by Proposition 6.1 we have that

$$\liminf_{k\to\infty} \|\nabla J(\underline{\Theta}_k)\| = 0 \ \ (a.s.) \tag{6.47}$$

we can choose a sequence of times $\{k_i\}$ along which $\{\nabla J(\underline{\Theta}_k)\}$ converges to zero with probability one. Since the sequence $\{\underline{\Theta}_k\}$ is bounded, we may restrict to a subsequence of parameter values $\underline{\Theta}_{k_{i_j}}$ which converges to some $\tilde{\underline{\Theta}}^*$ for which $\nabla J(\tilde{\underline{\Theta}}^*) = 0$. By our assumption, $\tilde{\underline{\Theta}}^* = \underline{\Theta}^*$. Since $\underline{\Theta}^*$ is a limit point of the sequence $\{\underline{\Theta}_k\}$, we conclude that $J(\underline{\Theta}^*)$ is equal to the limit of $\{J(\underline{\Theta}_k)\}$. Let $\underline{X}$ be an arbitrary limit point of $\{\underline{\Theta}_k\}$. Then $J(\underline{X})$ is a limit point of $\{J(\underline{\Theta}_k)\}$, and since the latter converges to $J(\underline{\Theta}^*)$, we conclude that $\underline{X} = \underline{\Theta}^*$. Therefore, $\underline{\Theta}^*$ is the unique limit point of $\{\underline{\Theta}_k\}$. Since the sequence $\{\underline{\Theta}_k\}$ is bounded, it converges to $\underline{\Theta}^*$.

■

Corollary 6.1 indicates that GSD algorithms can be expected to behave well on unimodal problems, such as the single DM problems of Chapter 4 and the person-by-person subproblems of each DM. Notice that we were able to circumvent the liminf problem in this case by the assumptions of a unique limit point and a bounded sequence of iterates.

The team problem is more difficult, because we cannot be sure that a single minimizing stationary point exists. We would like to draw analogous conclusions concerning our stochastic gradient algorithms to those made for deterministic gradient algorithms, namely that every limit point of $\{\underline{\Theta}_k\}$ is a stationary point with

probability one. This statement is equivalent to the conclusion

$$\lim_{k \to \infty} \|\nabla J(\underline{\Theta}_k)\| = 0 \quad (a.s.) \tag{6.48}$$

or alternatively the statement that for any $\epsilon > 0$,

$$\|\nabla J(\underline{\Theta}_k)\| > \epsilon \tag{6.49}$$

only finitely often with probability one. Equivalently, if we define the set of stationary points

$$\mathcal{D}_0 = \{\underline{\theta} | \nabla J(\underline{\theta}) = \underline{0}\} \tag{6.50}$$

we would like to claim $\underline{\Theta}_k \to \mathcal{D}_0$ $(a.s.)$, where convergence of the vector to the set is measured in terms of the distance

$$\rho(\underline{\theta}, \mathcal{D}_0) = \inf_{\underline{\theta}'} (\|\underline{\theta} - \underline{\theta}'\| | \underline{\theta}' \in \mathcal{D}_0) \tag{6.51}$$

Note that the statement that every limit point of the sequence is stationary does not imply that the sequence of parameter vectors converges. In practice, however, the sequence tends to have a unique limit point, since minima possess local domains of attraction. We note that multimodal stochastic approximation problems have been addressed by Nevel'son and Has'minskii [40] and Kushner [31].

In order to obtain such a result, we must address the difficulties introduced by the possibility that $\rho_k \alpha_k \to 0$. This same problem was faced by Kushner in [31], in which he used martingale arguments to establish that every limit point of the sequence produced by KW algorithms for optimizing functions with nonunique stationary points is stationary with probability one. The argument depends on boundedness of the sequence of parameter vectors, and relies on using compactness arguments to handle the sequence $\rho_k \alpha_k$ over intervals. It also requires describing the iteration

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k \tag{6.52}$$

in more detail, which we will subsequently verify holds for all of our algorithms. We thus adopt Kushner's proof [31] to obtain the following corollary.

Corollary 6.2 (Stationarity of Limit Points (a.s.) for GSD Iterations)
Assume that in addition to the conditions of Proposition 6.1 it also holds that

$$\|\underline{\Theta}_k\| < \infty \ (a.s.), \ \forall k \tag{6.53}$$

and that the iteration (6.52) can be expressed in the form

$$\underline{\Theta}_{k+1} - \underline{\Theta}_k = -\rho_k \alpha_k \nabla J(\underline{\Theta}_k) + \underline{\gamma}_k \tag{6.54}$$

where

$$\lim_M \sup_N \left\| \sum_M^N \underline{\gamma}_n \right\| = 0 \ \ (a.s.) \tag{6.55}$$

Then, in addition to the conclusions of Proposition 6.1

$$\nabla J(\underline{\Theta}_k) \to \underline{0} \ \ (a.s.) \tag{6.56}$$

$$\underline{\Theta}_k \to \mathcal{D}_0 \ \ (a.s.) \tag{6.57}$$

where $\mathcal{D}_0 = \{\underline{\theta} | \nabla J(\underline{\theta}) = \underline{0}\}$. *In other words, for any* $\epsilon > 0$,

$$\|\nabla J(\underline{\Theta}_k)\| > \epsilon \tag{6.58}$$

only finitely often with probability one.

Proof. In the proof of Proposition 6.1 we showed that

$$E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\} - J(\underline{\Theta}_k) \leq -\rho_k \alpha_k K_1 \|\nabla J(\underline{\Theta}_k)\|^2 + V_k \tag{6.59}$$

where

$$\sum_{k=1}^{\infty} V_k < \infty, \quad E\{\sum_{k=1}^{\infty} V_k\} < \infty \tag{6.60}$$

since V_k was deterministic. This implies

$$E\{J(\underline{\Theta}_{k+1})\} - E\{J(\underline{\Theta}_1)\} \leq -\sum_{i=1}^{k} \rho_i \alpha_i K_1 E\{\|\nabla J(\underline{\Theta}_i)\|^2\} + E\{\sum_{i=1}^{k} V_i\} \tag{6.61}$$

Let the sets $\mathcal{D}_1$ and $\mathcal{D}$ denote any compact sets with the properties that $\mathcal{D} \subset \mathcal{D}_1$ and

(i) ($\mathcal{D}$ is strictly interior to $\mathcal{D}_1$) $\inf_{\underline{\Theta}_1 \in \mathcal{D}, \underline{\Theta}_2 \notin \mathcal{D}_1} \|\underline{\Theta}_1 - \underline{\Theta}_2\| \equiv d_0 > 0$

(ii) ($\mathcal{D}_1$ does not intersect $\mathcal{D}_0$) $\inf_{\underline{\Theta} \in \mathcal{D}_1} \|\nabla J(\underline{\Theta})\|^2 \equiv d_1 > 0$

(iii) $\sup_{\underline{\Theta} \in \mathcal{D}_1} \|\nabla J(\underline{\Theta})\| \equiv d_2 < \infty$

The proof proceeds according to the following strategy. It will be shown that $\{\underline{\Theta}_k\}$ is in the set $\mathcal{D}$ only finitely often with probability one, for each compact $\mathcal{D}$ for which there is a compact $\mathcal{D}_1 \supset \mathcal{D}$ so that the sets $\mathcal{D}$ and $\mathcal{D}_1$ satisfy (i)-(iii). By the with-probability-one boundedness of $\{\underline{\Theta}_k\}$, and the fact that $\|\nabla J(\underline{\Theta}_k)\|^2$ is positive and continuous in $\Re^N - \mathcal{D}_0$, the conclusions of the corollary will follow. (Indeed, if $\{\underline{\Theta}_k\}$ is not required to be bounded, and $\{\underline{\Theta}_k\} \not\to \mathcal{D}_0$, then it will follow that $\|\underline{\Theta}_k\| \to \infty$). The main difficulty in the proof is due to the possibility that $\rho_k \alpha_k \to 0$. Thus, it is necessary to estimate the "length" of stay of each visit to the sets $\mathcal{D}_1$ or $\mathcal{D}$.

Define the sequences $\{m_i'\}$ and $\{m_i\}$ by

$$\begin{aligned} m_i &= \inf\{k : k \geq m_{i-1}', \underline{\Theta}_k \in \mathcal{D}\}, \ i = 1, 2, \ldots \\ m_i' &= \inf\{k : k \geq m_i, \underline{\Theta}_k \notin \mathcal{D}_1\}, \ i = 1, 2, \ldots \\ m_1' &= 1 \end{aligned} \tag{6.62}$$

Thus, m_i is essentially the index of the ith entry of $\underline{\Theta}_k$ into $\mathcal{D}$, after being out of $\mathcal{D}_1$ $i-1$ times. m_i is finite (i.e., defined) on an ω-set Ω_i''. m_i' is essentially the earliest time of exit from $\mathcal{D}_1$ following an entry into set $\mathcal{D}$. Observe that the theorem follows if it is shown that $m_i < \infty$ only finitely often with probability one.

Define

$$L_i = \sup_k \left\| \sum_{m_i}^{m_i+k} \underline{\gamma}_j \right\| \tag{6.63}$$

Then $L_i \to 0$, $(a.s.)$ (by assumption). Now,

$$\begin{aligned} -\infty &< \liminf_{k\to\infty} (E\{J(\underline{\Theta}_k)\} - E\{J(\underline{\Theta}_1)\}) \\ &\leq -E\left\{\sum_{i=1}^{\infty} \rho_i \alpha_i K_1 \|\nabla J(\underline{\Theta}_i)\|^2\right\} + E\left\{\sum_{i=1}^{\infty} V_i\right\} \\ &\leq -E\left\{\sum_i d_1 \sum_{j=m_i}^{m_i'-1} \rho_j \alpha_j I_{m_i<\infty} I_{L_i<d_0/2}\right\} + E\left\{\sum_{i=1}^{\infty} V_i\right\} \end{aligned} \tag{6.64}$$

where the notation I_A denotes the indicator function of set A. The third line follows by counting (and bounding) the random strings between the m_i and $m_i' - 1$, for which $m_i < \infty$ and $L_i < d_0/2$ (i.e., where the effects of the term $\underline{\gamma}_i$ alone are insufficient to move $\underline{\Theta}_k$ by more than half the distance between $\mathcal{D}_1$ and $\mathcal{D}$ in the interval $[m_i, m_i' - 1]$).

Suppose $\underline{\Theta}_{m_i} \in \mathcal{D}$ (i.e., $m_i < \infty$). Then, as j increases, $\underline{\Theta}_{m_i+j}$ must leave $\mathcal{D}_1$ eventually, or else the ith term in (6.64) would be infinite. Thus $m_i' < \infty$ if $m_i < \infty$ with probability one. Let $L_i < d_0/2$. Then, by (6.54), if $\underline{\Theta}_{m_i'}$ is to be outside of $\mathcal{D}_1$ for a finite m_i', it is required *at least* that

$$\left\| \sum_{m_i}^{m_i'-1} \rho_j \alpha_j \nabla J(\underline{\Theta}_j) \right\| \geq d_0/2 \tag{6.65}$$

But then by (iii),

$$d_2 \sum_{m_i}^{m_i'-1} \rho_j \alpha_j \geq \sum_{m_i}^{m_i'-1} \rho_j \alpha_j \|\nabla J(\underline{\Theta}_j)\| \geq d_0/2 \tag{6.66}$$

so that

$$\sum_{m_i}^{m_i'-1} \rho_j \alpha_j \geq d_0/2d_2 > 0 \tag{6.67}$$

Then (6.64) yields

$$-\infty < -\frac{d_1 d_0}{2d_2} E\left\{\sum_i I_{m_i<\infty} I_{L_i<d_0/2}\right\} + E\left\{\sum_{i=1}^{\infty} V_i\right\} \tag{6.68}$$

Since $L_i \to 0$ $(a.s.)$, it is necessary that $m_i < \infty$ only finitely often with probability one if the RHS is to be finite.

■

6.2.1 Convergence Under Componentwise GSD Conditions

In this section, we allow each component to be updated with possibly different stepsize and perturbation/window width sequences. This means they may be generated by different rules, but operating on the same underlying time sequence $\{k\}$. For example, we are interested in considering at processors i and j the distinct stepsize rules

$$\rho_k^i = \rho_1^i / k^{a_i}, \qquad \rho_k^j = \rho_1^j / k^{a_j} \tag{6.69}$$

where ρ_1^i and ρ_1^j are initial stepsizes, and a_i and a_j are constant exponents. Incorporating this additional generality is relevant, not only because it suits the distributed algorithms which concern us, but also because allowing different sequences to update each component may have favorable impact on convergence in the algorithms we consider.

We now assume an iteration of the form

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \sum_{i=1}^{N} (\rho_k^i Z_{i(k)}) \underline{e}_i, \quad k = 1, 2, \ldots \tag{6.70}$$

where $\underline{e}_i$ is the unit vector along the ith coordinate, so that at each time k the ith component is updated according to

$$\Theta_{i(k+1)} = \Theta_{i(k)} - \rho_k^i Z_{i(k)}, \quad k = 1, 2, \ldots \tag{6.71}$$

Assumption 6.1 remains unchanged from the previous section, Assumption 6.2 is the same if the definition of the iteration (6.15) is replaced with (6.70). The following two assumptions are intended to replace Assumptions 6.3 and 6.4.

The first assumption states that the step along each coordinate represents generalized stochastic descent with respect to the corresponding true partial derivative.

Assumption 6.5 (GSD Along Each Coordinate)

There exists a positive constant K_1^i, nonnegative constants K_2^i, K_3^i and K_4^i, and nonnegative sequences $\{\alpha_k^i\}, \{\beta_k^i\}, \{\nu_k^i\}$ for each coordinate $i = 1, \ldots, N$ such that

$$\frac{dJ}{d\theta_i}(\underline{\Theta}_k) E\{Z_{i(k)} | \underline{\Theta}_k\} \geq K_1^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 - K_2^i \beta_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|, \quad \alpha_k^i, \beta_k^i \geq 0 \quad (6.72)$$

$$E\{|Z_{i(k)}|^2 | \underline{\Theta}_k\} \leq K_3^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 + K_4^i \nu_k^i, \quad \nu_k^i \geq 0$$

hold w.p.1 for all k and all i.

The second assumption allows for different choice of stepsizes at each processor, and defines restrictions on the sequences α_k^i, β_k^i, and ν_k^i used in the specification of the coordinatewise GSD conditions.

Assumption 6.6 (Stepsizes Along Each Component)

The real-valued sequences $\{\rho_k^i\}$, $\{\alpha_k^i\}$, $\{\beta_k^i\}$, and $\{\nu_k^i\}$ are such that for all $i = 1, \ldots, N$

$$\rho_k^i \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k^i = \infty, \qquad \sum_{k=1}^{\infty} (\rho_k^i)^2 < \infty \quad (6.73)$$

$$\alpha_k^i \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k^i \alpha_k^i = \infty \quad (6.74)$$

$$\beta_k^i \geq 0, \qquad \sum_{k=1}^{\infty} \rho_k^i \beta_k^i < \infty \quad (6.75)$$

$$\nu_k^i \geq 0, \qquad \sum_{k=1}^{\infty} (\rho_k^i)^2 \nu_k^i < \infty \quad (6.76)$$

Proposition 6.2 (Convergence of Componentwise GSD Iterations)

Let Assumptions 6.1-6.2 and 6.5-6.6 hold. Then, for any initial fixed starting value $\underline{\theta}_1$, *the iteration*

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \sum_{i=1}^{N} (\rho_k^i Z_{i(k)}) \underline{e}_i, \quad k = 1, 2, \ldots \tag{6.77}$$

is such that

1. $\lim_{k\to\infty} J(\underline{\Theta}_k) = J$ *(a.s.), for some random variable* J

2. $\liminf_{k\to\infty} \|\nabla J(\underline{\Theta}_k)\| = 0$ *(a.s.)*

Proof. We use (6.77) and the first-order descent lemma (Appendix B) to obtain

$$J(\underline{\Theta}_{k+1}) \leq J(\underline{\Theta}_k) - \sum_{i=1}^{N} \rho_k^i \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) Z_{i(k)} + (L/2) \sum_{i=1}^{N} (\rho_k^i)^2 |Z_{i(k)}|^2 \tag{6.78}$$

Taking conditional expectations of both sides of this inequality, conditioned on $\underline{\Theta}_k$, we obtain

$$\begin{aligned} E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\} &\leq J(\underline{\Theta}_k) - \sum_{i=1}^{N} \rho_k^i \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) E\{Z_{i(k)} | \underline{\Theta}_k\} \\ &+ (L/2) \sum_{i=1}^{N} (\rho_k^i)^2 E\{|Z_{i(k)}|^2 | \underline{\Theta}_k\} \quad (a.s.) \end{aligned} \tag{6.79}$$

Using Assumption 6.5 we obtain

$$\begin{aligned} E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\} &\leq J(\underline{\Theta}_k) - \sum_{i=1}^{N} \rho_k^i \left(K_1^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 - K_2^i \beta_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right| \right) \\ &+ (L/2) \sum_{i=1}^{N} (\rho_k^i)^2 \left(K_3^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 + K_4^i \nu_k^i \right) \end{aligned} \tag{6.80}$$

$$\begin{aligned} &= J(\underline{\Theta}_k) - \sum_{i=1}^{N} \rho_k^i \alpha_k^i K_1^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 + \sum_{i=1}^{N} \rho_k^i \beta_k^i K_2^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right| \\ &+ (L/2) \sum_{i=1}^{N} (\rho_k^i)^2 K_3^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 + (L/2) \sum_{i=1}^{N} (\rho_k^i)^2 \nu_k^i K_4^i \end{aligned} \tag{6.81}$$

$$\leq J(\underline{\Theta}_k) + \sum_{i=1}^{N} \rho_k^i \beta_k^i K_2^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right| + (L/2) \sum_{i=1}^{N} (\rho_k^i)^2 K_3^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2$$
$$+(L/2) \sum_{i=1}^{N} (\rho_k^i)^2 \nu_k^i K_4^i \ \ (a.s.) \tag{6.82}$$

By Assumption 6.1, there exists K_5 such that

$$\left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right| \leq \|\nabla J(\underline{\Theta}_k)\| \leq K_5, \ \ \forall \underline{\Theta}_k \in \Re^N \tag{6.83}$$

If we define

$$\begin{aligned} K_2 &= \max_i \{K_2^i;\ i = 1, \ldots, N\} \\ K_3 &= \max_i \{K_3^i;\ i = 1, \ldots, N\} \\ K_4 &= \max_i \{K_4^i;\ i = 1, \ldots, N\} \end{aligned} \tag{6.84}$$

we may write

$$E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\} \leq J(\underline{\Theta}_k) + K_6 \sum_{i=1}^{N} \rho_k^i \beta_k^i + K_7 \sum_{i=1}^{N} (\rho_k^i)^2 + K_8 \sum_{i=1}^{N} (\rho_k^i)^2 \nu_k^i \ \ (a.s.) \tag{6.85}$$

where $K_6 = K_2 K_5$, $K_7 = (L/2) K_3 K_5^2$, and $K_8 = (L/2) K_4$.

Let us now define the quantities

$$X_k \triangleq J(\underline{\Theta}_k) \tag{6.86}$$

$$V_k \triangleq K_6 \sum_{i=1}^{N} \rho_k^i \beta_k^i + K_7 \sum_{i=1}^{N} (\rho_k^i)^2 + K_8 \sum_{i=1}^{N} (\rho_k^i)^2 \nu_k^i \tag{6.87}$$

$$\mathcal{F}_k \triangleq \{\underline{Z}_\kappa | \kappa = 1, 2, \ldots, k-1\} \tag{6.88}$$

Note that by Assumption 6.6

$$\begin{aligned} \sum_{k=1}^{\infty} V_k &= \sum_{k=1}^{\infty} \left(K_6 \sum_{i=1}^{N} \rho_k^i \beta_k^i + K_7 \sum_{i=1}^{N} (\rho_k^i)^2 + K_8 \sum_{i=1}^{N} (\rho_k^i)^2 \nu_k^i \right) \\ &= K_6 \sum_{i=1}^{N} \sum_{k=1}^{\infty} \rho_k^i \beta_k^i + K_7 \sum_{i=1}^{N} \sum_{k=1}^{\infty} (\rho_k^i)^2 + K_8 \sum_{i=1}^{N} \sum_{k=1}^{\infty} (\rho_k^i)^2 \nu_k^i \\ &\leq \infty \end{aligned} \tag{6.89}$$

so that we may invoke the MacQueen Lemma to argue that the sequence of costs $\{J(\underline{\Theta}_k)\}$ converges with probability one; the first assertion of the proposition is thus proved.

Now, defining

$$K_1 = \min_i \{K_1^i;\ i = 1, \ldots, N\} \tag{6.90}$$

reordering (6.81) and summing both sides we obtain

$$K_1 \sum_{k=1}^{\infty} \left(\sum_{i=1}^{N} \rho_k^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 \right) \leq \sum_{k=1}^{\infty} (J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\}) \tag{6.91}$$

$$+K_6 \sum_{k=1}^{\infty} \left(\sum_{i=1}^{N} \rho_k^i \beta_k^i \right) + K_7 \sum_{k=1}^{\infty} \left(\sum_{i=1}^{N} (\rho_k^i)^2 \right)$$

$$+K_8 \sum_{k=1}^{\infty} \left(\sum_{i=1}^{N} (\rho_k^i)^2 \nu_k \right) \quad (a.s.) \tag{6.92}$$

and reordering the sums we obtain

$$K_1 \sum_{i=1}^{N} \left(\sum_{k=1}^{\infty} \rho_k^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 \right) \leq \sum_{k=1}^{\infty} (J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\}) \tag{6.93}$$

$$+K_6 \sum_{i=1}^{N} \left(\sum_{k=1}^{\infty} \rho_k^i \beta_k^i \right) + K_7 \sum_{i=1}^{N} \left(\sum_{k=1}^{\infty} (\rho_k^i)^2 \right)$$

$$+K_8 \sum_{i=1}^{N} \left(\sum_{k=1}^{\infty} (\rho_k^i)^2 \nu_k \right) \quad (a.s.) \tag{6.94}$$

Invoking the MacQueen Lemma once again, the term

$$\sum_{k=1}^{\infty} (J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1}) | \underline{\Theta}_k\}) \tag{6.95}$$

is bounded w.p.1, so that it follows that

$$\sum_{i=1}^{N} \left(\sum_{k=1}^{\infty} \rho_k^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 \right) < \infty \ (a.s.) \tag{6.96}$$

which implies that

$$\sum_{k=1}^{\infty} \rho_k^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 < \infty \ \ (a.s.) \text{ and for all } i \tag{6.97}$$

Since, by Assumption 6.4, we have that

$$\sum_{k=1}^{\infty} \rho_k^i \alpha_k^i = \infty, \ \ \forall i \tag{6.98}$$

we conclude that

$$\liminf_{k \to \infty} \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right|^2 = 0 \ \ (a.s.) \text{ and for all } i \tag{6.99}$$

so that

$$\liminf_{k \to \infty} \|\nabla J(\underline{\Theta}_k)\|^2 = 0 \ \ (a.s.) \tag{6.100}$$

and the second assertion of the proposition is proved.

■

Corollaries 6.1, 6.2 follow identically as before.

6.3 Convergence of the Training Algorithms

In this section, we relate each of the training algorithms to the general framework we have just established. Since we have presented a general proof of convergence encompassing all classes of algorithms which obey Assumptions 6.1 - 6.4, the problem of proving convergence of the training algorithms has been reduced to the problem of verifying that the training algorithms actually satisfy the required conditions.

It is through Assumption 6.1 that the statistical properties of the underlying hypothesis test are constrained, because it is directly from the properties of the conditional densities that the properties of the cost derive. Hence the relevance of the discussion in Chapter 3. We can be sure that, at least in the case of the Gaussian detection problem, these conditions are satisfied. Notice that although our methods are statedly nonparametric, establishing sufficient conditions for convergence requires that we know the statistics, or equivalently the cost function deriving from the statis-

tics, at least well enough to verify Assumption 6.1. The validity of Assumption 6.2, the Markov property of the iterations, is guaranteed for every algorithm by our assumption of an IID measurement sequence. Assumptions 6.1 and 6.2 represent aspects of the problem which are beyond the optimizor's control. Their satisfaction depends fundamentally on the problem setup.

The allowable form of the stepsize sequence $\{\rho_k\}$ will be dictated by the exact form of the sequences $\{\beta_k\}$ and $\{\nu_k\}$ which arise in the context of a particular algorithm. But for the training algorithms of this report we will always be able to make a choice of $\{\rho_k\}$ which assures convergence. We will present these conditions by assuming ρ_k is of the form $\rho_k = \rho_1/k^a$ where a is a positive constant, and then express the allowable stepsizes as an allowable range of a.

The primary thing which remains to be proved then, is that the various schemes suggested in Chapters 4 and 5 for generating the steps $\underline{Z}_k$ result in steps for which the Generalized Stochastic Descent Assumption holds.

6.3.1 WIN Algorithms

Our strategy with the window algorithms will be to consider the one-dimensional and multidimensional cases separately. The reason for this is that we will show that the conditions for the one-dimensional case will hold componentwise for the multidimensional case.

Single DM Case: Constant λ_0, λ_1

We will proceed with the analysis in the following steps. We first establish the functional relationship between the derivative approximation created by the window technique and the true derivative, in the process providing an alternative interpretation of the windowing action. We then make certain restrictions on the window functions and develop an analytic bound on the bias present in the gradient measurement as a result of windowing. Finally, using the bound on the bias, we demonstrate that the window steps are a special case of the algorithm considered in the previous section.

We develop the results here for the general cost problem of optimizing the function

$J_B : \Re \mapsto \Re$ given by

$$J_B(\theta) = \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \tag{6.101}$$

for which

$$P_\epsilon(\theta) = J_B(\theta)\Big|_{\lambda_0=\lambda_1=1} \tag{6.102}$$

We begin our analysis with the unnormalized variant of the algorithm we first introduced in Chapter 4, and then consider the normalized variant. Recall from the discussion in Section 4.3.1 that the WIN algorithm was easily adapted to handle unequal bounded real-valued costs λ_0, λ_1 on the two types of errors by defining the labeling random variable

$$\tilde{Q}_B(X, \lambda_0, \lambda_1) = \begin{cases} +\lambda_1 & \text{if } H = H_1 \\ -\lambda_0 & \text{if } H = H_0 \end{cases} \tag{6.103}$$

Then we may express

$$\begin{aligned} J_B(\theta) &= \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 \int_{-\infty}^\theta p_{Y|H_1}(y|H_1)\,dy \\ &= \lambda_0 p_0 \int_\theta^\infty p_{Y|H_0}(y|H_0)\,dy + \lambda_1 p_1 (1 - \int_\theta^\infty p_{Y|H_1}(y|H_1)\,dy) \\ &= \lambda_1 p_1 - E_X\{\tilde{Q}_B u(Y - \Theta)|\Theta = \theta\} \end{aligned} \tag{6.104}$$

We can define the related modified performance index by replacing the hard-limiting threshold function $u(\cdot)$ with an approximation $\hat{u}(\cdot)$ as discussed in Chapter 4, and then defining

$$\hat{J}_B(\theta, \delta) = \lambda_1 p_1 - E_X\{\tilde{Q}_B \hat{u}(Y - \Theta, \delta)|\Theta = \theta\} \tag{6.105}$$

for which the derivative is given by

$$\frac{\partial \hat{J}_B}{\partial \theta}(\theta, \delta) = E_X\{\tilde{Q}_B h(Y - \Theta, \delta)|\Theta = \theta\} \tag{6.106}$$

where $\hat{u}$ and h are related as

$$\hat{u}(y-\theta,\delta) = \int_{-\infty}^{y-\theta} h(s,\delta)\,ds \tag{6.107}$$

We now establish the relationship of the modified performance measure $\hat{J}_B(\theta,\delta)$ and its derivative to the true performance measure $J_B(\theta)$ and its derivative, in particular establishing the validity of equations (4.41) and (4.43). The result implies that the modified performance measure and its derivative can be interpreted as smoothed versions of the true performance measure and derivative. Establishing this fact facilitates demonstration of the bound on the bias in the window estimate of the derivative. The proof uses the same approach as that of [14].

Lemma 6.2 (Relationship between Modified and True Performance Measures)

It holds that

(a) $\hat{J}_B(\theta,\delta) = \int_{-\infty}^{\infty} J_B(s+\theta)h(s,\delta)\,ds$

(b) $\frac{\partial \hat{J}_B}{\partial \theta}(\theta,\delta) = \int_{-\infty}^{\infty} \frac{dJ_B}{d\theta}(s+\theta)h(s,\delta)\,ds$

Proof. (a) Using equations (6.105) and (6.107) and the fact that

$$\int_{-\infty}^{\infty} h(s,\delta)\,ds = 1,\ \forall\,\delta > 0 \tag{6.108}$$

we obtain

$$\begin{aligned}
\hat{J}_B(\theta,\delta) &= \lambda_1 p_1 - E_X\{\tilde{Q}_B \hat{u}(Y-\Theta,\delta)|\Theta=\theta\} \\
&= \lambda_1 p_1 - E_{Y,H}\left\{\tilde{Q}_B(H,\lambda_0,\lambda_1)\int_{-\infty}^{Y-\Theta} h(s,\delta)\,ds \,\middle|\, \Theta=\theta\right\} \\
&= \lambda_1 p_1 - E_H\left\{E_Y\left\{\tilde{Q}_B(H=H_i,\lambda_0,\lambda_1)\cdot \right.\right. \\
&\qquad \left.\left.\int_{-\infty}^{Y-\Theta} h(s,\delta)\,ds \,\middle|\, \Theta=\theta, H=H_i\right\} \,\middle|\, \Theta=\theta\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \lambda_1 p_1 - E_H \left\{ (-1)^{i+1} \lambda_i \int_{-\infty}^{\infty} \left[\int_{-\infty}^{y-\Theta} h(s,\delta)\, ds \right] p_{Y|H_i}(y|H_i)\, dy \,\middle|\, \Theta = \theta \right\} \\
&= \lambda_1 p_1 - \left[\lambda_1 p_1 \int_{-\infty}^{\infty} \left(\int_{-\infty}^{y-\theta} h(s,\delta)\, ds \right) p_{Y|H_1}(y|H_1)\, dy \right. \\
&\quad \left. - \lambda_0 p_0 \int_{-\infty}^{\infty} \left(\int_{-\infty}^{y-\theta} h(s,\delta)\, ds \right) p_{Y|H_0}(y|H_0)\, dy \right] \\
&= \lambda_1 p_1 - \int_{-\infty}^{\infty} \left[\lambda_1 p_1 \int_{s+\theta}^{\infty} p_{Y|H_1}(y|H_1)\, dy \right] h(s,\delta)\, ds \\
&\quad + \int_{-\infty}^{\infty} \left[\lambda_0 p_0 \int_{s+\theta}^{\infty} p_{Y|H_0}(y|H_0)\, dy \right] h(s,\delta)\, ds \\
&= \int_{-\infty}^{\infty} \left[\lambda_1 p_1 - \lambda_1 p_1 \int_{s+\theta}^{\infty} p_{Y|H_1}(y|H_1)\, dy + \lambda_0 p_0 \int_{s+\theta}^{\infty} p_{Y|H_0}(y|H_0)\, dy \right] h(s,\delta)\, ds \\
&= \int_{-\infty}^{\infty} J_B(s+\theta) h(s,\delta)\, ds \qquad (6.109)
\end{aligned}
$$

(b) It immediately follows from (a) that

$$
\frac{\partial \hat{J}_B}{\partial \theta}(\theta,\delta) = \frac{\partial}{\partial \theta} \left(\int_{-\infty}^{\infty} J_B(s+\theta) h(s,\delta)\, ds \right) = \int_{-\infty}^{\infty} \frac{dJ_B}{d\theta}(s+\theta) h(s,\delta)\, ds \qquad (6.110)
$$

■

Before proceeding, we summarize all the technical conditions we require on the window function h.

Assumption 6.7 (Required Properties of Window Function [14])
It holds that

(a) $h(s,\delta) \geq 0,\ \forall \delta > 0, and\ \forall s$

(b) $h(s,\delta) = h(-s,\delta)$

(c) $\lim_{\delta \to 0} h(s,\delta) = \delta(s)$

(d) $\int_{-\infty}^{\infty} h(s,\delta)\, ds = 1,\ \forall \delta > 0$

(e) $h(s,\delta) = \frac{1}{\delta} g(s/\delta)$ *where* $g(s/\delta)$ *is any function of* s/δ *which satisfies*

$$0 \leq g(s/\delta) < \infty,\ \forall s,\ \forall \delta \geq 0 \tag{6.111}$$

(f) *There exists a constant* $B > 0$ *such that*

$$\int_{-\infty}^{\infty} \frac{|s|}{\delta} h(s,\delta)\, ds \leq B < \infty,\ \forall\, \delta > 0 \tag{6.112}$$

The symmetry assumption (b) is not strictly necessary, but there is no reason to choose an asymmetric function. Condition (e) requires that the window function $h(s,\delta)$ be within a factor $1/\delta$ of a function which is bounded for all choice of δ non-negative. Assumption (f) stipulates the existence of a constant bound, independent of δ, such that the integral (6.112) is bounded for a given choice of h for any choice of δ. The reason for requiring (f) will be clear momentarily. These conditions can be shown to hold for all of the window functions illustrated in Figure 4-4, Section 4.3.

We now proceed with the derivation of the bound on the bias present in the window estimate of the gradient.

Lemma 6.3 (Bias in Window Estimate of Derivative)

Let $J_B(\theta) : \Re \mapsto \Re$ be a differentiable function such that $\partial J_B(\theta)/\partial\theta$ is Lipschitz continuous with constant $L > 0$. Let $h(s,\delta)$ be a window function satisfying Assumption 6.7. Then, for the window function approximation of the first derivative of J_B given by

$$\frac{\partial \hat{J}_B}{\partial\theta}(\Theta,\delta) = \tilde{Q}_B(X,\lambda_0,\lambda_1)h(Y-\Theta,\delta) \tag{6.113}$$

with $\tilde{Q}_B$ defined by (6.103) it holds that

$$\left|\frac{\partial \hat{J}_B}{\partial\theta}(\theta,\delta) - \frac{dJ_B}{d\theta}(\theta)\right| \leq LB\delta \tag{6.114}$$

where B is the positive constant of Assumption 6.7(f).

Proof. Using Lemma 6.2, Assumptions 6.7(d),(f), and the Lipschitz continuity of the derivative of J_B, it follows that

$$\begin{aligned}
\frac{\partial \hat{J}_B}{\partial\theta}(\theta,\delta) &= \int_{-\infty}^{\infty} \frac{dJ_B}{d\theta}(s+\theta)h(s,\delta)\,ds \\
&= \int_{-\infty}^{\infty} \left(\frac{dJ_B}{d\theta}(s+\theta) - \frac{dJ_B}{d\theta}(\theta) + \frac{dJ_B}{d\theta}(\theta)\right) h(s,\delta)\,ds \\
&\leq \int_{-\infty}^{\infty} \left|\frac{dJ_B}{d\theta}(s+\theta) - \frac{dJ_B}{d\theta}(\theta)\right| h(s,\delta)\,ds + \int_{-\infty}^{\infty} \frac{dJ_B}{d\theta}(\theta)h(s,\delta)\,ds \\
&\leq \int_{-\infty}^{\infty} L|s|h(s,\delta)\,ds + \frac{dJ_B}{d\theta}(\theta)
\end{aligned} \tag{6.115}$$

Equation (6.115) then implies

$$\begin{aligned}
\frac{\partial \hat{J}_B}{\partial\theta}(\theta,\delta) - \frac{dJ_B}{d\theta}(\theta) &\leq L\int_{-\infty}^{\infty} |s|h(s,\delta)\,ds \\
&= L\delta\int_{-\infty}^{\infty} \frac{|s|}{\delta}h(s,\delta)\,ds \\
&\leq LB\delta
\end{aligned} \tag{6.116}$$

It may be similarly argued that

$$\begin{aligned} -\frac{\partial \hat{J}_B}{\partial \theta}(\theta, \delta) &= \int_{-\infty}^{\infty} -\frac{dJ_B}{d\theta}(s+\theta) h(s, \delta)\, ds \\ &\leq \int_{-\infty}^{\infty} \left| \frac{dJ_B}{d\theta}(\theta) - \frac{dJ_B}{d\theta}(s+\theta) \right| h(s, \delta)\, ds \\ &\quad - \int_{-\infty}^{\infty} \frac{dJ_B}{d\theta}(\theta) h(s, \delta)\, ds \end{aligned} \tag{6.117}$$

yielding

$$\begin{aligned} \frac{dJ_B}{d\theta}(\theta) - \frac{\partial \hat{J}_B}{\partial \theta}(\theta, \delta) &\leq \int_{-\infty}^{\infty} L|-s| h(s, \delta)\, ds \\ &\leq LB\delta \end{aligned} \tag{6.118}$$

Therefore

$$\left| \frac{\partial \hat{J}_B}{\partial \theta}(\theta, \delta) - \frac{dJ_B}{d\theta}(\theta) \right| \leq LB\delta \tag{6.119}$$

■

We are now in position to demonstrate the fact that the unnormalized variant of the one-dimensional window algorithm possesses the required generalized stochastic descent property.

Proposition 6.3 (GSD Property: 1D WIN, Unnormalized Variant)

For the step

$$\begin{aligned} Z_k &= \tilde{Q}_{B(k)}(X_k, \lambda_0, \lambda_1) h(Y_k - \Theta_k, \delta_k) \\ &= \begin{cases} -\lambda_0 h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_0 \\ +\lambda_1 h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{6.120}$$

it holds that

(i)

$$\frac{dJ_B}{d\theta}(\Theta_k) E\{Z_k | \Theta_k\} \geq \left| \frac{dJ_B}{d\theta}(\Theta_k) \right|^2 - LB\delta_k \left| \frac{dJ_B}{d\theta}(\Theta_k) \right| \tag{6.121}$$

(ii)

$$E\{|Z_k|^2 | \Theta_k\} \leq (p_0 B_0 \lambda_0^2 + p_1 B_1 \lambda_1^2) D_k \frac{1}{\delta_k} \tag{6.122}$$

where L is the Lipschitz constant on the gradient of J, $0 \leq B < \infty$ is the nonnegative constant assumed in Assumption 6.7(f), λ_0 and λ_1 are given fixed bounded costs, B_0 and B_1 are the bounds on the conditional densities of Proposition 3.3, and $0 \leq D_k < \infty$ is a nonnegative constant given by

$$D_k = \sup_s g(s/\delta_k) \tag{6.123}$$

where the $g(s, \delta_k)$ are the bounded functions whose existence was assumed in Assumption 6.7(e).

Proof. (i) Using (6.106) and Lemma 6.3

$$\begin{aligned} \frac{dJ_B}{d\theta}(\Theta_k) E\{Z_k | \Theta_k\} &= \frac{dJ_B}{d\theta}(\Theta_k) \frac{\partial \hat{J}_B}{\partial \theta}(\Theta_k, \delta_k) \\ &= \frac{dJ_B}{d\theta}(\Theta_k) \left(\frac{\partial \hat{J}_B}{\partial \theta}(\Theta_k, \delta_k) - \frac{dJ_B}{d\theta}(\Theta_k) + \frac{dJ_B}{d\theta}(\Theta_k) \right) \\ &= \left(\frac{dJ_B}{d\theta}(\Theta_k) \right)^2 + \frac{dJ_B}{d\theta}(\Theta_k) \left(\frac{\partial \hat{J}_B}{\partial \theta}(\Theta_k, \delta_k) - \frac{dJ_B}{d\theta}(\Theta_k) \right) \end{aligned}$$

$$\geq \left|\frac{dJ_B}{d\theta}(\Theta_k)\right|^2 - LB\delta_k \left|\frac{dJ_B}{d\theta}(\Theta_k)\right| \tag{6.124}$$

(ii) Using Assumptions 6.7(a),(e), the positivity of the probabilities p_i and conditional densities $p_{Y|H_i}(y|H_i)$, and the fact that

$$\sup_y p_i p_{Y|H_i}(y|H_i) \leq p_i B_i, \ i = 0, 1 \tag{6.125}$$

by Proposition 3.3, we obtain

$$\begin{aligned}
E\{|Z_k|^2|\Theta_k\} &= E_{X_k}\{(\tilde{Q}_{B(k)}h(Y_k - \Theta_k, \delta_k))^2|\Theta_k\} \\
&= E_{X_k}\{\tilde{Q}^2_{B(k)}h^2(Y_k - \Theta_k, \delta_k)|\Theta_k\} \\
&= E_{H^k}\{E_{Y_k}\{\lambda_i^2 h^2(Y_k - \Theta_k, \delta_k)|\Theta_k, H^k = H_i\}|\Theta_k\} \\
&= E_{H^k}\left\{\lambda_i^2 \int_{-\infty}^{\infty} h^2(y - \Theta_k, \delta_k)p_{Y|H_i}(y|H_i)\,dy \middle| \Theta_k\right\} \\
&= \lambda_0^2 \int_{-\infty}^{\infty} h^2(y - \Theta_k, \delta_k)p_0 p_{Y|H_0}(y|H_0)\,dy \\
&\quad + \lambda_1^2 \int_{-\infty}^{\infty} h^2(y - \Theta_k, \delta_k)p_1 p_{Y|H_1}(y|H_1)\,dy \\
&\leq \lambda_0^2 \frac{1}{\delta_k}(\sup_s g(s/\delta_k)) \int_{-\infty}^{\infty} h(y - \Theta_k, \delta_k)p_0 p_{Y|H_0}(y|H_0)\,dy \\
&\quad + \lambda_1^2 \frac{1}{\delta_k}(\sup_s g(s/\delta_k)) \int_{-\infty}^{\infty} h(y - \Theta_k, \delta_k)p_1 p_{Y|H_1}(y|H_1)\,dy \\
&\leq (p_0 B_0 \lambda_0^2 + p_1 B_1 \lambda_1^2)\frac{1}{\delta_k}(\sup_s g(s/\delta_k)) \\
&= (p_0 B_0 \lambda_0^2 + p_1 B_1 \lambda_1^2) D_k \frac{1}{\delta_k}
\end{aligned} \tag{6.126}$$

■

Notice that the bias term decays to zero as δ_k, while the bound on the variance goes to infinity as $1/\delta_k$.

Conditions (6.121) and (6.122) may be related to the conditions of Assumption 6.3 by making the associations

$$K_1 = 1, \quad \alpha_k = 1, \ \forall k \tag{6.127}$$

$$K_2 = L, \quad \beta_k = B_k \delta_k \tag{6.128}$$

$$K_3 = 0, \quad K_4 = (p_0 B_0 \lambda_0^2 + p_1 B_1 \lambda_1^2), \quad \nu_k = D_k \frac{1}{\delta_k} \tag{6.129}$$

Since B, B_0, B_1, D_k and the costs λ_0, λ_1 are bounded, the requirements of Assumption 6.4 dictate that the stepsize sequence $\{\rho_k\}$ and the window-width sequence $\{\delta_k\}$ be such that

$$\rho_k \geq 0, \quad \sum_{i=1}^{\infty} \rho_k = \infty, \quad \sum_{i=1}^{\infty} \rho_k^2 < \infty \tag{6.130}$$

$$\delta_k \geq 0, \quad \sum_{i=1}^{\infty} \rho_k \delta_k < \infty, \quad \sum_{i=1}^{\infty} \frac{\rho_k^2}{\delta_k} < \infty \tag{6.131}$$

If we adopt the forms

$$\rho_k = \frac{\rho_1}{k^a}, \quad \delta_k = \frac{\delta_1}{k^b} \tag{6.132}$$

then the above conditions imply the set of inequalities

$$a \leq 1, \qquad a \geq (1/2) \tag{6.133}$$

$$b \geq -a + 1, \qquad b \leq 2a - 1 \tag{6.134}$$

which define the region of allowable stepsize exponents as shown in Figure 6-1. The shaded region contains all (a, b) pairs which are sufficient for the asymptotic convergence of the algorithm. Notice that the values of numerator constants ρ_1 and δ_1 have no impact on the asymptotic convergence analysis, although their choice certainly affects the behavior of the algorithms. Choosing a *best* pair of exponents and initial stepsizes ρ_1 and δ_1 requires rate of convergence analysis of the algorithm which we do not provide. We refer the reader to [14] for further discussion of these issues.

The preceding discussion indicates that the unnormalized variant of the one-dimensional window algorithm converges by Proposition 6.1.

The normalized variant [73],[57] is similarly analyzed. Recall that the steps for this variant were defined by

$$Z_k = 2\delta_k \tilde{Q}_{B(k)}(X_k, \lambda_0, \lambda_1) h(Y_k - \Theta_k, \delta_k)$$

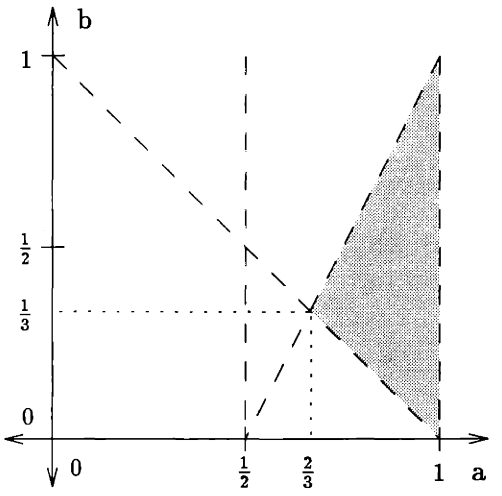


Figure 6-1: Allowable Ranges of a and b for the unnormalized WIN algorithm.

$$= \begin{cases} -2\delta_k \lambda_0 h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_0 \\ +2\delta_k \lambda_1 h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \tag{6.135}$$

where $\tilde{Q}_{B(k)}$ still defined as in (6.103), and where the additional factor of δ_k acts to normalize the magnitude of the steps. This has a favorable impact on convergence as we now argue.

The preceding analysis goes through essentially unchanged, with the additional factor of $2\delta_k$ carrying through. Thus, it follows that the modified cost is related to the true cost by

$$\hat{J}_B(\theta_k, \delta_k) = 2\delta_k \int_{-\infty}^{\infty} J_B(\theta_k + s) h(s, \delta_k)\, ds \tag{6.136}$$

with the derivatives related by

$$\begin{aligned} \frac{\partial \hat{J}_B}{\partial \theta}(\Theta_k, \delta_k) &= 2\delta_k \int_{-\infty}^{\infty} \frac{dJ_B}{d\theta}(\theta_k + s) h(s, \delta_k)\, ds \\ &= E_{X_k}\{2\delta_k \tilde{Q}_{B(k)}(X_k, \lambda_0, \lambda_1) h(Y_k - \Theta_k, \delta_k) | \Theta_k\} \\ &= E_{X_k}\{Z_k | \Theta_k\} \end{aligned} \tag{6.137}$$

Using these relations, we can establish the bias bound

$$\left|\frac{\partial \hat{J}_B}{\partial \theta}(\theta,\delta) - 2\delta\frac{dJ_B}{d\theta}(\theta)\right| \leq 2LB\delta^2 \tag{6.138}$$

from which the following proposition follows directly. Hence, we omit the proof.

Proposition 6.4 (GSD Property: 1D WIN, Normalized Variant)
For the step

$$\begin{aligned} Z_k &= 2\delta_k \tilde{Q}_{B(k)}(X_k,\lambda_0,\lambda_1)h(Y_k-\Theta_k,\delta_k) \\ &= \begin{cases} -2\lambda_0\delta_k h(Y_k-\Theta_k,\delta_k) & \text{if } H^k = H_0 \\ +2\lambda_1\delta_k h(Y_k-\Theta_k,\delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{6.139}$$

it holds that

(i)

$$\frac{dJ_B}{d\theta}(\Theta_k)E\{Z_k|\Theta_k\} \geq 2\delta_k\left|\frac{dJ_B}{d\theta}(\Theta_k)\right|^2 - 2LB\delta_k^2\left|\frac{dJ_B}{d\theta}(\Theta_k)\right| \tag{6.140}$$

(ii)

$$E\{|Z_k|^2|\Theta_k\} \leq 4(p_0B_0\lambda_0^2 + p_1B_1\lambda_1^2)D_k\delta_k \tag{6.141}$$

where L is the Lipschitz constant on the gradient of J, $0 \leq B < \infty$ is the nonnegative constant assumed in Assumption 6.7(f), λ_0 and λ_1 are given fixed bounded costs, B_0 and B_1 are the bounds on the conditional densities of Proposition 3.3, and $0 \leq D_k < \infty$ is a nonnegative constant given by

$$D_k = \sup_s g(s/\delta_k) \tag{6.142}$$

where the $g(s,\delta_k)$ are the bounded functions whose existence was assumed in Assumption 6.7(e).

Our claim that the normalization has favorably impacted the convergence is qualitative, and comes from the fact that these GSD step bounds indicate bias which

decays proportionally to δ_k^2, and a variance bound which is decaying to zero with δ_k.

Conditions (6.140) and (6.141) are related to the conditions of Assumption 6.3 by making the associations

$$K_1 = 2, \quad \alpha_k = \delta_k \tag{6.143}$$

$$K_2 = 2L, \quad \beta_k = B\delta_k^2 \tag{6.144}$$

$$K_3 = 0, \quad K_4 = 4(p_0 B_0 \lambda_0^2 + p_1 B_1 \lambda_1^2), \quad \nu_k = D_k \delta_k \tag{6.145}$$

Since B, B_0, B_1, D_k, and the costs λ_0, λ_1 are bounded, the requirements of Assumption 6.4 dictate that the stepsize sequence $\{\rho_k\}$ and window-width sequence $\{\delta_k\}$ be such that

$$\rho_k \geq 0, \qquad \sum_{i=1}^{\infty} \rho_k = \infty, \qquad \sum_{i=1}^{\infty} \rho_k^2 < \infty, \tag{6.146}$$

$$\delta_k \geq 0, \qquad \sum_{i=1}^{\infty} \rho_k \delta_k = \infty, \qquad \sum_{i=1}^{\infty} \rho_k \delta_k^2 < \infty, \qquad \sum_{i=1}^{\infty} \rho_k^2 \delta_k < \infty \tag{6.147}$$

If we adopt the forms

$$\rho_k = \frac{\rho_1}{k^a}, \quad \delta_k = \frac{\delta_1}{k^b} \tag{6.148}$$

then the above conditions imply the set of inequalities

$$a \leq 1, \qquad a \geq 1/2 \tag{6.149}$$

$$b \leq -a + 1, \quad b \geq -(1/2)a + 1/2, \quad b \geq -2a + 1 \tag{6.150}$$

which define the region of allowable stepsizes shown in Figure 6-2. Notice the inactive constraint. It is interesting that comparison of Figures 6-1 and 6-2 indicates that the regions of convergence for the unnormalized and normalized variants are completely nonoverlapping and that the size of the region for the normalized steps is in fact smaller. Also note that the choice of sequences

$$\rho_k = \frac{\rho_1}{\sqrt{k}}, \quad \delta_k = \frac{\delta_1}{\sqrt{k}} \tag{6.151}$$

is allowed for the normalized variant, but not the unnormalized one.

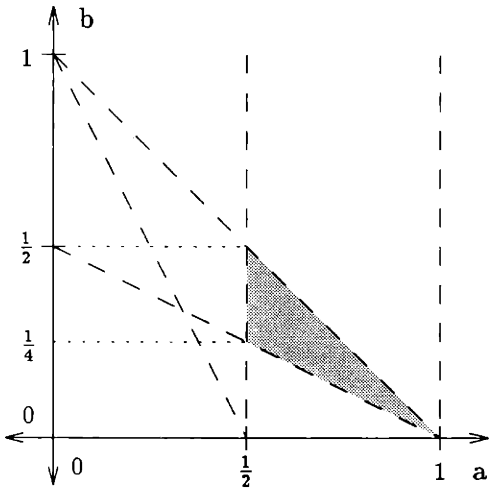


Figure 6-2: Allowable Ranges of a and b for the normalized WIN algorithm.

The preceding discussion indicates that the normalized variant of the one-dimensional window algorithm also converges by Proposition 6.1.

Network Case

We now address the network problem, for which we intend to show that each coordinate of the step obeys a GSD property. The major effort in the network problem involves establishing that our scheme of using *estimated* costs, based on combining relative frequency estimates of the operating points of the network DMs, is viable. In particular, we must determine sufficient conditions on the estimated costs which guarantee convergence, and then verify that these conditions hold under our suggested methodology.

To begin, suppose that rather than having the fixed costs λ_0 and λ_1 available, the unnormalized window algorithm utilizes estimates of those costs, denoted $\hat{\lambda}_0$ and $\hat{\lambda}_1$ We investigate the convergence properties of the algorithm

$$\Theta_{k+1} = \Theta_k - \rho_k Z_k, \; k = 1, 2, \ldots \tag{6.152}$$

employing steps of the form

$$Z_k(X_k, \hat{\lambda}_{0(k)}, \hat{\lambda}_{1(k)}, \Theta_k, \delta_k) = \begin{cases} -\hat{\lambda}_{0(k)} h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_0 \\ +\hat{\lambda}_{1(k)} h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases}$$
$$= \hat{\tilde{Q}}_{B(k)} h(Y_k - \Theta_k, \delta_k) \qquad (6.153)$$

where $\hat{\lambda}_{0(k)}, \hat{\lambda}_{1(k)}$ are estimates of the true costs λ_0 and λ_1 available at time k, and

$$\hat{\tilde{Q}}_{B(k)} = \begin{cases} -\hat{\lambda}_{0(k)} & \text{if } H^k = H_0 \\ +\hat{\lambda}_{1(k)} & \text{if } H^k = H_1 \end{cases} \qquad (6.154)$$

Suppose the following properties of the estimated costs hold.

Assumption 6.8 (Estimated Costs)

For the estimated costs $\hat{\lambda}_{0(k)}$ and $\hat{\lambda}_{1(k)}$ we assume the following hold for all k:

(a) *(Conditional Independence from the local measurement) $\lambda_{i(k)}(\underline{V}_k)$, $i = 0, 1$ where the random vector $\underline{V}_k$ is statistically conditionally independent of the measurement $X_k = \{Y_k, H^k\}$*

(b) *(Unbiased, conditioned on Θ_k) $E_{\underline{V}_k}\{\hat{\lambda}_{i(k)}(\underline{V}_k)|\Theta_k\} = \lambda_i$, $i = 0, 1, \forall \theta_k \in \Re$*

(c) *(Bounded Second Moment, conditioned on Θ_k) $\exists\ 0 < A_i < \infty$ such that $E_{\underline{V}_k}\{\hat{\lambda}^2_{i(k)}(\underline{V}_k)|\Theta_k\} \leq A_i$, $i = 0, 1, \forall \theta_k \in \Re$*

Assumption 6.8(a) indicates that realizations of the estimated costs depend on randomness which is statistically conditionally independent from the local measurement X_k at time k. We show that this property is guaranteed in the team problem by the conditional independence of the observations. Assumption (b) indicates that the estimated costs are unbiased, conditioned on the threshold parameter Θ. Assumption (c) implies that the estimates have bounded variance.

These properties are sufficient for us to establish that using the estimated costs in place of the true costs still gives an algorithm for which the following generalized

stochastic descent conditions hold.

Lemma 6.4 (Generalized Stochastic Descent Conditions: 1D WIN Unnormalized Variant, Estimated Costs)

Let $\hat{\lambda}_{0(k)}$, $\hat{\lambda}_{1(k)}$ be estimated costs obeying Assumption 6.8. Then for the step

$$\begin{aligned} Z_k &= \hat{\tilde{Q}}_{B(k)}(X_k, \hat{\lambda}_{0(k)}, \hat{\lambda}_{1(k)}) h(Y_k - \Theta_k, \delta_k) \\ &= \begin{cases} -\hat{\lambda}_{0(k)} h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_0 \\ +\hat{\lambda}_{1(k)} h(Y_k - \Theta_k, \delta_k) & \text{if } H^k = H_1 \end{cases} \end{aligned} \tag{6.155}$$

it holds that

(i)

$$\frac{dJ_B}{d\theta}(\Theta_k) E\{Z_k|\Theta_k\} \geq \left|\frac{dJ_B}{d\theta}(\Theta_k)\right|^2 - LB\delta_k \left|\frac{dJ_B}{d\theta}(\Theta_k)\right| \tag{6.156}$$

(ii)

$$E\{|Z_k|^2|\Theta_k\} \leq (p_0 B_0 A_0 + p_1 B_1 A_1) D_k \frac{1}{\delta_k} \tag{6.157}$$

where L is the Lipschitz constant on the gradient of J, $0 \leq B < \infty$ is the nonnegative constant assumed in Assumption 6.7(f), B_0 and B_1 are the bounds on the conditional densities of Proposition 3.3, $0 \leq D_k < \infty$ is a nonnegative constant given by

$$D_k = \sup_s g(s/\delta_k) \tag{6.158}$$

where the $g(s, \delta_k)$ are the bounded functions whóse existence was assumed in Assumption 6.7(e), and A_0, A_1 are the bounds on the second moments of the estimates $\hat{\lambda}_{0(k)}, \hat{\lambda}_{1(k)}$ defined in Assumption 6.8.

Proof.

(i) Assumptions 6.8 (a) and (b) imply that

$$E\{Z_k|\Theta_k\} = E_{Y_k, H^k, \hat{\tilde{Q}}_{B(k)}}\{\hat{\tilde{Q}}_{B(k)} h(Y_k - \Theta_k, \delta_k)|\Theta_k\}$$

$$
\begin{aligned}
&= E_{H^k}\{E_{\hat{\tilde{Q}}_{B(k)},Y_k}\{\hat{\tilde{Q}}_{B(k)}h(Y_k-\Theta_k,\delta_k)|\Theta_k,H^k=H_i\}|\Theta_k\}\\
&= E_{H^k}\{E_{\hat{\lambda}_{i(k)},Y_k}\{(-1)^{i+1}\hat{\lambda}_{i(k)}h(Y_k-\Theta_k,\delta_k)|\Theta_k,H^k=H_i\}|\Theta_k\}\\
&= E_{H^k}\{E_{\hat{\lambda}_{i(k)}}\{(-1)^{i+1}\hat{\lambda}_{i(k)}|\Theta_k,H^k=H_i\}E_{Y_k}\{h(Y_k-\Theta_k,\delta_k)|\Theta_k,H^k=H_i\}|\Theta_k\}\\
&= E_{H^k}\left\{(-1)^{i+1}\lambda_i\left(\int_{-\infty}^{\infty}h(y-\Theta_k,\delta_k)p_{Y|H_i}(y|H_i)\,dy\right)\right\}\\
&= -\lambda_0\int_{-\infty}^{\infty}h(y-\Theta_k,\delta_k)p_0p_{Y|H_0}(y|H_0)\,dy\\
&\quad +\lambda_1\int_{-\infty}^{\infty}h(y-\Theta_k,\delta_k)p_1p_{Y|H_1}(y|H_1)\,dy\\
&= \frac{\partial\hat{J}_B}{\partial\theta}(\Theta_k,\delta_k) \qquad (6.159)
\end{aligned}
$$

Then the development follows in identical fashion to (6.124).

(ii)

$$
\begin{aligned}
E\{|Z_k|^2|\Theta_k\} &= E_{\hat{\tilde{Q}}_{B(k)},Y_k,H^k}\{\hat{\tilde{Q}}^2_{B(k)}h^2(Y_k-\Theta_k,\delta_k)|\Theta_k\}\\
&= E_{H^k}\{E_{\hat{\lambda}_{i(k)},Y_k}\{\hat{\lambda}^2_{i(k)}h^2(Y_k-\Theta_k,\delta_k)|\Theta_k,H^k=H_i\}|\Theta_k\}\\
&= E_{H^k}\{E\{\hat{\lambda}^2_{i(k)}|\Theta_k,H^k=H_i\}E_{Y_k}\{h^2(Y_k-\Theta_k,\delta_k)|\Theta_k,H^k=H_i\}|\Theta_k\}\\
&= E\{\hat{\lambda}^2_{0(k)}|\Theta_k\}\int_{-\infty}^{\infty}h^2(y-\Theta_k,\delta_k)p_0p_{Y|H_0}(y|H_0)\,dy\\
&\quad +E\{\hat{\lambda}^2_{1(k)}|\Theta_k\}\int_{-\infty}^{\infty}h^2(y-\Theta_k,\delta_k)p_1p_{Y|H_1}(y|H_1)\,dy\\
&\leq (p_0B_0A_0+p_1B_1A_1)D_k\frac{1}{\delta_k} \qquad (6.160)
\end{aligned}
$$

where the argument has been finished in analogous fashion to (6.126).

■

The proposition indicates that the conditions of Assumption 6.8 are sufficient for the use of estimated costs in place of the true costs. Furthermore, the bounds that we demonstrate exhibit only minor differences to what we were able to prove concerning the unnormalized variant of the one-dimensional window algorithm using the true costs. In particular, the terms λ_0^2, λ_1^2) in condition (ii) have been replaced by the bounds on the second moments of the estimated costs A_0, A_1. Similar results follow for the normalized variant.

On the Estimated Costs in the Network Problem

The analysis of the preceding section indicates that if we can construct estimates of the costs for the network problem which obey Assumption 6.8, then we will be able to show that the step along each coordinate obeys a generalized stochastic descent property. We can then invoke Proposition 6.2 to prove convergence. Thus, in this section we focus on verifying that the estimation scheme suggested in Chapter 5 produces estimates of the costs which obey the conditions of Assumption 6.8.

We proceed as follows. We first consider a simple hypothetical example to illustrate the type of structural properties the coupling costs must have so that the analysis goes through. We then invoke the results of Proposition 3.7 to argue that these structural properties do in fact hold for general tree-structured networks with conditionally independent observations. The arguments are based on the fact that the coupling costs contain sums of products of probabilities of false alarm or detection which correspond to different DMs, and which are therefore estimated using conditionally independent observations. This is sufficient to separate expected values of products of estimated probabilities into products of expected values. Then properties of relative frequency estimators are used to argue that the overall cost has the correct expected value. The boundedness of the second moment of the costs follows in similar fashion.

Suppose that a set of N IID measurements $\underline{X}_1, \ldots, \underline{X}_N$ of the form $\underline{X}_k = \{Y_{A(k)}, Y_{B(k)}, H^k\}$ are available in synchronous fashion to two DMs A and B. DM A receives $X_{A(k)} = \{Y_{A(k)}, H^k\}$ and DM B receives $X_{B(k)} = \{Y_{B(k)}, H^k\}$. The DMs employ linear threshold rules α and β, respectively. Suppose that the measurements are used to compute two local estimates of the probability of false alarm of the form

$$\hat{P}_F^A = \frac{N_F^A}{N_{H_0}}, \ \hat{P}_F^B = \frac{N_F^B}{N_{H_0}} \tag{6.161}$$

where N_F^A denotes the number of times over the set of N measurements that $u_A = 1$ and $H = H_0$, N_F^B denotes the number of times $u_B = 1$ and $H = H_0$, and N_{H_0} denotes the number of occurrences of hypothesis H_0 in the string of N measurements. These

estimates represent the empirical relative frequency estimates of the probabilities of false alarm at each DM. The properties of these estimators are discussed in Appendix A. It is well known that the estimates are unbiased, given the thresholds α, β and the value of N_{H_0}. This is easily demonstrated for $\hat{P}_F^A$ by defining the indicator variable

$$Q_{A(k)}(Y_{A(k)}, H^k, \alpha) = \begin{cases} 1 & \text{if } Y_{A(k)} > \alpha \text{ and } H^k = H_0 \\ 0 & \text{else} \end{cases} \tag{6.162}$$

and taking

$$N_F^A = \sum_{k=1}^{N} Q_{A(k)} \tag{6.163}$$

Since

$$\Pr(Q_{A(k)} = 1) = \Pr(Y_{A(k)} > \alpha \text{ and } H^k = H_0) \tag{6.164}$$

and the measurements are assumed IID, it holds that (see discussion Appendix A, Section A.1)

$$\begin{aligned} E_{\bigcup_{k=1}^N \underline{X}_k}\{\hat{P}_F^A | \alpha, N_{H_0} = n_{H_0}\} &= E_{\bigcup_{k=1}^N X_{A(k)}}\{\hat{P}_F^A | \alpha, N_{H_0} = n_{H_0}\} \\ &= E_{\bigcup_{k=1}^N \{Y_{A(k)}, H^k\}}\left\{ \frac{N_F^A}{N_{H_0}} \middle| \alpha, N_{H_0} = n_{H_0} \right\} \\ &= E_{\bigcup_{k=1}^N \{Y_{A(k)}, H^k\}}\left\{ \frac{\sum_{k=1}^N Q_{A(k)}}{N_{H_0}} \middle| \alpha, N_{H_0} = n_{H_0} \right\} \\ &= \frac{1}{n_{H_0}} E_{\bigcup_{k=1}^N \{Y_{A(k)}, H^k\}}\left\{ \sum_{k=1}^{N} Q_{A(k)} \middle| \alpha, N_{H_0} = n_{H_0} \right\} \\ &= P_F^A \end{aligned} \tag{6.165}$$

It holds in similar fashion that

$$E_{\bigcup_{k=1}^N \underline{X}_k}\{\hat{P}_F^B | \beta, N_{H_0} = n_{H_0}\} = P_F^B \tag{6.166}$$

Averaging out the conditioning on N_{H_0} we obtain

$$\begin{aligned} E_{\bigcup_{k=1}^N X_{A(k)}}\{\hat{P}_F^A | \alpha\} &= E_{N_{H_0}}\{E_{\bigcup_{k=1}^N X_{A(k)}}\{\hat{P}_F^A | \alpha, N_{H_0}\} | \alpha\} \\ &= E_{N_{H_0}}\{P_F^A | \alpha\} \end{aligned}$$

$$= P_F^A \tag{6.167}$$

Similarly, the second moment of the estimate can be shown to be (Appendix A) given by

$$E_{\bigcup_{k=1}^N X_{A(k)}}\{(\hat{P}_F^A)^2|\alpha, N_{H_0} = n_{H_0}\} = \frac{1}{n_{H_0}} P_F^A(1 - P_F^A) + (P_F^A)^2 \tag{6.168}$$

which is clearly bounded such that

$$E_{\bigcup_{k=1}^N X_{A(k)}}\{(\hat{P}_F^A)^2|\alpha, N_{H_0} = n_{H_0}\} \leq P_F^A \leq 1, \ \forall n_{H_0}, P_F^A \tag{6.169}$$

so that

$$E_{\bigcup_{k=1}^N X_{A(k)}}\{(\hat{P}_F^A)^2|\alpha\} \leq 1 \tag{6.170}$$

as well.

It is also of interest in our application to determine whether estimators $\hat{P}_F^A$ and $\hat{P}_F^B$ are uncorrelated, given that they are computed over the same measurement stream. In other words, we wish to determine whether

$$E_{\bigcup_{k=1}^N \underline{X}_k}\{\hat{P}_F^A \hat{P}_F^B|\alpha, \beta\} \stackrel{?}{=} P_F^A P_F^B \tag{6.171}$$

Recall that the network observations Y_A and Y_B are independent, conditioned on the hypothesis, so that

$$\begin{aligned} E_{\bigcup_{k=1}^N \underline{X}_k}\{\hat{P}_F^A \hat{P}_F^B|\alpha, \beta\} &= E_{N_{H_0}}\{E_{\bigcup_{k=1}^N \underline{X}_k}\{\hat{P}_F^A \hat{P}_F^B|\alpha, \beta, N_{H_0}\}|\alpha, \beta\} \\ &= E_{N_{H_0}}\{E_{\bigcup_{k=1}^N X_{A(k)}}\{\hat{P}_F^A|\alpha, N_{H_0}\} E_{\bigcup_{k=1}^N X_{B(k)}}\{\hat{P}_F^B|\beta, N_{H_0}\}\} \\ &= E_{N_{H_0}}\{P_F^A P_F^B\} \\ &= P_F^A P_F^B \end{aligned} \tag{6.172}$$

The second moment of the product is also bounded, since

$$E_{\bigcup_{k=1}^N \underline{X}_k}\{(\hat{P}_F^A \hat{P}_F^B)^2|\alpha, \beta\} = E_{N_{H_0}}\{E_{\bigcup_{k=1}^N \underline{X}_k}\{(\hat{P}_F^A)^2(\hat{P}_F^B)^2|\alpha, \beta, N_{H_0}\}|\alpha, \beta\}$$

$$
\begin{aligned}
&= E_{N_{H_0}}\{E_{\bigcup_{k=1}^{N} X_{A(k)}}\{(\hat{P}_F^A)^2|\alpha, N_{H_0}\}E_{\bigcup_{k=1}^{N} X_{B(k)}}\{(\hat{P}_F^B)^2|\beta, N_{H_0}\}\} \\
&= E_{N_{H_0}}\{(\frac{1}{N_{H_0}}P_F^A(1-P_F^A)+(P_F^A)^2)(\frac{1}{N_{H_0}}P_F^B(1-P_F^B)+(P_F^B)^2)\} \\
&\leq (P_F^A(1-P_F^A)+(P_F^A)^2)(P_F^B(1-P_F^B)+(P_F^B)^2) \\
&\leq 1 \qquad (6.173)
\end{aligned}
$$

The above properties hold true for $M > 2$ DMs in identical fashion. Namely, it holds that

$$
\begin{aligned}
E_{\bigcup_{k=1}^{N} \underline{X}_k}\{\hat{P}_F^1 \cdots \hat{P}_F^M|\theta_1, \ldots, \theta_M\} &= P_F^1 \cdots P_F^M \\
E_{\bigcup_{k=1}^{N} \underline{X}_k}\{(\hat{P}_F^1)^2 \cdots (\hat{P}_F^M)^2|\theta_1, \ldots, \theta_M\} &\leq 1 \qquad (6.174)
\end{aligned}
$$

In view of the above discussion, we present the following proposition, indicating that the costs in the network problem do in fact obey Assumption 6.8.

Proposition 6.5 (Properties of the Estimated Costs for the Team Problem) *Let $\hat{\lambda}_0^{il}$, $\hat{\lambda}_1^{il}$ be coupling costs corresponding to parameter θ_l associated with DM i. Then the following properties hold for all components l, DMs i, and times k:*

(a) *(Conditional Independence from the Local Measurement) $\hat{\lambda}_0^{il}$, $\hat{\lambda}_1^{il}$ are statistically conditionally independent of the local measurement $X_{l(k)}$*

(b) *(Unbiased, conditioned on $\underline{\Theta}_k$) $E\{\hat{\lambda}_{j(k)}^{il}|\underline{\Theta}_k\} = \lambda_{j(k)}^{il}$, $j = 0, 1, \forall \underline{\theta}_k \in \Re^N$*

(c) *(Bounded Second Moment, conditioned on $\underline{\Theta}_k$) $\exists\ 0 < A_j^{il} < \infty$ such that $E\{(\hat{\lambda}_{j(k)}^{il})^2|\underline{\Theta}_k\} \leq A_j^{il}$, $j = 0, 1, \forall \underline{\theta}_k \in \Re^N$*

Proof. All three properties follow directly from Proposition 3.7 and the arguments presented above. Namely, the proposition indicates that the estimated costs are sums of products of estimated probabilities of false alarm or detection, where none of the component probabilities corresponds to DM i (so (1) follows), and where the products always consist of terms from different DMs (hence (2)), and where for networks with

finite numbers of DMs the sums always contain a finite number of terms(hence (3)).

■

The above proposition is clarified by considering an example, so we consider 3-Vee. If we expand out the coupling costs they are given by

$$\begin{aligned}
\hat{\lambda}_0^A &= \hat{P}_F^{C(10)} - \hat{P}_F^{C(00)} - \hat{P}_F^B \hat{P}_F^{C(10)} + \hat{P}_F^B \hat{P}_F^{C(00)} + \hat{P}_F^B \hat{P}_F^{C(11)} - \hat{P}_F^B \hat{P}_F^{C(01)} \\
\hat{\lambda}_1^A &= \hat{P}_D^{C(10)} - \hat{P}_D^{C(00)} - \hat{P}_D^B \hat{P}_D^{C(10)} + \hat{P}_D^B \hat{P}_D^{C(00)} + \hat{P}_D^B \hat{P}_D^{C(11)} - \hat{P}_D^B \hat{P}_D^{C(01)} \\
\hat{\lambda}_0^B &= \hat{P}_F^{C(01)} - \hat{P}_F^{C(00)} - \hat{P}_F^A \hat{P}_F^{C(01)} + \hat{P}_F^A \hat{P}_F^{C(00)} + \hat{P}_F^A \hat{P}_F^{C(11)} - \hat{P}_F^A \hat{P}_F^{C(10)} \\
\hat{\lambda}_1^B &= \hat{P}_D^{C(01)} - \hat{P}_D^{C(00)} - \hat{P}_D^A \hat{P}_D^{C(01)} + \hat{P}_D^A \hat{P}_D^{C(00)} + \hat{P}_D^A \hat{P}_D^{C(11)} - \hat{P}_D^A \hat{P}_D^{C(10)} \\
\hat{\lambda}_0^{C(00)} &= 1 - \hat{P}_F^A - \hat{P}_F^B + \hat{P}_F^A \hat{P}_F^B \\
\hat{\lambda}_1^{C(00)} &= 1 - \hat{P}_D^A - \hat{P}_D^B + \hat{P}_D^A \hat{P}_D^B \\
\hat{\lambda}_0^{C(01)} &= \hat{P}_F^B - \hat{P}_F^A \hat{P}_F^B \\
\hat{\lambda}_1^{C(01)} &= \hat{P}_D^B - \hat{P}_D^A \hat{P}_D^B \\
\hat{\lambda}_0^{C(10)} &= \hat{P}_F^A - \hat{P}_F^A \hat{P}_F^B \\
\hat{\lambda}_1^{C(10)} &= \hat{P}_D^A - \hat{P}_D^A \hat{P}_D^B \\
\hat{\lambda}_0^{C(11)} &= \hat{P}_F^A \hat{P}_F^B \\
\hat{\lambda}_1^{C(11)} &= \hat{P}_D^A \hat{P}_D^B
\end{aligned} \tag{6.175}$$

Suppose that a common set of N network measurements is made available to the three DMs of 3-Vee (6 processors), and that the processor updating α computes $(\hat{P}_F^A(\alpha), \hat{P}_D^A(\alpha))$ based on measurements $\bigcup_{k=1}^N X_{A(k)}$, the processor updating β computes $(\hat{P}_F^B(\beta), \hat{P}_D^B(\beta))$ based on measurements $\bigcup_{k=1}^N X_{B(k)}$, and that the processors updating $\xi_{00}, \xi_{01}, \xi_{10}, \xi_{11}$ compute $(\hat{P}_F^{C(00)}, \hat{P}_D^{C(00)})$, $(\hat{P}_F^{C(01)}, \hat{P}_D^{C(01)})$, $(\hat{P}_F^{C(10)}, \hat{P}_D^{C(10)})$, and $(\hat{P}_F^{C(11)}, \hat{P}_D^{C(11)})$, respectively, based on $\bigcup_{k=1}^N X_{C(k)}$. Each DM then communicates the estimates to those processors which require them, where they are combined according to the appropriate equation of (6.175).

It is evident from the indexing on the LHS, RHS of each equation of (6.175) that

the estimated costs at each DM depend only on operating point estimates computed at the other DMs, so that property (a) of the proposition holds. To illustrate property (b), consider for example the computation of

$$\begin{aligned} E\{\hat{\lambda}_0^{C(00)}|\underline{\Theta}_k = \underline{\theta}_k\} &= E\{1 - \hat{P}_F^A - \hat{P}_F^B + \hat{P}_F^A\hat{P}_F^B|\underline{\Theta}_k = \underline{\theta}_k\} \\ &= 1 - E\{\hat{P}_F^A|\alpha\} - E\{\hat{P}_F^B|\beta\} + E\{\hat{P}_F^A\hat{P}_F^B|\underline{\Theta}_k\} \\ &= 1 - P_F^A - P_F^B + P_F^A P_F^B \end{aligned} \tag{6.176}$$

where we have used the fact that the estimates from different DMs are uncorrelated. The second moment of $\hat{\lambda}_0^{C(00)}$ is also bounded,

$$\begin{aligned} E\{(\hat{\lambda}_0^{C(00)})^2|\underline{\Theta}_k = \underline{\theta}_k\} &= E\{(1 - \hat{P}_F^A - \hat{P}_F^B + \hat{P}_F^A\hat{P}_F^B)^2|\underline{\Theta}_k = \underline{\theta}_k\} \\ &= 1 - 2E\{\hat{P}_F^A|\alpha\} - 2E\{\hat{P}_F^B|\beta\} + 4E\{\hat{P}_F^A\hat{P}_F^B|\alpha,\beta\} \\ &\quad -2E\{(\hat{P}_F^A)^2\hat{P}_F^B|\alpha,\beta\} - 2E\{\hat{P}_F^A(\hat{P}_F^B)^2|\alpha,\beta\} + E\{(\hat{P}_F^A)^2|\alpha\} \\ &\quad +E\{(\hat{P}_F^A)^2|\alpha\} + E\{(\hat{P}_F^B)^2|\beta\} + E\{(\hat{P}_F^A)^2(\hat{P}_F^B)^2|\alpha,\beta\} \\ &\leq 1 + 4P_F^A P_F^B + P_F^A + P_F^B + P_F^A P_F^B \\ &< 8 \end{aligned} \tag{6.177}$$

so that property (c) holds as well.

We have thus shown that the estimation scheme suggested in Chapter 5 is sufficient to guarantee that the network window algorithm WIN possesses a GSD property along each component, and thus converges based on Proposition 6.2. The result holds for either unnormalized or normalized window steps along each coordinate. Note also that we are free to choose distinct sequences ρ_k^i, δ_k^i at each node, so long as the combination along each coordinate conforms to Figure 6-1 for the unnormalized algorithm and Figure 6-2 for the normalized algorithm.

One implication of the required properties on the network costs which is perhaps surprising is the fact that asymptotic convergence is guaranteed even in the worst case, for which estimates of the operating points are computed using $N_{H_0} = N_{H_1} = 1$,

i.e., even if the operating point estimate of each DM is a 0-1 estimate[6]. In the context of interpreting the estimated coupling costs as local models, this means that the local models can be extremely crude, so long as they are correct in an average sense. However, since the quality of the local models enters the analysis through the bound (6.157), the expected rate of convergence is expected to vary with the quality of the local models. But the fact that asymptotic convergence is guaranteed for a wide range of quality in the local models suggests the possibility that many interesting issues may be explored in this context.

The arguments demonstrating convergence of the WIN-BP algorithm are even cleaner, since the method of computing the costs decomposes the computation of the estimated coupling costs into terms arriving from downstream DMs and terms arriving from upstream DMs. The required structure is clearly evident in the computations; hence we omit a detailed analysis.

[6]Of course, it is not clear how many local estimation trials must be performed so that at least one case of H_0 and one case of H_1 is observed. This is a function of the prior probabilities.

6.3.2 KW-Type Algorithms

Because the multivariable KW techniques explicitly construct estimates of each partial derivative, we can handle the one-dimensional and multivariable cases simultaneously. In this section, we consider the application of both one-sided and two-sided KW techniques to $P_\epsilon(\underline{\theta})$([7]), where $\underline{\theta} \in \Re^N$ for a team of $M \geq 1$ DMs and $N \geq 1$. The general Bayes problem $J_B(\theta)$ is handled with very minor changes in our analysis. We initially restrict to use of the same stepsize parameters at each processor, and show that Proposition 6.1 applies.

The application of KW-Type methods to the training problem is fairly straightforward once the sampling error is characterized. Recall that in the KW setting, only samples of the function itself are used to perform the optimization. For a team of $M \geq 1$ DMs

$$Q(\underline{X}, \underline{\Theta}) = \begin{cases} 1 & \text{if team decides } H_1 \text{ and } H = H_0 \\ 1 & \text{if team decides } H_0 \text{ and } H = H_1 \\ 0 & \text{else} \end{cases} \tag{6.178}$$

where

$$E_{\underline{X}}\{Q(\underline{X}, \underline{\Theta}) | \underline{\Theta} = \underline{\theta}\} = P_\epsilon(\underline{\theta}) \tag{6.179}$$

Sampling Error

Define the sampling error[8]

$$\eta(\underline{X}, \underline{\Theta}) = Q(\underline{X}, \underline{\Theta}) - P_\epsilon(\underline{\Theta}) \tag{6.180}$$

This error cannot be characterized as additive or multiplicative, and results from the fact we are approximating a fixed unknown probability with a single sample estimate of the empirical relative frequency. As discussed in Appendix A, Section

[7] We omit the superscript *Team* to emphasize that our analysis encompasses the single DM problem as well.

[8] Note η denotes a random variable despite begin indicated in lower case.

A.1, the random variable $Q(\underline{X}, \underline{\Theta})$ is distributed Bernoulli. Using this fact, it is straightforward to establish the following properties of the sampling error.

Lemma 6.5 (Properties of Error in Sampling $P_\epsilon(\underline{\theta})$)

(a) *(Bounded)* η *is bounded such that* $-1 \leq \eta \leq +1$, *i.e.,* $|\eta| \leq 1$

(b) *(First and Second Moments, conditioned on* $\underline{\Theta}$*)* $E\{\eta|\underline{\Theta}\} = 0$

$E\{\eta^2|\underline{\Theta}\} = P_\epsilon(\underline{\Theta})(1 - P_\epsilon(\underline{\Theta})) \leq \frac{1}{4}$

(c) *(First and Second Moments, unconditional)* $E\{\eta\} = 0$

$E\{\eta^2\} = E_{\underline{\Theta}}\{P_\epsilon(\underline{\theta})(1 - P_\epsilon(\underline{\theta}))\} \leq \frac{1}{4}$

Proof. These assertions follow immediately from the properties of Bernoulli random variables as described in Appendix A.

■

Most importantly, the sampling error is *zero-mean* and *bounded variance*, conditioned on the parameter vector $\underline{\Theta}$. Notice that the variance of the sampling error depends on the conditional densities of the hypothesis test only insofar as they dictate the true $P_\epsilon(\underline{\theta})$ corresponding to the network threshold settings $\underline{\theta}$.

Define the quantities

$$\begin{aligned} \eta_k^i &= Q(\underline{X}_k, \underline{\Theta}_k) - P_\epsilon(\underline{\Theta}_k) \\ \eta_k^{i(+)} &= Q(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - P_\epsilon(\underline{\Theta}_k + \delta_k \underline{e}_i) \\ \eta_k^{i(-)} &= Q(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{e}_i) - P_\epsilon(\underline{\Theta}_k - \delta_k \underline{e}_i) \end{aligned} \tag{6.181}$$

where $\underline{e}_i$ denotes the unit vector along the ith coordinate, which represent the sampling errors associated with a finite difference approximation along the ith coordinate at time k. Our method of proof will specifically require that the following conditions on the sample errors be satisfied along each coordinate. For the one-sided variant

$$E_{\underline{X}_k}\{(\eta_k^{i(+)} - \eta_k^i)|\underline{\Theta}_k\} = 0$$

$$E_{\underline{X}_k}\{(\eta_k^{i(+)} - \eta_k^i)^2 | \underline{\Theta}_k\} \leq 2\sigma^2 \tag{6.182}$$

and for the two-sided variant

$$\begin{aligned} E_{\underline{X}_k}\{(\eta_k^{i(+)} - \eta_k^{i(-)}) | \underline{\Theta}_k\} &= 0 \\ E_{\underline{X}_k}\{(\eta_k^{i(+)} - \eta_k^{i(-)})^2 | \underline{\Theta}_k\} &\leq 2\sigma^2 \end{aligned} \tag{6.183}$$

where σ^2 is a bound on the variance of each noise sample. The first of the two conditions is commonly referred to as the martingale difference noise assumption [58], and is clearly satisfied for conditionally zero-mean sampling errors. The second condition is a bound on the conditional second moment of the noise difference, and is satisfied for bounded variance sampling errors. Notice that conditional statistical independence of the errors along each coordinate is *not* required. This means there is no difficulty in using the same network measurement $\underline{X}_k$ to update every component at each k.

One-Sided Variant

We begin with the one-sided variant, also referred to as the asymmetric finite difference technique, because its proof is the easiest. The one-sided variant approximates the ith partial derivative using the one-sided finite difference approximation

$$Z_{i(k)}(\underline{X}_k, \underline{\Theta}_k, \delta_k) = [Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k)]/\delta_k \tag{6.184}$$

where $\{\delta_k\}$ is a decreasing sequence of perturbations.

We begin by obtaining a bound on the bias introduced by estimating the directional derivative of a function using a one-sided finite difference approximation.

Lemma 6.6 (Bias in One-Sided FD Estimate of Gradient)
Let $J(\underline{\theta}) : \Re^N \mapsto \Re$ be a differentiable function such that $\nabla J(\underline{\theta})$ is Lipschitz continuous with constant $L > 0$. Then for the finite-difference approximation of the directional derivative in direction $\underline{q}$ given by

$$\hat{J}'(\underline{\theta}; \underline{q}) = (J(\underline{\theta} + \delta \underline{q}) - J(\underline{\theta}))/\delta \tag{6.185}$$

it holds that

$$|\hat{J}'(\underline{\theta}; \underline{q}) - \nabla J(\underline{\theta})^T \underline{q}| \leq (L/2)\|\underline{q}\|^2 \delta \tag{6.186}$$

Proof. We express (6.186) as

$$|\hat{J}'(\underline{\theta}; \underline{q}) - \nabla J(\underline{\theta})^T \underline{q}| = \frac{1}{\delta} |J(\underline{\theta} + \delta \underline{q}) - J(\underline{\theta}) - \delta \nabla J(\underline{\theta})^T \underline{q}| \tag{6.187}$$

and invoke the first-order descent lemma (Appendix B) with the association $\underline{y} = \delta \underline{q}$ to obtain the result.

■

Thus, the one-sided finite difference approximation to the directional derivative contains bias proportional to δ. As the quantity δ approaches zero, this bias goes to zero.

Let $\underline{e}_i$ denote the unit vector along the ith coordinate. Then, the one-sided KW algorithm uses steps of the form

$$\begin{aligned}
\underline{Z}_k(\underline{X}_k, \underline{\Theta}_k, \delta_k) &= \sum_{i=1}^{N} \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k)]\underline{e}_i}{\delta_k} \\
&= \sum_{i=1}^{N} \frac{[(J(\underline{\Theta}_k + \delta_k \underline{e}_i) + \eta_k^{i(+)}) - (J(\underline{\Theta}_k) + \eta_k^i)]\underline{e}_i}{\delta_k} \\
&= \sum_{i=1}^{N} \frac{[J(\underline{\Theta}_k + \delta_k \underline{e}_i)) - J(\underline{\Theta}_k)]\underline{e}_i}{\delta_k} + \sum_{i=1}^{N} \frac{[\eta_k^{i(+)} - \eta_k^i]\underline{e}_i}{\delta_k}
\end{aligned}$$

$$= \sum_{i=1}^{N} \frac{\partial J}{\partial \Theta_i}(\underline{\Theta}_k)\underline{e}_i + \sum_{i=1}^{N} \beta_{i(k)}\underline{e}_i + \sum_{i=1}^{N} \xi_{i(k)}\underline{e}_i \tag{6.188}$$

where the $\beta_{i(k)}$ are deterministic biases proportional to δ_k cf. Lemma 6.6 and the $\xi_{i(k)}$([9]) denote the "effective" sampling error, which is proportional to $1/\delta_k$. Thus, the effective sampling error becomes unbounded as $\delta_k \to 0$, while the bias goes to zero.

The immediately preceding discussion has touched on the points of major relevance in the development of the stepsize bounds which we now present. We follow the development of Polyak and Tsypkin in [50] and Polyak [51].

Proposition 6.6 (GSD Property: KW, 1-Sided Variant)

For the step

$$\underline{Z}_k = \sum_{i=1}^{N} \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k)]\underline{e}_i}{\delta_k} \tag{6.189}$$

it holds that

(i)

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} \geq \|\nabla J(\underline{\Theta}_k)\|^2 - (NL/2)\delta_k \|\nabla J(\underline{\Theta}_k)\| \tag{6.190}$$

(ii)

$$E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} \leq 2N\|\nabla J(\underline{\Theta}_k)\|^2 + (NL^2/2)\delta_k^2 + (2N\sigma^2)\frac{1}{\delta_k^2} \tag{6.191}$$

where L is the Lipschitz constant on the gradient of J, N is the dimension of the vector $\underline{\theta}$, and σ^2 is a bound on the variance of the sampling error η.

Proof.

(i) Using Lemma 6.5, Lemma 6.6 and the Schwartz Inequality (Appendix B) we obtain

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} = \nabla J(\underline{\Theta}_k)^T E_{\underline{X}_k}\left\{ \sum_{i=1}^{N} \frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k))\underline{e}_i}{\delta_k} \middle| \Theta_k \right\}$$

[9] Note $\xi_{i(k)}$ is a random variable.

$$
\begin{aligned}
&= \nabla J(\underline{\Theta}_k)^T \left(E_{\underline{X}_k} \left\{ \sum_{i=1}^{N} \frac{(J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k))\underline{e}_i}{\delta_k} \middle| \underline{\Theta}_k \right\} \right. \\
&\quad \left. + E_{\underline{X}_k} \left\{ \sum_{i=1}^{N} \frac{(\eta_k^{i(+)} - \eta_k^i)\underline{e}_i}{\delta_k} \middle| \underline{\Theta}_k \right\} \right) \\
&= \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} \frac{(J(\underline{\Theta}_k + \delta \underline{e}_i) - J(\underline{\Theta}_k))\underline{e}_i}{\delta_k} \\
&= \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} \left[\frac{J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k)}{\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right. \\
&\quad \left. + \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right] \underline{e}_i \\
&= \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} \left[\frac{J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k)}{\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right] \underline{e}_i \\
&\quad + \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} [\nabla J(\underline{\Theta}_k)^T \underline{e}_i] \underline{e}_i \\
&\geq -\nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} [(L/2)\|\underline{e}_i\|^2 \delta_k] \underline{e}_i + \sum_{i=1}^{N} (\nabla J(\underline{\Theta}_k)^T \underline{e}_i)^2 \\
&= -(L/2)\delta_k \sum_{i=1}^{N} \nabla J(\underline{\Theta}_k)^T \underline{e}_i + \|\nabla J(\underline{\Theta}_k)\|^2 \\
&\geq -(L/2)\delta_k \sum_{i=1}^{N} \|\nabla J(\underline{\Theta}_k)\| \|\underline{e}_i\| + \|\nabla J(\underline{\Theta}_k)\|^2 \\
&= \|\nabla J(\underline{\Theta}_k)\|^2 - (NL/2)\delta_k \|\nabla J(\underline{\Theta}_k)\| \qquad (6.192)
\end{aligned}
$$

(ii) Using the elementary inequality

$$
\left(\sum_{i=1}^{p} a_i \right)^2 \leq p \sum_{i=1}^{p} a_i^2 \qquad (6.193)
$$

for the case $p = 2$, the fact that

$$
E_{\underline{X}_k} \{ (\eta_k^{i(+)} - \eta_k^{i(-)})^2 | \underline{\Theta}_k \} \leq 2\sigma^2 \qquad (6.194)
$$

Lemma 6.6, and the Schwartz Inequality we obtain

$$
E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} = E_{\underline{X}_k} \left\{ \left\| \sum_{i=1}^{N} \frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k))\underline{e}_i}{\delta_k} \right\|^2 \middle| \underline{\Theta}_k \right\}
$$

$$
\begin{aligned}
&= E_{\underline{X}_k}\left\{\left\|\sum_{i=1}^{N}\frac{(J(\underline{\Theta}_k+\delta_k\underline{e}_i)-J(\underline{\Theta}_k))\underline{e}_i}{\delta_k}+\frac{(\eta_k^{i(+)}-\eta_k^i)\underline{e}_i}{\delta_k}\right\|^2\middle|\underline{\Theta}_k\right\}\\
&= \sum_{i=1}^{N}\left(\frac{J(\underline{\Theta}_k+\delta_k\underline{e}_i)-J(\underline{\Theta}_k)}{\delta_k}\right)^2\|\underline{e}_i\|^2\\
&\quad +E_{\underline{X}_k}\left\{\sum_{i=1}^{N}\left(\frac{\eta_k^{i(+)}-\eta_k^i}{\delta_k}\right)^2\|\underline{e}_i\|^2\middle|\underline{\Theta}_k\right\}\\
&\leq \sum_{i=1}^{N}\left[\frac{J(\underline{\Theta}_k+\delta_k\underline{e}_i)-J(\underline{\Theta}_k)}{\delta_k}-\nabla J(\underline{\Theta}_k)^T\underline{e}_i\right.\\
&\quad \left.+\nabla J(\underline{\Theta}_k)^T\underline{e}_i\right]^2\|\underline{e}_i\|^2+\sum_{i=1}^{N}\frac{2\sigma^2}{\delta_k^2}\|\underline{e}_i\|^2\\
&\leq \sum_{i=1}^{N}\left(\left|\frac{J(\underline{\Theta}_k+\delta_k\underline{e}_i)-J(\underline{\Theta}_k)}{\delta_k}-\nabla J(\underline{\Theta}_k)^T\underline{e}_i\right|\right.\\
&\quad \left.+|\nabla J(\underline{\Theta}_k)^T\underline{e}_i|\right)^2\|\underline{e}_i\|^2+(2N\sigma^2)\frac{1}{\delta_k^2}\\
&\leq \sum_{i=1}^{N}\left(\frac{L}{2}\delta_k\|\underline{e}_i\|^2+|\nabla J(\underline{\Theta}_k)^T\underline{e}_i|\right)^2\|\underline{e}_i\|^2+(2N\sigma^2)\frac{1}{\delta_k^2}\\
&\leq \sum_{i=1}^{N}\left[2\left(\frac{L}{2}\delta_k\|\underline{e}_i\|^2\right)^2+2\left(|\nabla J(\underline{\Theta}_k)^T\underline{e}_i|\right)^2\right]\|\underline{e}_i\|^2+(2N\sigma^2)\frac{1}{\delta_k^2}\\
&\leq \sum_{i=1}^{N}[(L^2/2)\|\underline{e}_i\|^4\delta_k^2+2\|\nabla J(\underline{\Theta}_k)\|^2\|\underline{e}_i\|^2]\|\underline{e}_i\|^2+(2N\sigma^2)\frac{1}{\delta_k^2}\\
&= 2N\|\nabla J(\underline{\Theta}_k)\|^2+(NL^2/2)\delta_k^2+(2N\sigma^2)\frac{1}{\delta_k^2} \qquad (6.195)
\end{aligned}
$$

■

We may now relate conditions (6.190) and (6.191) with the conditions of Assumption 6.3 by making the associations

$$K_1 = 1, \quad \alpha_k = 1, \ \forall k \qquad (6.196)$$

$$K_2 = (NL/2), \quad \beta_k = \delta_k \qquad (6.197)$$

$$K_3 = 2N, \quad K_4\nu_k = (NL^2/2)\delta_k^2+(2N\sigma^2)\frac{1}{\delta_k^2} \qquad (6.198)$$

where we have not separated K_4 and ν_k. Then Assumption 6.4 requires that the

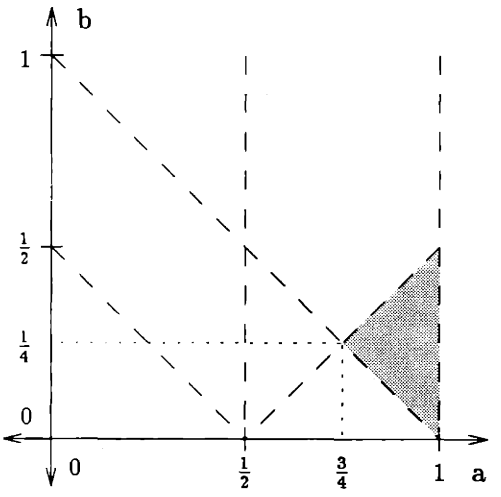


Figure 6-3: Allowable Ranges of a and b for one-sided KW.

stepsize sequence $\{\rho_k\}$ and perturbation sequence $\{\delta_k\}$ be such that

$$\rho_k \geq 0, \qquad \sum_{i=1}^{\infty} \rho_k = \infty, \qquad \sum_{i=1}^{\infty} \rho_k^2 < \infty \tag{6.199}$$

$$\delta_k \geq 0, \qquad \sum_{i=1}^{\infty} \rho_k \delta_k < \infty, \qquad \sum_{i=1}^{\infty} \rho_k^2 \delta_k^2 < \infty, \qquad \sum_{i=1}^{\infty} \frac{\rho_k^2}{\delta_k^2} < \infty \tag{6.200}$$

If we adopt the forms

$$\rho_k = \frac{\rho_1}{k^a}, \qquad \delta_k = \frac{\delta_1}{k^b} \tag{6.201}$$

then the above conditions imply the inequalities

$$a \leq 1, \qquad a \geq 1/2 \tag{6.202}$$

$$b \geq -a + 1, \quad b \geq -a + (1/2), \quad b \leq a - (1/2) \tag{6.203}$$

which define the allowable stepsize ranges depicted in Figure 6-3. Notice the inactive constraint.

Two-Sided Variant

An alternative to the one-sided implementation approximates the ith partial derivative at time k using two-sided finite differences of the form

$$Z_{i(k)}(\underline{X}_k, \underline{\Theta}_k, \delta_k) = (Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{e}_i))/2\delta_k \tag{6.204}$$

This technique typically exhibits better performance than the one-sided variant, but with attendant increases in computational overhead and measurements. In particular, $2N$ samples are required, where N is the dimension of the parameter vector $\underline{\Theta}$, as opposed to $N + 1$ for the one-sided approximation. The technique also requires stronger smoothness conditions on the function being optimized. In particular, we must augment Assumption 6.1 with the following Lipschitz continuity assumption on the Hessian (see Polyak [51]).

Assumption 6.9 (Additional Smoothness Requirement)
Let $J(\underline{\theta}) : \Re^N \mapsto \Re$ be a twice differentiable function for which, in addition to Assumption 6.1 there exists a constant $L > 0$ such that

$$\|\nabla^2 J(\underline{\theta}_1) - \nabla^2 J(\underline{\theta}_2)\| \leq L\|\underline{\theta}_1 - \underline{\theta}_2\| \tag{6.205}$$

for every $\underline{\theta}_1, \underline{\theta}_2 \in \Re^N$, where the matrix norm of A is given by

$$\|A\| = \max_{\{\underline{\theta} \in \Re^N \mid \|\underline{\theta}\|=1\}} \|A\underline{\theta}\| \tag{6.206}$$

The proof of this technique follows in identical fashion to the one-sided method, once we've established the following bound on the bias, which is the analog to Lemma 6.6.

Lemma 6.7 (Bias in Two-Sided FD Estimate of Gradient)
Let $J(\underline{\theta}) : \Re^N \mapsto \Re$ be a twice differentiable function such that the Hessian $\nabla^2 J(\underline{\theta})$ obeys the Lipschitz continuity condition of Assumption 6.9 with constant $L > 0$. Then for the finite-difference approximation of the directional derivative in direction $\underline{q}$ given by

$$\hat{J}'(\underline{\theta}; \underline{q}) = (J(\underline{\theta} + \delta\underline{q}) - J(\underline{\theta} - \delta\underline{q}))/2\delta \tag{6.207}$$

it holds that

$$|\hat{J}'(\underline{\theta}; \underline{q}) - \nabla J(\underline{\theta})^T \underline{q}| \leq (L/6)\|\underline{q}\|^3 \delta^2 \tag{6.208}$$

Proof. Using the second-order descent lemma (Appendix B) we obtain

$$\left| \frac{J(\underline{\theta} + \delta\underline{q}) - J(\underline{\theta} - \delta\underline{q})}{2\delta} - \nabla J(\underline{\theta})^T \underline{q} \right| = \tag{6.209}$$

$$\begin{aligned}
&\frac{1}{2\delta} |J(\underline{\theta} + \delta\underline{q}) - J(\underline{\theta} - \delta\underline{q}) - 2\delta \nabla J(\underline{\theta})^T \underline{q}| \\
&= \frac{1}{2\delta} \Big| (J(\underline{\theta} + \delta\underline{q}) - J(\underline{\theta}) - \delta \nabla J(\underline{\theta})^T \underline{q} - (1/2)\delta^2 \underline{q}^T \nabla^2 J(\underline{\theta}) \underline{q}) \\
&\quad - (J(\underline{\theta} - \delta\underline{q}) - J(\underline{\theta}) + \delta \nabla J(\underline{\theta})^T \underline{q} - (1/2)\delta^2 \underline{q}^T \nabla^2 J(\underline{\theta}) \underline{q}) \Big| \\
&\leq \frac{1}{2\delta} \Big[|J(\underline{\theta} + \delta\underline{q}) - J(\underline{\theta}) - \delta \nabla J(\underline{\theta})^T \underline{q} - (1/2)\delta^2 \underline{q}^T \nabla^2 J(\underline{\theta}) \underline{q}| \\
&\quad + |J(\underline{\theta} - \delta\underline{q}) - J(\underline{\theta}) + \delta \nabla J(\underline{\theta})^T \underline{q} - (1/2)\delta^2 \underline{q}^T \nabla^2 J(\underline{\theta}) \underline{q})| \Big] \\
&\leq \frac{1}{2\delta} \left[(L/6)\|\underline{q}\|^3 \delta^3 + (L/6)\|\underline{q}\|^3 \delta^3 \right] \\
&= (L/6)\|\underline{q}\|^3 \delta^2
\end{aligned} \tag{6.210}$$

■

The required algebraic manipulations are analogous to those required to establish the bounds for the one-sided case. Hence, we present only abbreviated proofs. Notice that with respect to the GSD conditions for the one-sided approximation given in Proposition 6.6, the two-sided approximation gives a more rapidly decaying bias and smaller variance bounds, both of which are indicative of superior convergence

properties.

Proposition 6.7 (GSD Property: 2-Sided Variant)

For the step

$$\underline{Z}_k = \sum_{i=1}^{N} \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{e}_i)]\underline{e}_i}{2\delta_k} \tag{6.211}$$

it holds that

(i)

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} \geq \|\nabla J(\underline{\Theta}_k)\|^2 - (NL/6)\delta_k^2 \|\nabla J(\underline{\Theta}_k)\| \tag{6.212}$$

(ii)

$$E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} \leq 2N\|\nabla J(\underline{\Theta}_k)\|^2 + (NL^2/18)\delta_k^4 + (N\sigma^2/2)\frac{1}{\delta_k^2} \tag{6.213}$$

where L is the Lipschitz constant on Hessian of J of Assumption 6.9, N is the dimension of the vector $\underline{\theta}$, and σ^2 is a bound on the variance of the sampling error η.

Proof.

(i) Using Lemma 6.5, Lemma 6.7 and the Schwartz Inequality (Appendix B) we obtain

$$\begin{aligned}
\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} &= \nabla J(\underline{\Theta}_k)^T E_{\underline{X}_k}\left\{ \sum_{i=1}^{N} \frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{e}_i))\underline{e}_i}{2\delta_k} \middle| \Theta_k \right\} \\
&= \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} \left[\frac{J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k - \delta_k \underline{e}_i)}{2\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right. \\
&\quad \left. + \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right] \underline{e}_i \\
&= \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} \left[\frac{J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k - \delta_k \underline{e}_i)}{2\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right] \underline{e}_i \\
&\quad + \nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} [\nabla J(\underline{\Theta}_k)^T \underline{e}_i] \underline{e}_i \\
&\geq -\nabla J(\underline{\Theta}_k)^T \sum_{i=1}^{N} [(L/6)\|\underline{e}_i\|^3 \delta_k^2] \underline{e}_i + \sum_{i=1}^{N} (\nabla J(\underline{\Theta}_k)^T \underline{e}_i)^2
\end{aligned}$$

$$= -(L/6)\delta_k^2 \sum_{i=1}^{N} \nabla J(\underline{\Theta}_k)^T \underline{e}_i + \|\nabla J(\underline{\Theta}_k)\|^2$$
$$\geq \|\nabla J(\underline{\Theta}_k)\|^2 - (NL/6)\delta^2 \|\nabla J(\underline{\Theta}_k)\| \tag{6.214}$$

(ii) Using the elementary inequality

$$\left(\sum_{i=1}^{p} a_i\right)^2 \leq p \sum_{i=1}^{p} a_i^2 \tag{6.215}$$

for the case $p = 2$, the fact that

$$E_{\underline{X}_k}\{(\eta_k^{i(+)} - \eta_k^{i(-)})^2 | \underline{\Theta}_k\} \leq 2\sigma^2 \tag{6.216}$$

Lemma 6.7, and the Schwartz Inequality we obtain

$$\begin{aligned}
E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} &= E_{\underline{X}_k}\left\{ \left\| \sum_{i=1}^{N} \frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{e}_i) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{e}_i))\underline{e}_i}{\delta_k} \right\|^2 \middle| \underline{\Theta}_k \right\} \\
&= \sum_{i=1}^{N} \left(\frac{J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k - \delta_k \underline{e}_i)}{2\delta_k} \right)^2 \|\underline{e}_i\|^2 \\
&\quad + E_{\underline{X}_k}\left\{ \sum_{i=1}^{N} \left(\frac{\eta_k^{i(+)} - \eta_k^{i(-)}}{2\delta_k} \right)^2 \|\underline{e}_i\|^2 \middle| \underline{\Theta}_k \right\} \\
&\leq \sum_{i=1}^{N} \left[\frac{J(\underline{\Theta}_k + \delta_k \underline{e}_i) - J(\underline{\Theta}_k - \delta_k \underline{e}_i)}{2\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right. \\
&\quad \left. + \nabla J(\underline{\Theta}_k)^T \underline{e}_i \right]^2 \|\underline{e}_i\|^2 + \sum_{i=1}^{N} \frac{\sigma^2}{2\delta_k^2} \|\underline{e}_i\|^2 \\
&\leq \sum_{i=1}^{N} \left(\frac{L}{6}\delta_k^2 \|\underline{e}_i\|^3 + |\nabla J(\underline{\Theta}_k)^T \underline{e}_i| \right)^2 \|\underline{e}_i\|^2 + (N\sigma^2/2)\frac{1}{\delta_k^2} \\
&\leq \sum_{i=1}^{N} \left[(NL^2/18)\|\underline{e}_i\|^6 \delta_k^4 + 2\|\nabla J(\underline{\Theta}_k)\|^2 \|\underline{e}_i\|^2 \right] \|\underline{e}_i\|^2 + (N\sigma^2/2)\frac{1}{\delta_k^2} \\
&= 2N\|\nabla J(\underline{\Theta}_k)\|^2 + (NL^2/18)\delta_k^4 + (N\sigma^2/2)\frac{1}{\delta_k^2}
\end{aligned} \tag{6.217}$$

■

We may now relate conditions (6.212) and (6.213) with the conditions of Assump-

tion 6.3 by making the associations

$$K_1 = 1, \quad \alpha_k = 1, \ \forall k \tag{6.218}$$

$$K_2 = (NL/6), \quad \beta_k = \delta_k^2 \tag{6.219}$$

$$K_3 = 2N, \quad K_4\nu_k = (NL^2/18)\delta_k^4 + (N\sigma^2/2)\frac{1}{\delta_k^2} \tag{6.220}$$

Then Assumption 6.4 requires that the stepsize sequence $\{\rho_k\}$ and perturbation sequence $\{\delta_k\}$ be such that

$$\rho_k \geq 0, \qquad \sum_{i=1}^{\infty} \rho_k = \infty, \qquad \sum_{i=1}^{\infty} \rho_k^2 < \infty \tag{6.221}$$

$$\delta_k \geq 0, \qquad \sum_{i=1}^{\infty} \rho_k \delta_k^2 < \infty, \qquad \sum_{i=1}^{\infty} \rho_k^2 \delta_k^4 < \infty, \qquad \sum_{i=1}^{\infty} \frac{\rho_k^2}{\delta_k^2} < \infty \tag{6.222}$$

If we adopt the forms

$$\rho_k = \frac{\rho_1}{k^a}, \quad \delta_k = \frac{\delta_1}{k^b} \tag{6.223}$$

then the above conditions imply the inequalities

$$a \leq 1, \qquad a \geq 1/2 \tag{6.224}$$

$$b \geq -(1/2)a + 1/2, \quad b \geq -(1/2)a + (1/4), \quad b \leq a - (1/2) \tag{6.225}$$

which define the allowable stepsize ranges depicted in Figure 6-4. Again notice the inactive constraint.

It is interesting to note that the set of allowable exponent values for the one-sided approximation form a subset of the values allowed for the two-sided approximation. It is also interesting that although the two-sided KW approximation appeared as a special case of the unnormalized window algorithm, using a rectangular window, there is a larger region of validity for the WIN algorithm, and there are choices of steps which assure convergence of one algorithm and not the other. This is a result of the fact that we have independently established *sufficient* conditions for convergence of

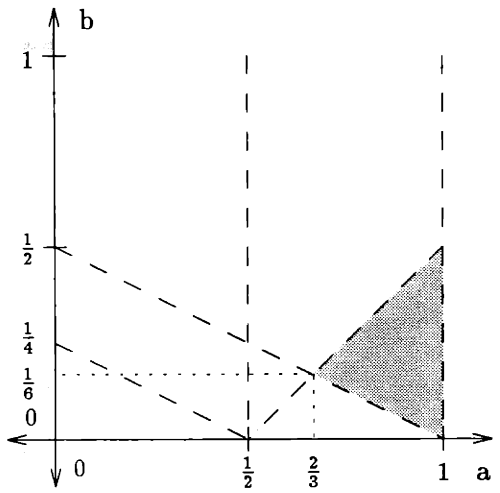


Figure 6-4: Allowable Ranges of a and b for two-sided KW.

each algorithm by different routes.

We have now demonstrated asymptotic convergence of the one-dimensional KW methods, both one-sided and two-sided, as well as the multivariable network KW algorithm, by appealing to Proposition 6.1. But since the multivariable algorithm explicitly constructs each partial derivative, we could also have shown that Proposition 6.2 applies, and that the stepsize and perturbation sequences could be chosen differently by each processor.

Random Directions Variant

In this section we demonstrate that the random directions methods KW-RD also possess the required generalized stochastic descent property. The method of proof follows in virtually identical fashion to the proofs of the standard KW techniques, once we have guaranteed certain properties of the random directions.

Recall that these techniques used only 2 samples of the function in order to directly construct the gradient estimate at each time k. One sample is taken randomly taken from the sphere of points described by $\underline{\Theta}_k + \delta_k \underline{q}_k$, where $\underline{q}_k$ is randomly chosen from the set

$$\mathcal{S}_N = \{\underline{X} \in \Re^N \mid \|\underline{X}\| = 1\} \tag{6.226}$$

In the one-sided random directions variant, the second sample is taken at the current iterate $\underline{\Theta}_k$ while in the two-sided variant the second sample is taken at $\underline{\Theta} - \delta \underline{q}_k$.

Our method of proof directly relies on the validity of the following assumptions on the random vector $\underline{q}$ (cf. [50]), both of which clearly hold for the case of uniform sampling from the unit sphere. We give the conditions to indicate that the result may be proved for more general choices of the random direction.

Assumption 6.10 (Conditions on the Random Sampling Direction)
For the random sampling directions $\underline{q}_k$ it holds for every time k that

(a) *(Nonzero expected projection onto $\underline{\Theta}$)*

$$E_{\underline{q}_k}\{(\underline{\theta}^T \underline{q}_k)^2\} \geq c\|\underline{\theta}\|^2, \; c > 0 \tag{6.227}$$

(b) *(Not too large)*

$$E_{\underline{q}_k}\{\|\underline{q}_k\|^j\} \leq a_j, \tag{6.228}$$

(c) *(Independently generated) $\underline{q}_k$ is statistically independent from $\underline{X}_k$ (equivalently η_k)*

Random Search, One-Sided Variant The proof for the one-sided random directions variant will be noted to be very similar to the proof for the one-sided KW method of the previous section. In fact, the proofs may be handled together [50], although we have separated them for clarity.

Proposition 6.8 (GSD Property: KW-RD, 1-Sided Variant)

For the step

$$\underline{Z}_k = \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{q}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k)]\underline{q}_k}{\delta_k} \tag{6.229}$$

it holds that

(i)

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} \geq c \|\nabla J(\underline{\Theta}_k)\|^2 - (a_3 L/2)\delta_k \|\nabla J(\underline{\Theta}_k)\| \tag{6.230}$$

(ii)

$$E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} \leq 2a_4 \|\nabla J(\underline{\Theta}_k)\|^2 + (a_6 L^2/2)\delta_k^2 + (2a_2\sigma^2)\frac{1}{\delta_k^2} \tag{6.231}$$

where L is the Lipschitz constant on the gradient of J, c is the constant of Assumption 6.10(a), a_3 is defined as in Assumption 6.10(b), and σ^2 is a bound on the variance of the sampling error η.

Proof.

(i) Using Lemma 6.5, Lemma 6.6, Assumption 6.10, the modified martingale difference noise assumption

$$E_{\underline{X}_k}\{(\eta_k^+ - \eta_k) | \underline{\Theta}_k, \underline{q}_k\} = 0 \tag{6.232}$$

and the Schwartz Inequality (Appendix B), we obtain

$$\begin{aligned}
\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} &= \nabla J(\underline{\Theta}_k)^T E_{\underline{X}_k, \underline{q}_k}\left\{ \frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{q}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k))\underline{q}_k}{\delta_k} \middle| \Theta_k \right\} \\
&= \nabla J(\underline{\Theta}_k)^T \left(E_{\underline{X}_k, \underline{q}_k}\left\{ \frac{(J(\underline{\Theta}_k + \delta_k \underline{q}_k) - J(\underline{\Theta}_k))\underline{q}_k}{\delta_k} \middle| \underline{\Theta}_k \right\} \right. \\
&\quad \left. + E_{\underline{X}_k, \underline{q}_k}\left\{ \frac{(\eta_k^+ - \eta_k)\underline{q}_k}{\delta_k} \middle| \underline{\Theta}_k \right\} \right) \\
&= \nabla J(\underline{\Theta}_k)^T E_{\underline{q}_k}\left\{ \frac{(J(\underline{\Theta}_k + \delta \underline{q}_k) - J(\underline{\Theta}_k))\underline{q}_k}{\delta_k} \middle| \underline{\Theta}_k \right\} \\
&= \nabla J(\underline{\Theta}_k)^T E_{\underline{q}_k}\left\{ \left[\frac{J(\underline{\Theta}_k + \delta_k \underline{q}_k) - J(\underline{\Theta}_k)}{\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{q}_k \right. \right.
\end{aligned}$$

$$
\begin{aligned}
&+\nabla J(\underline{\Theta}_k)^T \underline{q}_k \Big] \underline{q}_k | \underline{\Theta}_k \Big\} \\
&\geq -\nabla J(\underline{\Theta}_k)^T E_{\underline{q}_k}\{[(L/2)\|\underline{q}_k\|^2 \delta_k]\underline{q}_k | \underline{\Theta}_k\} + E_{\underline{q}_k}\{(\nabla J(\underline{\Theta}_k)^T \underline{q}_k)^2 | \underline{\Theta}_k\} \\
&\geq -(L/2)\delta_k E_{\underline{q}_k}\{\nabla J(\underline{\Theta}_k)^T \underline{q}_k \|\underline{q}_k\|^2 | \underline{\Theta}_k\} + c\|\nabla J(\underline{\Theta}_k)\|^2 \\
&\geq -(L/2)\delta_k E_{\underline{q}_k}\{\|\nabla J(\underline{\Theta}_k)\| \|\underline{q}_k\|^3 | \underline{\Theta}_k\} + c\|\nabla J(\underline{\Theta}_k)\|^2 \\
&\geq c\|\nabla J(\underline{\Theta}_k)\|^2 - (a_3 L/2)\delta_k \|\nabla J(\underline{\Theta}_k)\| \qquad (6.233)
\end{aligned}
$$

(ii) Using the elementary inequality

$$
\left(\sum_{i=1}^{p} a_i\right)^2 \leq p \sum_{i=1}^{p} a_i^2 \qquad (6.234)
$$

for the case $p = 2$, the fact that

$$
E_{\underline{X}_k}\{(\eta_k^+ - \eta_k)^2 | \underline{\Theta}_k, \underline{q}_k\} \leq 2\sigma^2 \qquad (6.235)
$$

Lemma 6.6, Assumption 6.10, and the Schwartz Inequality we obtain

$$
\begin{aligned}
E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} &= E_{\underline{X}_k, \underline{q}_k}\left\{\left\|\frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{q}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k))\underline{q}_k}{\delta_k}\right\|^2 \middle| \underline{\Theta}_k\right\} \\
&= E_{\underline{X}_k, \underline{q}_k}\left\{\left\|\frac{(J(\underline{\Theta}_k + \delta_k \underline{q}_k) - J(\underline{\Theta}_k))\underline{q}_k}{\delta_k} + \frac{(\eta_k^+ - \eta_k)\underline{q}_k}{\delta_k}\right\|^2 \middle| \underline{\Theta}_k\right\} \\
&= E_{\underline{q}_k}\left\{\left(\frac{J(\underline{\Theta}_k + \delta_k \underline{q}_k) - J(\underline{\Theta}_k)}{\delta_k}\right)^2 \|\underline{q}_k\|^2 \middle| \underline{\Theta}_k\right\} \\
&\quad + E_{\underline{X}_k, \underline{q}_k}\left\{\left(\frac{\eta_k^+ - \eta_k}{\delta_k}\right)^2 \|\underline{q}_k\|^2 \middle| \underline{\Theta}_k\right\} \\
&\leq E_{\underline{q}_k}\left\{\left[\frac{J(\underline{\Theta}_k + \delta_k \underline{q}_k) - J(\underline{\Theta}_k)}{\delta_k} - \nabla J(\underline{\Theta}_k)^T \underline{q}_k \right.\right. \\
&\quad \left.\left. + \nabla J(\underline{\Theta}_k)^T \underline{q}_k\right]^2 \|\underline{q}_k\|^2 | \underline{\Theta}_k\right\} + \frac{2\sigma^2}{\delta_k^2} E_{\underline{q}_k}\{\|\underline{q}_k\|^2 | \underline{\Theta}_k\} \\
&\leq E_{\underline{q}_k}\left\{\left(\frac{L}{2}\delta_k \|\underline{q}_k\|^2 + |\nabla J(\underline{\Theta}_k)^T \underline{q}_k|\right)^2 \|\underline{q}_k\|^2 | \underline{\Theta}_k\right\} + (2a_2\sigma^2)\frac{1}{\delta_k^2} \\
&\leq E_{\underline{q}_k}\left\{\left[2\left(\frac{L}{2}\delta_k \|\underline{q}_k\|^2\right)^2 + 2\left(|\nabla J(\underline{\Theta}_k)^T \underline{q}_k|\right)^2\right] \|\underline{q}_k\|^2 \middle| \underline{\Theta}_k\right\} + (2a_2\sigma^2)\frac{1}{\delta_k^2} \\
&\leq E_{\underline{q}_k}\left\{(L^2/2)\|\underline{q}_k\|^4 \delta_k^2 \|\underline{q}_k\|^2 | \underline{\Theta}_k\right\} + 2E_{\underline{q}_k}\left\{\|\nabla J(\underline{\Theta}_k)\|^2 \|\underline{q}_k\|^2 \|\underline{q}_k\|^2 | \underline{\Theta}_k\right\}
\end{aligned}
$$

$$+(2a_2\sigma^2)\frac{1}{\delta_k^2}$$

$$\leq 2a_2\|\nabla J(\underline{\Theta}_k)\|^2 + (a_6L^2/2)\delta_k^2 + (2a_2\sigma^2)\frac{1}{\delta_k^2} \tag{6.236}$$

■

The GSD bounds obtained for the KW-RD technique are very similar to those obtained for the standard KW technique, the only differences being minor changes to some of the constants. The implication of the results is that the same restrictions on the stepsize and perturbation sequences apply as for one-sided KW.

In contrast to the standard KW methods, the GSD property cannot be shown to hold componentwise for these methods. Another drawback we previously mentioned is that generation of the direction vector $\underline{q}_k$ presents a difficulty.

Random Search, Two-Sided Variant Under the same additional smoothness assumptions as were required for the two-sided KW method, the two-sided KW-RD can be shown to have virtually identical GSD bounds. Again there are only minor changes to some of the constants. Because of the similarity, we do not present the result.

It is worth commenting here that the same approach can be used to prove convergence of a wide range of stochastic search methods. For example, it follows directly from the previous development that random coordinate descent, whereby coordinate i is selected for update at time k with probability $p_{i(k)} \geq \varepsilon > 0$ converges under the same set of assumptions as the nonrandom technique.

Simultaneous Perturbation Method

As indicated in Chapter 5, the Simultaneous Perturbation technique introduced by Spall [58] is not a special case of the random directions techniques described above. Recall that the algorithm generated the step

$$\underline{Z}_k(\underline{X}_k, \underline{\Theta}_k, \underline{\Delta}_k, \delta_k) = \sum_{i=1}^{N} \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k\underline{\Delta}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k\underline{\Delta}_k)]\underline{e}_i}{2\delta_k\Delta_{i(k)}} \tag{6.237}$$

where $\underline{\Delta}_k$ is a vector with independently generated zero-mean components. We require the following conditions on $\underline{\Delta}_k$.

Assumption 6.11 (Conditions on the Perturbation Vector)
For the perturbation vector $\underline{\Delta}_k \in \Re^N$ it holds for every time k that

(a) *(Mutually Independent Components) $\Delta_{i(k)}$ is statistically independent of $\Delta_{j(k)}$ for $i \neq j$*

(b) *(Zero-Mean) $E\{\Delta_{l(k)}\} = 0,\ l = 1, \ldots, N$*

(c) *(Independently generated) $\underline{\Delta}_k$ is statistically independent from $\underline{X}_k$ (equivalently η_k), and $\{\underline{\Delta}_k\}$ is an independent identically distributed sequence*

(d) *(Bounded, Bounded Inverse First and Second Moments) There exist K_1, K_2, and K_3 such that $|\Delta_{l(k)}| \leq K_1$, $E\{|\Delta_{l(k)}^{-1}|\} \leq K_2$, and $E\{\Delta_{l(k)}^{-2}\} \leq K_3$*

We now present an alternative proof of convergence to that in [58] by characterizing the method as a generalized stochastic descent technique. The bounds can be seen to be very similar to those obtained for the two-sided KW method. We again require the additional smoothness condition Assumption 6.9.

Proposition 6.9 (GSD Property: KW-SP)

For the step

$$\underline{Z}_k = \sum_{i=1}^{N} \frac{[Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k \underline{\Delta}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k \underline{\Delta}_k)]\underline{e}_i}{2\delta_k \Delta_{i(k)}} \tag{6.238}$$

it holds that

(i)

$$\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k | \underline{\Theta}_k\} \geq \|\nabla J(\underline{\Theta}_k)\|^2 - (NLK/6)\delta_k^2 \|\nabla J(\underline{\Theta}_k)\| \tag{6.239}$$

where $K = (N-1)^3 K_1^3 K_2 + (N^3 - (N-1)^3)K_1^2$.

(ii)

$$E\{\|\underline{Z}_k\|^2 | \underline{\Theta}_k\} \leq 2N\|\nabla J(\underline{\Theta}_k)\|^2 + (NL^2K'/18)\delta_k^4 + (N\sigma^2 K_3/2)\frac{1}{\delta_k^2} \tag{6.240}$$

where $K' = (N-1)^6(\frac{K_1^6}{K_3}) + 6(N-1)^5(\frac{K_1^5}{K_2}) + 15(N-1)^4(K_1^4)$.

L is the Lipschitz constant on the gradient of J, N is the dimension of the parameter vector, K_1, K_2 and K_3 are the constants of Assumption 6.11, and σ^2 is a bound on the variance of the sampling error η.

Proof.

(i) Using Lemma 6.5, Lemma 6.7, Assumption 6.11, the modified martingale difference noise assumption

$$E_{\underline{X}_k}\{(\eta_k^+ - \eta_k^-)|\underline{\Theta}_k, \underline{\Delta}_k\} = 0 \tag{6.241}$$

and the Schwartz Inequality (Appendix B), we obtain

$$\begin{aligned}
\nabla J(\underline{\Theta}_k)^T E\{\underline{Z}_k|\underline{\Theta}_k\} &= \nabla J(\underline{\Theta}_k)^T E_{\underline{X}_k, \underline{\Delta}_k}\left\{\sum_{i=1}^{N} \frac{(Q_k(\underline{X}_k, \underline{\Theta}_k + \delta_k\underline{\Delta}_k) - Q_k(\underline{X}_k, \underline{\Theta}_k - \delta_k\underline{\Delta}_k))\underline{e}_i}{2\delta_k\Delta_{i(k)}}\middle| \Theta_k\right\} \\
&= \nabla J(\underline{\Theta}_k)^T \left(E_{\underline{X}_k, \underline{\Delta}_k}\left\{\sum_{i=1}^{N} \frac{(J(\underline{\Theta}_k + \delta_k\underline{\Delta}_k) - J(\underline{\Theta}_k - \delta_k\underline{\Delta}_k))\underline{e}_i}{2\delta_k\Delta_{i(k)}}\middle| \underline{\Theta}_k\right\}\right.
\end{aligned}$$

$$
\begin{aligned}
&+E_{\underline{X}_k,\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\frac{(\eta_k^+-\eta_k^-)\underline{e}_i}{2\delta_k\Delta_{i(k)}}\middle|\underline{\Theta}_k\right\}\Bigg)\\
&= \nabla J(\underline{\Theta}_k)^T E_{\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\frac{(J(\underline{\Theta}_k+\delta\underline{\Delta}_k)-J(\underline{\Theta}_k-\delta_k\underline{\Delta}_k))\underline{e}_i}{2\delta_k\Delta_{i(k)}}\middle|\underline{\Theta}_k\right\}\\
&= \nabla J(\underline{\Theta}_k)^T E_{\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\frac{1}{\Delta_{i(k)}}\left[\frac{J(\underline{\Theta}_k+\delta_k\underline{\Delta}_k)-J(\underline{\Theta}_k-\delta_k\underline{\Delta}_k)}{2\delta_k}\right.\right.\\
&\quad \left.\left.-\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k+\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k\right]\underline{e}_i|\underline{\Theta}_k\right\}\\
&\geq -\nabla J(\underline{\Theta}_k)^T E_{\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\frac{1}{\Delta_{i(k)}}[(L/6)\|\underline{\Delta}_k\|^3\delta_k^2]\underline{e}_i|\underline{\Theta}_k\right\}\\
&\quad +\nabla J(\underline{\Theta}_k)^T E_{\underline{\Delta}_k}\{\sum_{i=1}^{N}\frac{1}{\Delta_{i(k)}}(\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k)\underline{e}_i|\underline{\Theta}_k\}\\
&\geq -(L/6)\delta_k^2\sum_{i=1}^{N}\nabla J(\underline{\Theta}_k)^T\underline{e}_i E_{\underline{\Delta}_k}\left\{\frac{1}{\Delta_{i(k)}}\|\underline{\Delta}_k\|^3|\underline{\Theta}_k\right\}+\|\nabla J(\underline{\Theta}_k)\|^2\\
&\geq -(L/6)\delta_k^2\|\nabla J(\underline{\Theta}_k)\|\sum_{i=1}^{N}E_{\underline{\Delta}_k}\left\{\frac{1}{|\Delta_{i(k)}|}\|\underline{\Delta}_k\|^3|\underline{\Theta}_k\right\}+\|\nabla J(\underline{\Theta}_k)\|^2\\
&\geq -(L/6)\delta_k^2\|\nabla J(\underline{\Theta}_k)\|\sum_{i=1}^{N}\sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\sum_{i_3=1}^{N}E_{\underline{\Delta}_k}\left\{\frac{|\Delta_{i_1(k)}||\Delta_{i_2(k)}||\Delta_{i_3(k)}|}{|\Delta_{i(k)}|}\middle|\underline{\Theta}_k\right\}\\
&\quad +\|\nabla J(\underline{\Theta}_k)\|^2\\
&\geq -(LNK/6)\delta_k^2\|\nabla J(\underline{\Theta}_k)\|+\|\nabla J(\underline{\Theta}_k)\|^2 \qquad (6.242)
\end{aligned}
$$

where

$$K=(N-1)^3K_1^3K_2+(N^3-(N-1)^3)K_1^2 \qquad (6.243)$$

which follows from the fact that $(N-1)^3$ summands have no $|\Delta_{i(k)}|$ term in the numerator.

(ii) Using the elementary inequality

$$\left(\sum_{i=1}^{p}a_i\right)^2\leq p\sum_{i=1}^{p}a_i^2 \qquad (6.244)$$

for the case $p=2$, the fact that

$$E_{\underline{X}_k}\{(\eta_k^+-\eta_k^-)^2|\underline{\Theta}_k,\underline{\Delta}_k\}\leq 2\sigma^2 \qquad (6.245)$$

Lemma 6.7, Assumption 6.11, and the Schwartz Inequality we obtain

$$
\begin{aligned}
E\{\|\underline{Z}_k\|^2|\underline{\Theta}_k\} &= E_{\underline{X}_k,\underline{\Delta}_k}\left\{\left\|\sum_{i=1}^{N}\frac{(Q_k(\underline{X}_k,\underline{\Theta}_k+\delta_k\underline{\Delta}_k)-Q_k(\underline{X}_k,\underline{\Theta}_k-\delta_k\underline{\Delta}_k))\underline{e}_i}{2\delta_k\Delta_{i(k)}}\right\|^2\middle|\underline{\Theta}_k\right\} \\
&= E_{\underline{X}_k,\underline{\Delta}_k}\left\{\left\|\sum_{i=1}^{N}\frac{(J(\underline{\Theta}_k+\delta_k\underline{\Delta}_k)-J(\underline{\Theta}_k-\delta_k\underline{\Delta}_k))\underline{e}_i}{2\delta_k\Delta_{i(k)}}+\frac{(\eta_k^+-\eta_k^-)\underline{e}_i}{2\delta_k\Delta_{i(k)}}\right\|^2\middle|\underline{\Theta}_k\right\} \\
&= E_{\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\left(\frac{J(\underline{\Theta}_k+\delta_k\underline{\Delta}_k)-J(\underline{\Theta}_k-\delta_k\underline{\Delta}_k)}{2\delta_k\Delta_{i(k)}}\right)^2\|\underline{e}_i\|^2\middle|\underline{\Theta}_k\right\} \\
&\quad +E_{\underline{X}_k,\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\left(\frac{\eta_k^+-\eta_k^-}{2\delta_k\Delta_{i(k)}}\right)^2\|\underline{e}_i\|^2\middle|\underline{\Theta}_k\right\} \\
&\leq E_{\underline{\Delta}_k}\left\{\sum_{i=1}^{N}\frac{1}{\Delta_{i(k)}^2}\left[\frac{J(\underline{\Theta}_k+\delta_k\underline{\Delta}_k)-J(\underline{\Theta}_k-\delta_k\underline{\Delta}_k)}{2\delta_k}-\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k\right.\right. \\
&\quad \left.\left.+\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k\right]^2\|\underline{e}_i\|^2|\underline{\Theta}_k\right\}+\frac{\sigma^2}{2\delta_k^2}\sum_{i=1}^{N}E_{\underline{\Delta}_k}\left\{\left(\frac{1}{\Delta_{i(k)}}\right)^2\|\underline{e}_i\|^2|\underline{\Theta}_k\right\} \\
&\leq \sum_{i=1}^{N}E_{\underline{\Delta}_k}\left\{\left(\frac{1}{\Delta_{i(k)}}\right)^2\left(\frac{L}{6}\delta_k^2\|\underline{e}_i\|^3+|\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k|\right)^2\|\underline{e}_i\|^2|\underline{\Theta}_k\right\}+(NK_3\sigma^2/2)\frac{1}{\delta_k^2} \\
&\leq \sum_{i=1}^{N}E_{\underline{\Delta}_k}\left\{\left(\frac{1}{\Delta_{i(k)}}\right)^2\left[2\left(\frac{L}{6}\delta_k^2\|\underline{\Delta}_k\|^3\right)^2+2\left(|\nabla J(\underline{\Theta}_k)^T\underline{\Delta}_k|\right)^2\right]\|\underline{e}_i\|^2\middle|\underline{\Theta}_k\right\} \\
&\quad +(NK_3\sigma^2/2)\frac{1}{\delta_k^2} \\
&\leq \frac{L^2}{18}\delta_k^4\sum_{i=1}^{N}E_{\underline{\Delta}_k}\left\{\left(\frac{1}{\Delta_{i(k)}}\right)^2\|\underline{e}_i\|^6|\underline{\Theta}_k\right\} \\
&\quad +2\|\nabla J(\underline{\Theta}_k)\|^2\sum_{i=1}^{N}E_{\underline{\Delta}_k}\left\{\left(\frac{1}{\Delta_{i(k)}}\right)^2\|\underline{\Delta}_k\|^2|\underline{\Theta}_k\right\}+(NK_3\sigma^2/2)\frac{1}{\delta_k^2} \\
&\leq 2N\|\nabla J(\underline{\Theta}_k)\|^2+(NL^2K'/18)\delta_k^4+(N\sigma^2K_3/2)\frac{1}{\delta_k^2} \qquad (6.246)
\end{aligned}
$$

where $K'=(N-1)^6(\frac{K_1^6}{K_3})+6(N-1)^5(\frac{K_1^5}{K_2})+15(N-1)^4(K_1^4)$.

■

As these bounds differ from the bounds for the two-sided KW technique only by constants, the same choices of stepsize are sufficient for convergence.

6.4 On the Rate of Convergence

As pointed out before, establishing the fact of asymptotic convergence does not necessarily give an indication of an algorithm's usefulness, since in spite of such guarantees, the rate of convergence can be impractically slow. However, demonstrating asymptotic convergence does identify the algorithm as being reasonable in a certain sense, and provides an endorsement which would certainly be lacking were it not possible to demonstrate this property. Furthermore, in the process of proving asymptotic convergence the class of problems on which the algorithm can be expected to be successful is identified, and guidelines for designing the algorithm, for example in choosing window functions or stepsize sequences, are established.

Nevertheless, we should emphasize that the picture is not truly complete without a rate of convergence analysis, and this was beyond the scope of this report. Such analysis typically requires a unimodality assumption on J, and then the rate of convergence of the quantity $E\{J(\underline{\Theta}_k) - J(\underline{\Theta}^*)\}$ or $E\{\|\underline{\Theta}_k - \underline{\Theta}^*\|\}$ to zero is examined. If the dependence of this rate on the parameter sequences ρ_k and δ_k can be established, then optimal choices of the time-dependence in these sequences can be determined [72]. However, optimal choice of the gain coefficients ρ_1 and δ_1 depends on specific knowledge of the cost function, and cannot be performed "nonparametrically". An alternative approach [32], [4] is to examine the expected rate of convergence of the parameter sample path to the limit ODE. Another useful analysis is to establish conditions under which the iterates are asymptotically distributed normally, so that the asymptotic mean square error of the method can be established. There are also a multitude of techniques for accelerating the convergence of the algorithms, such as the use of second-order techniques, sign methods, etc. [56].

In practice, choice of the parameter sequences, and an initial starting point for the algorithm are made according to heuristic considerations, usually by exploiting observed properties of the data sequence [57].

6.5 Chapter Conclusions

In this chapter we established the asymptotic convergence of the sequence of cost realizations for both the one-dimensional and network versions of the window and KW-type training algorithms WIN, WIN-BP, KW, KW-RD and KW-SP. The key assumptions were on the differentiability and boundedness properties of the cost discussed in Chapter 3, and the assumption of an IID measurement sequence as imposed in Chapters 4 and 5. With an additional a posteriori assumption of boundedness of the parameter sequence, we demonstrated that every limit point of the sequence of threshold parameters is a stationary point (a.s.), and for a unimodal cost with a single global minimum, (a.s.) convergence to the optimal threshold value was demonstrated. The approach corresponded to Lyapunov's second method, and involved demonstrating that the algorithms possessed a common generalized stochastic descent property. This fact was used to argue that iterations of the algorithm result, on the average, in a generalized type of descent on the cost surface. The fact that the algorithms possessed a descent property of a common form allowed a single proof to be constructed which encompassed all of them.

The distinguishing feature of our descent conditions, and the reason that we termed them "generalized" stochastic descent conditions, involved the presence of a decaying bias term in the measurement of the gradient, and a step variance which had to be allowed to become unbounded in the limit. These complications were effectively handled by proper choice of the stepsize sequences. In order to demonstrate that each particular training algorithm obeyed the conditions required for application of our convergence results, we first established a bound for each algorithm which characterized the error in the measurement of the gradient. We then demonstrated, using this bound, that steps of the algorithm possessed the necessary GSD property.

The proof of convergence for the WIN algorithm relied directly on the method of construction of the coupling costs; in particular, our argument relied directly on the conditional independence of the observations and the assumption of a tree-type topology. The analysis uncovered the interesting fact it is sufficient for convergence to

employ unbiased bounded variance estimates of the coupling costs, and that the estimates of these costs can be computed on the basis of operating point estimates of 0-1 quality. So not only can interesting issues, such as the effects of varying the quality of the local models be explored, but we also find that the training can be accomplished without increasing the bandwidth of the communication channels to handle the communication of real-valued quantities. Of course, the observed rate of convergence will certainly be higher if better estimates are allowed to be communicated.

Chapter 7

Asynchronous Network Training Algorithms

The present chapter focuses on an implementation issue, namely on asynchronous implementations of the distributed synchronous network training algorithms of Chapter 5. Our goal is to provide sufficient conditions under which the convergence results of Chapter 6 continue to hold, despite relaxing the rigid timing constraints according to which the algorithms were assumed to execute. Furthermore, the techniques of this chapter provide proof of convergence for all the timing variations of the WIN-GS and KW-GS algorithms.

Our interest in allowing asynchronism in our iterative stochastic optimization schemes is motivated by our desire to enhance the modeling capability of our DBHT training methodology to better reflect the actual circumstances under which training and adaptation in distributed systems such as human organizations and biological neural networks frequently occurs. In particular, it occurs not only under conditions of local partial and uncertain information, but also without any centralized authority which provides coordinating timing mechanisms to ensure that events are carried out in an orderly manner. We believe that the mechanisms of distributed computation, applied within a suitable framework, provide a mathematical vehicle for addressing many of these issues. Specifically, such study may aid the development of a normative theory of team training in decentralized environments.

The convergence of distributed asynchronous stochastic gradient methods was first investigated by Tsitsiklis in his PhD thesis [66], leading to several publications on this topic including a paper by Tsitsiklis, Bertsekas, and Athans [67] and a section in the text by Bertsekas and Tsitsiklis [6]. Surveys of convergence theories of distributed iterative processes were also presented in [7] and [8]. Work by Kushner and Yin [33], [34] addressed the problem of distributed stochastic approximation from a weak convergence point of view, rather than the descent approach adopted by Tsitsiklis.

In this chapter, our goal is to apply the ideas of Tsitsiklis to first develop, and then experiment with, asynchronous versions of the network training algorithms of Chapter 5. In order to do so, we must first extend the results of Tsitsiklis to handle more general conditions on the steps. What requires proof is that the additional bias term in the stochastic descent assumption, and the possibly unbounded variance of the steps, do not undermine the validity of Tsitsiklis' results, which did not consider such conditions. Demonstrating that the results of Tsitsiklis hold under these more general conditions will imply that several of our previous training algorithms admit asynchronous implementations. More generally, it would expand the set of algorithms covered by Tsitsiklis' results to include KW-type search algorithms as well as the class of window algorithms.

Our proof generally follows the proof of Tsitsiklis in [6], but with several differences that result in our proof being more transparent, and better suiting our needs in this report. The price which is paid is in the generality of the results. In the first place, we again exploit our assumptions on the cost to structure the proof after the proof in Chapter 6, rather than the argument in [6]. Secondly, the results of Tsitsiklis are designed to handle the more general case in which several processors may update the same parameter, for example to increase the signal-to-noise ratio in distributed stochastic estimation problems. This generalization complicates the analysis because it requires that certain properties of the underlying agreement algorithm be established. We restrict to the specialized computation case, and thus eliminate these more complicated aspects of Tsitsiklis' proof, although it seems that our results should also hold under these more general conditions as well.

7.1 Key Issues

Before presenting the proof, we comment on certain aspects of the asynchronous setting, in particular exactly what we mean by asynchronism, how it can be introduced into our classes of algorithms, and the notions of local and global clocks.

7.1.1 Asynchronism

The general topic of asynchronism in distributed iterative algorithms, such as those for distributed optimization, is covered in depth in the text by Bertsekas and Tsitsiklis [6]. The discussion here will focus specifically on asynchronism as it would impact the network training algorithms of Chapter 5.

There are two sources of asynchronism, resulting from differences in the frequency and timing of updates at different processors, and the frequency and timing of communications between processors. We would like our algorithms to tolerate disruption and randomness in both the ordering and timing of updates as well as inter-DM communications.

With respect to the timing of updates, the timing diagrams of Chapter 5 indicated an orderly relative timing of events. But suppose it could no longer be enforced that the network of processors updated at the same time, or with the same relative timing to one another, or even at the same rate. For example, consider the typical timing diagram corresponding to an asynchronous implementation of WIN depicted in Figure 7-1. After the initial estimated operating points are obtained the processors are in various states of estimating, updating, or sitting idle. Processors may make several consecutive updates without intervening estimation periods. Note that the type of asynchronism shown in Figure 7-1 is sufficient to describe all the timing variations of algorithm WIN-GS. In the WIN algorithms, the local nature of the feedback allows the DMs to move through their measurement sequences at possibly different rates. We will have more to say about Figure 7-1 momentarily.

For KW-Type algorithms, an entire network decision process is required to sample the team error surface and update any parameter. Thus, a maximum pace is

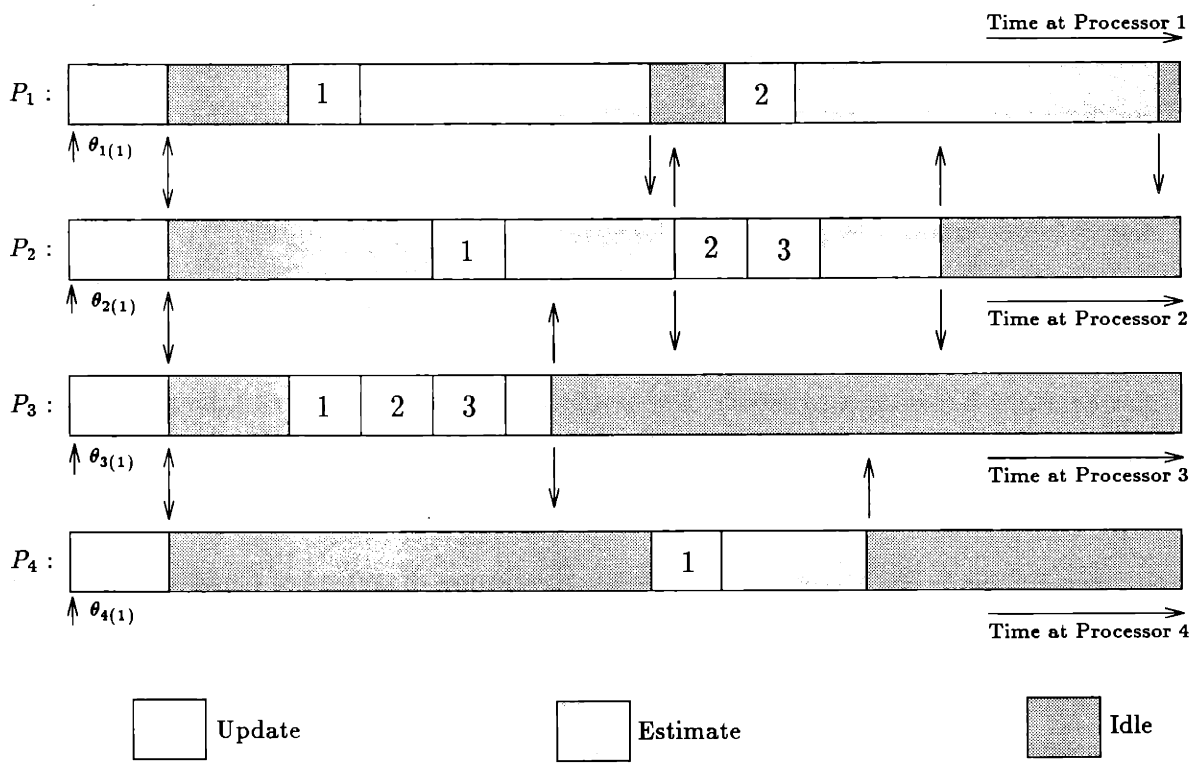


Figure 7-1: Typical Timing Diagram for an Asynchronous Implementation of the WIN Algorithm for a network of 4 processors. Vertical arrows indicate communications of local operating point information to other processors.

established by the arrival of network measurements. Asynchronism in the updates is restricted to be of the variety in which each processor decides whether or not to update on each decision cycle. A typical timing diagram corresponding to an asynchronous implementation of the one-sided KW method is shown in Figure 7-2. Here, only a (possibly random) subset of the threshold parameters are updated each time a network update is performed. Note that the type of asynchronism indicated in the figure is clearly capable of describing all the timing variations of algorithm KW-GS. One difference of this type of asynchronism from the type allowed for WIN algorithms is that some coordination between the set of updating processors is still required, since the updating processors must each obtain samples of the cost for which only their parameter is perturbed. Thus, it is necessary that a currently sampling processor be identified. This could be accomplished by a token passing scheme. Despite this complication, from the point of view of modeling and running numerical experiments, this notion of asynchronism may still be useful. We omit consideration of asynchronous variants of the remaining KW-Type algorithms at present.

The second source of asynchronism is in the relative timing and frequency of communication. Since KW-Type algorithms require no communication, this type of asynchronism has meaning only for the WIN algorithms. Recall that the WIN algorithms were model-dependent, with local updates depending on evaluating cost coefficients which were functions of the current operating points of the network DMs. Suppose that at the end of every estimation phase, each processor transmits its current operating point to those processors that require it, as indicated in Figure 7-1 by the vertical arrows. In the scheme depicted, some processors may be computing local updates on the basis of *outdated information*, as is the case for the first update depicted for processor 2, which is based on outdated information with respect to the current operating points of both processors 1 and 3. The situation may be exacerbated if the communications, rather than always occurring immediately after estimation periods, occur haphazardly, for example at random times, or if the communications are subject to delays. Each processor has only the latest versions of the operating points on which to base the estimates of its coupling costs, meaning that it operates

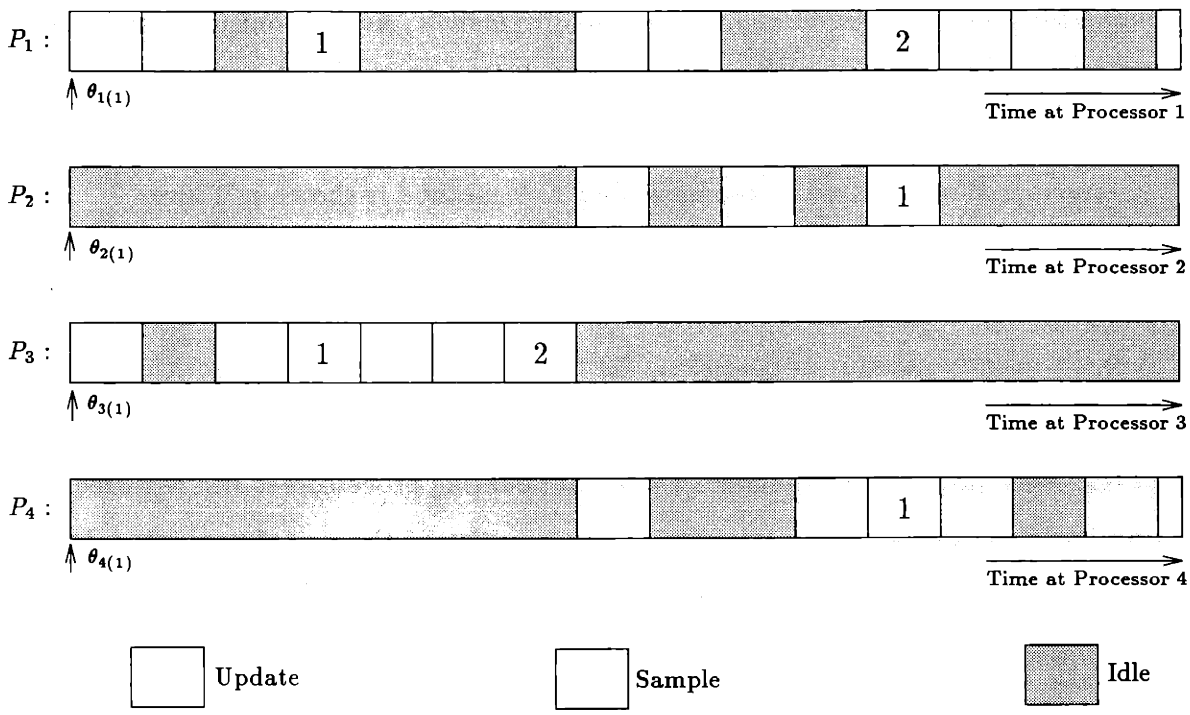


Figure 7-2: Typical Timing Diagram for an Asynchronous Implementation of One-Sided KW Algorithm by a network of 4 processors.

on the basis of an outdated model of the rest of the network. The processors simply retain and continue to use the most up-to-date information available to them in order to perform the updates. It is easy to imagine that these updates based on incorrect outdated information might even be counterproductive.

It is certainly not clear that these types of asynchronism do not act to destroy the convergence properties that the training algorithms were shown to possess when executed synchronously. In fact, it has previously been established by Bertsekas and Tsitsiklis that no deterministic or stochastic gradient algorithm can tolerate an unbounded amount of asynchronism [6]. However, as long as a finite bound is imposed on the amount of asynchronism, convergence of decreasing stepsize gradient-type algorithms can be established. In particular, the required assumptions are (cf.[6]) that there exists a finite constant $B > 0$ such that:

(1) Each processor performs an update *at least once* during any time interval of length B

(2) The information used by any processor is outdated by at most B time units

Notice the use of a common constant B to upper bound both the time between each processor's updates and the amount by which information at any processor can be outdated. This is not critical to the proof; a single constant is adopted only for notational convenience. An asynchronous algorithm obeying these assumptions is termed *partially asynchronous*. The bound B quantifies the maximum amount by which the activities of the processors can be uncoordinated.

7.1.2 Clock

The distinct notions of local and global clock provide useful analysis tools for distributed iterative algorithms. A global clock is a timing mechanism to which all processors have access, and by which the processors can coordinate their activities with respect to one another. The global clock can be thought of as providing an absolute time reference, or a measure of true or real time, although it is not necessary that it correspond to real time. It may actually be an index of all the events in the distributed system, which may be occurring at nonuniform time intervals. The important thing is that it represents time from the point of view of an external observer of the system.

The notion of global clock we adopt is that it is an index of all the events of interest in the network. Events of interest would correspond to updates, communications, arrival of measurements, etc. Thus, every event of interest corresponds to some index of the global clock. We denote the global clock with the integer time variable k.

In contrast, a local clock measures time, or equivalently indexes events, as viewed from a given processor, and its value is assumed known only to that processor. In asynchronous algorithms, we would like to remove any reference of local to global time, so that the processors may not coordinate with global time, and so that local time at the various processors may evolve at possibly different rates. For example, if the local clock at one processor were running significantly faster than the local clock at another, the processor with the faster clock might make substantially more updates

in the same interval of global clock time than the processor with the slower clock.

We adopt a simple notion of local clock; the local clock at processor i is simply a counter specifying the number of times processor i has performed an update[1]. Since the global clock is assumed to index every event of interest in the network, each local update occurs on some value of the global clock, it is simply unknown to the processor what that value is.

The notion of global time is particularly useful for analysis, exactly because from it we can obtain a common reference for all events in the network. With one additional assumption, we can replace each of the local times with global time in the analysis. Namely, we require that each local clock does not run arbitrarily fast or slow with respect to the global clock. In other words, the local and global clocks run in the same time scale, and are always within a constant factor of one another. If we let $k_i(k)$ denote the value of the local time index k_i at processor i when k is the global time. Then we require that for each processor i there exist constants K_1^i and K_2^i such that an equation of the form

$$K_1^i k \leq k_i(k) \leq K_2^i k, \tag{7.1}$$

holds for all i, k. This allows us to express locally time-dependent terms in terms of global time.

Our scheme of taking the local clock at i to represent the number of updates which have been made by processor i (plus one) is assured of obeying (7.1), so long as the (global) time between the processor updates is bounded above and below by positive constants. To illustrate, suppose that an update at i is completed at least once every B global time units. Then

$$\max\left\{1, \left\lfloor \frac{k}{B} \right\rfloor\right\} \leq k_i(k) \leq k \tag{7.2}$$

[1]There is a minor technical issue here in that our algorithms have been defined for positive integer times. Thus, strictly speaking, we must define the local counter to be the number of local updates + 1.

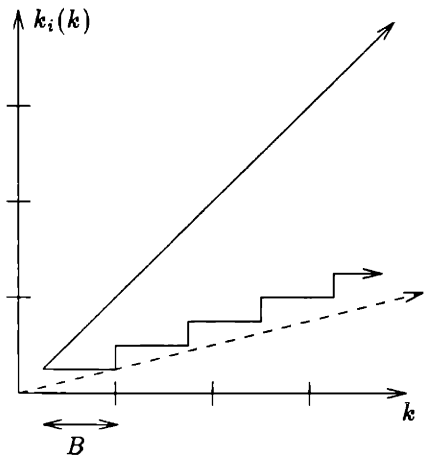


Figure 7-3: Linear Bounds on Local Clock.

where the notation $\lfloor \cdot \rfloor$ denotes truncation. Pictorially, the situation is as shown in Figure 7-3.

Clearly (7.1) holds with the associations $K_1^i = 1/B$ and $K_2^i = 1$.

To see how this is used to express locally time-dependent terms in terms of global time, consider the local stepsize rule at processor i given by

$$\rho_k^i = \frac{\rho_1^i}{(k_i(k))^a} \tag{7.3}$$

where ρ_k^i is the stepsize used by processor i to update parameter $\theta_{i(k)}$ at global time k, ρ_1^i is the initial stepsize for processor i, $k_i(k)$ is a local counter representing the number of updates performed by processor i up to time k, and a is a constant exponent. Suppose that an update is completed at processor i at least once every B global time units. Then, if we define the globally time dependent sequence

$$\rho_k = \frac{\rho_1}{k^a} \tag{7.4}$$

we can relate local and global time per

$$K_3 \rho_k \leq \rho_k^i \leq K_4 \rho_k \tag{7.5}$$

where

$$K_3 = \rho_1^i/\rho_1, \quad K_4 = \rho_1^i/\left(\left(\frac{1}{B}\right)^a \rho_1\right) \tag{7.6}$$

These issues have particular significance in our analysis because the generalized stochastic descent assumptions under which we attempt to prove convergence also include bias and step variance terms with dependencies on local counters. These terms must be expressed in terms of global time in order to proceed with the analysis.

7.2 A Model of Distributed Training

It is useful at this point to introduce a general model of distributed training which encompasses all of the algorithms of this report. The formalism allows us to specify, for each instant of global time, the activities of every processor in the network. In addition, describing the training algorithms with a single set of notation will facilitate the discussion of asynchronous training.

We introduce $\underline{\Psi}_k^i$ to denote a vector of so-called *endogenous* measurements available to processor i at time k, where the term endogenous refers to the fact that the measurements are generated by some activity within or by the network. A vector of endogenous measurements is maintained by each processor. The endogenous measurement vector $\underline{\Psi}_k^i$ is a function of the total (exogenous) network measurement vector $\underline{X}_k$ and parameter vector $\underline{\Theta}_k$ that suitably represents information concerning the state of the rest of the network such that processor i can update its parameter $\Theta_{i(k)}$ in the distributed setting.

We first consider the synchronous case, where k indexes updates. For the WIN algorithm, the vector of endogenous measurements available to processor i at time k is given by

$$\underline{\Psi}_k^i = \begin{bmatrix} (\hat{P}^1_{F(k)}, \hat{P}^1_{D(k)}) \\ (\hat{P}^2_{F(k)}, \hat{P}^2_{D(k)}) \\ \vdots \\ (\hat{P}^{M_i}_{F(k)}, \hat{P}^{M_i}_{D(k)}) \end{bmatrix} \tag{7.7}$$

where $\underline{\Psi}_k^i$ is a vector containing the $M_i - 1$ estimated network operating points from

the other DMs required by processor i to update its threshold parameter $\theta_{i(k)}$, as well as its most recent estimate of its own current operating point. We treat each row of the endogenous measurement vector as a single component, so that the jth component is given by

$$\Psi^i_{j(k)} = [(\hat{P}^j_{F(k)}, \hat{P}^j_{D(k)})] \tag{7.8}$$

For notational convenience we assume that the ith component of $\underline{\Psi}^i_k$ corresponds to the estimate of the processor's own operating point. Components of $\underline{\Psi}^i_k$ corresponding to the operating points of other network DMs are updated by replacing old information with newly received information when it arrives. The operating point corresponding to the processor's own threshold parameter can be viewed as being updated by a self-generated measurement. In terms of the new notation, the steps $Z_{i(k)}$ at processor i updating component i at time k that were formerly indicated with the argument vector

$$Z_{i(k)}(X_{i(k)}, \hat{\lambda}^i_{0(k)}, \hat{\lambda}^i_{1(k)}, \Theta_{i(k)}, \delta_k) \tag{7.9}$$

are expressed in the present notational scheme with the argument

$$Z_{i(k)}(X_{i(k)}, \underline{\Psi}^i_k(\underline{\Theta}_k), \Theta_{i(k)}, \delta_k) \tag{7.10}$$

where we have made the dependence of $\underline{\Psi}^i_k$ on $\underline{\Theta}_k$ explicit.

For the synchronous KW-type algorithms, the endogenous measurements obtained by the processors are the team decisions which are used to locally infer the sample Q of the cost surface. For example, for the two-sided KW algorithm, the endogenous measurements at a processor i would correspond to the values of the team decision obtained in response to perturbations up and down in its threshold parameter given by

$$\underline{\Psi}^i_{i(k)} = (U^{i+}_{Team(k)}, U^{i-}_{Team(k)}) \tag{7.11}$$

where for notational convenience we have written this information as corresponding to the ith component of $\underline{\Psi}^i_k$. The other components are left undefined. We do this in order to more easily describe the WIN and KW algorithms in the same notational

framework. The measurements are updated only when new samples of the cost function are obtained. In terms of the new notation, the steps $Z_{i(k)}$ at processor i updating component i at time k that were formerly indicated with the argument

$$Z_{i(k)}(X_{i(k)}, U^{i+}_{Team(k)}, U^{i-}_{Team(k)}, \Theta_{i(k)}, \delta_k) \tag{7.12}$$

are expressed in the present notation with the argument

$$Z_{i(k)}(X_{i(k)}, \underline{\Psi}^i_k(\underline{\Theta}_k), \Theta_{i(k)}, \delta_k) \tag{7.13}$$

where again we have made the dependence of $\underline{\Psi}^i_k$ on $\underline{\Theta}_k$ explicit.

Thus, introduction of the concept of an endogenous measurement has allowed the functional dependence of the steps of the two different classes of algorithms to be expressed in the same form. Dependence of the local updates on information regarding the current state of the rest of the network is completely captured through the endogenous measurement.

7.2.1 Local vs. Global Information

Consider a generic specialized distributed optimization scheme, which is simpler to describe than our network training algorithms, in which each local update depends directly on the value of the other network parameters, and the network parameters themselves are the communicated quantities. In the case of specialized computation, the processors iterate on the parameter vector $\underline{\Theta}_k$, where the ith processor updates component $\underline{\Theta}_{i(k)}$. The overall state of the distributed specialized computation is summarized by the vector

$$\underline{\Theta}_k = [\Theta_{1(k)}, \Theta_{2(k)}, \ldots, \Theta_{N(k)}]^T \tag{7.14}$$

where each component of the summary vector corresponds to the value of the component at the processor updating that component. Suppose that processor i maintains a *local version* of the overall network parameter vector $\underline{\Theta}_k$, denoted $\underline{\Theta}^i_k$, where

$\Theta^i_{i(k)} = \Theta_{i(k)}$. The processor must rely on communication from the other processors in the network to keep coordinates $\Theta^i_{j(k)}$, $j \neq i$ of its local copy up to date. Because it is responsible for updating component $\Theta_{i(k)}$, it always has the latest version of this parameter available. In synchronous algorithms, communication occurs after every update, so that $\underline{\Theta}^i_k = \underline{\Theta}_k$, $\forall i, k$. However, it is clear that in the presence of asynchronism, the local copy $\underline{\Theta}^i_k$ can become out of date with respect to the current value of the summary vector $\underline{\Theta}_k$. This might occur because updates are more frequent than communications, or because of communication delays. Thus, in asynchronous algorithms the local copies are generally not in agreement with one another or the global vector. A result of this is that local steps will usually be computed on the basis of old information, and will not be the same as steps computed based on the most current information. Specifically, steps in the generic algorithm are computed based on the local versions of the parameter vector as in $Z_{i(k)}(\underline{\Theta}^i_k)$.

This generic description of local and global information does not correspond exactly to our network training algorithms since:

(1) local updates do not depend directly on the network parameter vector

(2) local copies of the network parameter vector are not maintained by each processor

(3) the parameters themselves are not the quantities which are communicated (in the WIN algorithms)

The concept of an endogenous measurement was introduced to reconcile these differences. The purpose was to show that, for analysis purposes, the training algorithms can be described by the generic model, provided that the proper notational associations are made. Specifically, for our training algorithms the above statements are all true if the network parameter vector is replaced by the endogenous measurement vector.

To clarify, note that in the asynchronous WIN setting each processor may be updating on the basis of operating point information for another processor which is out of date with respect to that processor's current threshold value. For example, suppose we define a variable $1 \leq \kappa^i_j(k) \leq k$ which denotes the time at which operating

point information which is received by processor i from processor j at time k was actually transmitted by j. Then, until the next communication of operating point from j arrives, processor i operates at times $k \geq k'$ using the information

$$\Psi^i_{j(k)} = \left(\hat{P}^j_F(\theta_{j(\kappa^i_j(k'))}), \hat{P}^j_D(\theta_{j(\kappa^i_j(k'))})\right) \tag{7.15}$$

which depends on the value of the threshold that was used by processor j to obtain the estimate. Thus, for analysis purposes, if we imagine that the processors simultaneously transmitted the value of the threshold associated with the operating point estimate along with the estimate, and that this information was stored locally as part of a local parameter vector, then at time k the endogenous measurement vector at i can be viewed as a function of a local parameter vector $\underline{\theta}^i_k$ of the form $\underline{\Psi}^i_k(\underline{\Theta}^i_k)$.

For the KW algorithm, the situation is a bit different since there is no notion of local information to speak of. When the cost is sampled, the sample always depends on the *current* team threshold settings. Even under the type of asynchronism indicated in the timing diagram of Figure 7-2, it holds that $\underline{\Psi}^i_k(\underline{\Theta}_k)$. However, we can think of the local update as depending on a locally maintained parameter vector so long as we take

$$\underline{\Theta}^i_k = \underline{\Theta}_k, \ \forall i \tag{7.16}$$

Note that this also applies for the WIN-GS and KW-GS algorithms, for which there is also no notion of outdated information. These algorithms all represent varieties of pure update asynchronism.

Thus, we see that while our training algorithms are not strictly described by the generic distributed scheme, for analysis purposes they may be viewed as being equivalent. In particular, we may speak of local versions of the parameter vector being maintained and communicated, while keeping in mind that for our training algorithms, the dependence of the local updates on the current state of the rest of the network is actually reflected in the endogenous measurements.

7.2.2 The Algorithmic Description

With the introduction of endogenous measurements, all of our distributed training algorithms may be viewed as special cases of the the following algorithmic description. The behavior of each processor $i = 1, 2, \ldots, N$ is completely described by the following set of coupled update equations.

Endogenous Measurement Update:

$$\Psi^i_{j(k+1)} = \begin{cases} \Psi^j_{j(\kappa^i_j(k))} & k \in \mathcal{K}^i_j, j \neq i \\ f(\Psi^i_{j(k)}) & k \in \mathcal{K}^i_j, j = i \\ \Psi^i_{j(k)} & k \notin \mathcal{K}^i_j \end{cases} \tag{7.17}$$

where $\mathcal{K}^i_j$ denotes the set of global time indices at which processor i receives a communication from processor j, $\kappa^i_j(k)$ denotes the time at which information received by i from j at time k was actually transmitted.

Component $i = j$ corresponds to the most current estimate of the processor's own operating point in the case of WIN algorithms, and to the values of U_{Team} obtained by the processor in the case of the KW algorithms. The mapping f indicates generation of a self-generated measurement. In the case of WIN algorithms, this might correspond to updating the previous relative frequency estimate of the processor's operating point, while for the KW algorithms, f corresponds to obtaining a sample of the cost. Self-generated measurements at processor i occur at times $k \in \mathcal{K}^i_i$.

The additional components $i \neq j$ have meaning only for the WIN algorithms. They represent the most recent estimates of the current operating points of the other network DMs.

Parameter Update:

$$\theta_{i(k+1)} = \begin{cases} \theta_{i(k)} - \rho^i_k Z_{i(k)}(X_{i(k)}, \underline{\Psi}^i_k, \Theta_{i(k)}, \delta_k) & k \in \mathcal{K}^i \\ \theta_{i(k)} & \text{otherwise} \end{cases} \tag{7.18}$$

where $\mathcal{K}^i$ denotes the set of global time indices at which the ith component is updated.

The above description is suitable for describing both synchronous and asynchronous

versions of the training algorithms, where the differences lie in the structure imposed on the sets $\mathcal{K}_j^i$, $\mathcal{K}^i$, and on the delays $\kappa_j^i(k)$.

The above description is notationally cumbersome, and more specific than we require to analyze the algorithms. Thus, we will adopt the simpler generic description for the remaining analysis of this chapter. We have previously described the sense in which these schemes may be considered to be equivalent.

In the generic distributed algorithm, a local version of the parameter vector is maintained and updated by processor i according to

$$\theta_{j(k+1)}^i = \begin{cases} \theta_{j(\kappa_j^i(k))}^j & k \in \mathcal{K}_j^i,\ j \neq i \\ \theta_{i(k)}^i - \rho_k^i Z_{i(k)}(\underline{\theta}_k^i) & k \in \mathcal{K}^i,\ j = i \\ \theta_{j(k)}^i & \text{otherwise} \end{cases} \tag{7.19}$$

where $\mathcal{K}_j^i$ denotes the set of global time indices at which processor i receives a communication from processor j, and $\mathcal{K}^i$ denotes the set of global time indices at which processor i updates its own component.

We wish to note that much of the previous notation was adapted from Bertsekas and Tsitsiklis [6].

7.2.3 Notational Summary

Before presenting the main proof, we summarize the notation of the previous section for easy reference.

k : integer-valued global time index

$\underline{\theta}_k$: summary (global) parameter vector

$\theta_{i(k)}$: value of parameter θ_i at global time k

$\underline{\theta}_k^i$: version of the parameter vector $\underline{\theta}_k$
held by processor i at global time index k

$\mathcal{K}^i$: set of global time indices at which the ith component is updated

$\kappa_j^i(k)$: amount by which information used in an update of θ_i

which comes from processor j is outdated, i.e., the time at which the information was sent to i from j

$\mathcal{K}_j^i$: set of global time indices at which processor i receives a communication from processor j

B : asynchronism measure

(7.20)

An additional point of notation is that we define the set

$$\mathcal{F}_k = \{Z_{i(\kappa)} | i \in \{1, \ldots, N\}, \kappa < k\} \tag{7.21}$$

which, given the assumption of deterministic initial conditions, represents the only source of randomness in the algorithm up to time k. This set may therefore be viewed as a representation of the entire history of the algorithm up to the moment that the update directions $Z_{i(k)}$ are to be generated.

7.3 Main Proof: Convergence of Distributed Generalized Stochastic Descent Iterations under Partial Asynchronism

The general strategy in proving convergence of the asynchronous iterations is as follows. We identify the vector

$$\underline{\theta}_k = [\theta_{1(k)}, \theta_{2(k)}, \ldots, \theta_{N(k)}]^T \tag{7.22}$$

as summarizing the current global state of the distributed computation since it contains the most up-to-date version of each parameter. We then impose partial asynchronism assumptions which ensure that the local versions of each parameter vector cannot be too far from this global vector. If we then assume that behavior of each of

the processors is reasonable (generalized stochastic descent) with respect to its local information, then it can be argued that the entire system is doing approximately the correct thing. In particular, if behavior of the sequence of costs on iterates of the global (summary) vector can be shown to be favorable, then favorable behavior of entire distributed asynchronous algorithm can be guaranteed as well. Our characterization of favorable behavior will again be convergence of the sequence of costs as demonstrated by martingale arguments. The primary difference with the method of proof in Chapter 6 is the presence of an additional error term due to asynchronism that must be controlled.

We now present the main result of this chapter, which specifies sufficient conditions under which convergence of the type described in Chapter 6 is preserved in the presence of partial asynchronism.

Let Assumptions 6.1, 6.2 on the cost and Markov nature of the process hold as before. We make the following assumption of partial asynchronism in order that the deleterious effects of asynchronism may be bounded. The assumption represents the only restriction we place on the specific timing of the updates and communications.

Assumption 7.1 (Partial Asynchronism)

There exists an integer $B > 0$, termed the asynchronism measure, such that

(a) *(Frequency of Updates) For every processor i and for every global time index $k \geq 0$, at least one of the elements of the set*

$$\{k, k+1, \ldots, k+B-1\} \tag{7.23}$$

belongs to the set $\mathcal{K}^i$

(b) *(Outdated Information) There holds*

$$\max\{1, k-B+1\} \leq \kappa_j^i(k) \leq k \tag{7.24}$$

for all i, j and $k \geq 0$.

Notice that the choice $B = 1$ reduces to the synchronous case since it implies that $\mathcal{K}^i = \mathbf{Z}^+ = \{1, 2, \ldots\}$ and that $\kappa_j^i(k) = k$, $\forall k$.

We replace Assumptions 6.3 and 6.5 with the following assumption, which places restrictions on the updates along each coordinate. The assumption requires that the expected step direction along each coordinate, given the past history of the algorithm, and *with respect to local information*, is in a generalized stochastic descent direction. It is in this sense that the behavior of each of the processors in the distributed algorithm is required to be reasonable.

Assumption 7.2 (GSD Property Along Each Coordinate)
There exist nonnegative constants $K_1^i, K_2^i, K_3^i, K_4^i$ *and nonnegative sequences* $\{\alpha_k^i\}, \{\beta_k^i\}, \{\nu_k^i\}$ *such that for all processors* i *and times* $k \in \mathcal{K}^i$

$$\begin{aligned} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) E\{Z_{i(k)} | \mathcal{F}_k\} &\geq K_1^i \alpha_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right|^2 - K_2^i \beta_k^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right|, \ \alpha_k^i, \beta_k^i \geq 0 \\ E\{|Z_{i(k)}|^2 | \mathcal{F}_k\} &\leq K_3^i \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right|^2 + K_4^i \nu_k^i, \ \nu_k^i \geq 0 \end{aligned} \tag{7.25}$$

hold w.p.1. Furthermore, $Z_{i(k)} = 0$ *for all* $k \notin \mathcal{K}^i$.

We have already demonstrated in Chapter 6 that Assumption 7.2 holds for the WIN and WIN-GS algorithms, since each partial derivative was constructed explicitly. Thus, we may immediately discuss asynchronous versions of these algorithms with no extra work concerning specifically how the updates are generated. Although it may also be possible to demonstrate convergence of a suitably defined asynchronous variant of WIN-BP, we omit consideration of this possibility in this report. For the KW and KW-GS algorithms Assumption 7.2 also holds, again because each partial derivative is explicitly constructed. However, it is not readily apparent that the KW-RD and KW-SP algorithms obey the assumption, so we do not consider asynchronous versions of them.

We assume that each local update stepsize ρ_k^i, and the sequences α_k^i, β_k^i, and ν_k^i of Assumption 7.2, may be related to sequences depending on global time as follows.

Assumption 7.3 (Local Stepsizes)

Let $\{\rho_k\},\{\alpha_k\},\{\beta_k\},\{\nu_k\}$, be sequences obeying Assumption 6.4. Then we assume that there exist constants $K_5 - K_{12}$ such that the local sequences

$$\begin{aligned} K_5\rho_k \leq \rho_k^i \leq K_6\rho_k \\ K_7\alpha_k \leq \alpha_k^i \leq K_8\alpha_k \\ K_9\beta_k \leq \beta_k^i \leq K_{10}\beta_k \\ K_{11}\nu_k \leq \nu_k^i \leq K_{12}\nu_k \end{aligned} \tag{7.26}$$

hold for all i.

As previously discussed, these conditions are easily verified under Assumption 7.1 assuming that local time is generated using a counter of the number of local updates.

We find that the above conditions are sufficient to draw the *same* conclusions regarding asymptotic convergence of the algorithm as in Chapter 6. This fact is perhaps surprising in view of the extremely weak structure we have imposed on the frequency of updates and timing of communications.

Proposition 7.1 (Convergence of Partially Asynchronous GSD Iterations) *We assume specialized computation, that the set $\mathcal{K}^i$ is infinite for every processor i, and that Assumptions 6.1, 6.2, 7.1, 7.2, and 7.3 hold. Then it follows that:*

(a) $\lim_{k\to\infty} J(\underline{\Theta}_k) = J$ *(a.s.), for some random variable* J

(b) $\liminf_{k\to\infty} \|\nabla J(\underline{\Theta}_k)\| = 0$ *(a.s.)*

where $\underline{\Theta}_k$ *is the summary vector*

$$\underline{\Theta}_k = [\Theta_{1(k)}, \ldots, \Theta_{N(k)}]^T \tag{7.27}$$

Proof. We prove the result under the assumption that $\theta_{i(1)} = 0$ for all i. Only minor modifications to the following argument to cover the general case.

For simplicity, we do away with the local stepsizes ρ_k^i by defining the modified step

$$\bar{Z}_{i(k)} = \frac{\rho_k^i}{\rho_k} Z_{i(k)} \tag{7.28}$$

and viewing $\bar{Z}_{i(k)}$ as the update direction at processor i, with stepsize ρ_k. Under Assumption 7.3, we can show that Assumption 7.2 also holds for the modified step $\bar{Z}_{i(k)}$, with possibly different choices of the constants $K_1^i - K_4^i$ and sequences $\{\alpha_k^i\}, \{\beta_k^i\}$ and $\{\nu_k^i\}$. Thus, without loss of generality, we can and will assume that $\rho_k^i = \rho_k$ for all i and k ([2]).

This modification allows us to write the recursion

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k \tag{7.29}$$

for the summary vector $\underline{\theta}_k$, where $Z_{i(k)} = 0$, $\forall k \notin \mathcal{K}^i$.

We will need the following two lemmas.

[2]For notational convenience we do not carry the overbars throughout.

Lemma 7.1 (System-wide GSD Properties)

(a) *If $k \in \mathcal{K}^i$, then there exists a positive constant K_1, a nonnegative constant K_2 and nonnegative sequences $\{\alpha_k\}$ and $\{\beta_k\}$ such that*

$$E\left\{\sum_{i=1}^{N}\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)Z_{i(k)}\middle|\mathcal{F}_k\right\} \geq K_1\alpha_k\sum_{\{i|k\in\mathcal{K}^i\}}\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|^2 - K_2\beta_k\sum_{\{i|k\in\mathcal{K}^i\}}\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right| \quad (a.s.) \tag{7.30}$$

(b) *If $k \in \mathcal{K}^i$, then there exist nonnegative constants K_3, K_4, and a nonnegative sequence $\{\nu_k\}$ such that*

$$E\left\{\sum_{i=1}^{N}Z_{i(k)}^2\middle|\mathcal{F}_k\right\} \leq K_3\sum_{\{i|k\in\mathcal{K}^i\}}\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|^2 + K_4\nu_k \quad (a.s.) \tag{7.31}$$

Proof.

(a)

$$\begin{aligned}
E\left\{\sum_{i=1}^{N}\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)Z_{i(k)}|\mathcal{F}_k\right\} &= \sum_{\{i|k\in\mathcal{K}^i\}}\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)E\{Z_{i(k)}|\mathcal{F}_k\} \\
&\geq \sum_{\{i|k\in\mathcal{K}^i\}}\left(K_1^i\alpha_k^i\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|^2 - K_2^i\beta_k^i\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|\right) \\
&\geq K_1'K_7\alpha_k\sum_{\{i|k\in\mathcal{K}^i\}}\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|^2 \\
&\quad -K_2'K_{10}\beta_k\sum_{\{i|k\in\mathcal{K}^i\}}\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right| \quad (a.s.)
\end{aligned} \tag{7.32}$$

where

$$K_1' = \min_i\{K_1^i; i=1,\ldots,N\}, \quad K_2' = \max_i\{K_2^i; i=1,\ldots,N\} \tag{7.33}$$

and K_7 and K_{10} are the constants of Assumption 7.3. If we now make the assignment

$$K_1 = K_1'K_7, \quad K_2 = K_2'K_{10} \tag{7.34}$$

part (a) of the lemma is proved.

(b)

$$\begin{aligned} E\{\sum_{i=1}^{N}|Z_{i(k)}|^2|\mathcal{F}_k\} &= \sum_{i=1}^{N}E\{|Z_{i(k)}|^2|\mathcal{F}_k\} \\ &\leq \sum_{\{i|k\in K^i\}}\left(K_3^i\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|^2+K_4^i\nu_k^i\right) \\ &\leq K_3\sum_{\{i|k\in K^i\}}\left|\frac{\partial J}{\partial\theta_i}(\underline{\Theta}_k^i)\right|^2+NK_4'K_{12}\nu_k \ \ (a.s.) \end{aligned} \tag{7.35}$$

where

$$K_3=\max_i\{K_3^i;i=1,\ldots,N\},\quad K_4'=\max_i\{K_4^i;i=1,\ldots,N\} \tag{7.36}$$

and K_{12} is the constant of Assumption 7.3. If we now make the assignment

$$K_4=NK_4'K_{12} \tag{7.37}$$

part (b) of the lemma is proved.

□

We now obtain a bound, based on our assumption of partial asynchronism, on how far off from the summary vector $\underline{\theta}_k$, the version $\underline{\theta}_k^i$ at processor i can be.

Lemma 7.2 (Errors due to Asynchronism)

For every time $k\geq 1$, and all processors i it holds that

$$\|\underline{\Theta}_k-\underline{\Theta}_k^i\|\leq\sum_{\kappa=\max\{1,k-B\}}^{k-1}\rho_\kappa\|\underline{Z}_\kappa\| \tag{7.38}$$

where B is the bound on outdated information of Assumption .

Proof. We consider the jth component of the quantity $\|\underline{\Theta}_k - \underline{\Theta}_k^i\|$ to obtain

$$\begin{aligned} |\Theta^i_{j(k)} - \Theta_{j(k)}| &= |\Theta_{j(\kappa^i_j(k))} - \Theta_{j(k)}| \\ &= |\sum_{\kappa=\kappa^i_j(k)}^{k-1} \rho_\kappa Z_{j(\kappa)}| \\ &\leq \sum_{\kappa=\max\{1,k-B\}}^{k-1} |\rho_\kappa Z_{j(\kappa)}| \\ &= \sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa |Z_{j(\kappa)}| \end{aligned} \tag{7.39}$$

This inequality holds componentwise. Using this and the triangle inequality we obtain

$$\|\underline{\Theta}_k - \underline{\Theta}_k^i\| \leq \sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa \|\underline{Z}_\kappa\| \tag{7.40}$$

□

Notice that if the asynchronism measure $B = 1$ corresponding to the synchronous case, then we obtain a bound

$$\|\underline{\Theta}_k - \underline{\Theta}_k^i\| \leq \rho_{k-1} \|\underline{Z}_k\| \tag{7.41}$$

which is certainly true since in fact $\|\underline{\Theta}_k - \underline{\Theta}_k^i\| = 0$ because the local information is always kept current. Also note that this bound on the errors due to asynchronism applies only to the WIN algorithms since only they incorporate notions of outdated information.

Using (7.29), the first-order descent lemma (see Appendix B), Lipschitz continuity of the gradient, the inequality

$$|Z_{i(k)}| \|\underline{Z}_k\| \leq |Z_{i(k)}|^2 + \|\underline{Z}_k\|^2 \tag{7.42}$$

and Lemma 7.2 we obtain

$$J(\underline{\Theta}_{k+1}) = J(\underline{\Theta}_k - \rho_k \underline{Z}_k)$$

$$
\begin{aligned}
&\leq J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) Z_{i(k)} + (L/2)\rho_k^2 \sum_{i=1}^{N} |Z_{i(k)}|^2 \\
&= J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + \rho_k \sum_{i=1}^{N} \left(\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) - \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k) \right) Z_{i(k)} \\
&\quad + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \\
&\leq J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + \rho_k L \sum_{i=1}^{N} \|\underline{\Theta}_k - \underline{\Theta}_k^i\| |Z_{i(k)}| \\
&\quad + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \\
&\leq J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} \\
&\quad + \rho_k L \sum_{i=1}^{N} \left(\sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa \|\underline{Z}_\kappa\| \right) |Z_{i(k)}| + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \\
&\leq J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + L \sum_{i=1}^{N} \sum_{\kappa=\max\{1,k-B\}}^{k-1} (\rho_k^2 |Z_{i(k)}|^2 + \rho_\kappa^2 \|\underline{Z}_\kappa\|^2) \\
&\quad + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \\
&\leq J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + L \sum_{i=1}^{N} \left(B\rho_k^2 |Z_{i(k)}|^2 + \sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \right) \\
&\quad + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \\
&= J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + LB\rho_k^2 \|\underline{Z}_k\|^2 + LN \sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \\
&\quad + (L/2)\rho_k^2 \|\underline{Z}_k\|^2 \\
&= J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + L(B + \frac{1}{2})\rho_k^2 \|\underline{Z}_k\|^2 + \\
&\quad LN \sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \\
&\leq J(\underline{\Theta}_k) - \rho_k \sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} + LA \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \quad (a.s.) \qquad (7.43)
\end{aligned}
$$

where

$$
A = \max\{(B + \frac{1}{2}, N)\} \qquad (7.44)
$$

Taking conditional expectations on both sides of this inequality, conditioned on

$\mathcal{F}_k$, we obtain

$$E\{J(\underline{\Theta}_{k+1})|\mathcal{F}_k\} \leq J(\underline{\Theta}_k) - \rho_k E\left\{\sum_{i=1}^{N} \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) Z_{i(k)} \middle| \mathcal{F}_k\right\} + (LA)E\left\{\sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \middle| \mathcal{F}_k\right\} \quad (a.s.) \tag{7.45}$$

Using Lemma 7.1 we obtain

$$E\{J(\underline{\Theta}_{k+1})|\mathcal{F}_k\} \leq J(\underline{\Theta}_k) - \rho_k \left(K_1\alpha_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|^2 - K_2\beta_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|\right) + (LA)E\left\{\sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \middle| \mathcal{F}_k\right\} \tag{7.46}$$

$$= J(\underline{\Theta}_k) - K_1\rho_k\alpha_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|^2 + K_2\rho_k\beta_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right| + (LA)E\left\{\sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \|\underline{Z}_\kappa\|^2 \middle| \mathcal{F}_k\right\} \tag{7.47}$$

$$\leq J(\underline{\Theta}_k) + K_2\rho_k\beta_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right| + LA \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 E\{\|\underline{Z}_\kappa\|^2 |\mathcal{F}_k\} \quad (a.s.) \tag{7.48}$$

Define the quantities

$$X_k \triangleq J(\underline{\Theta}_k) \tag{7.49}$$

$$V_k \triangleq K_2\rho_k\beta_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right| + LA \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 E\{\|\underline{Z}_\kappa\|^2|\mathcal{F}_k\} \tag{7.50}$$

$$\mathcal{F}_k \triangleq \{Z_{i(\kappa)}|i \in \{1,\ldots,N\}, \kappa < k\} \tag{7.51}$$

Since the cost is nonnegative and bounded, and the process is Markov, and V_k is nonnegative, we are in position to invoke the following extension of the Supermartingale Convergence theorem, whose proof is provided in Appendix A, Section A.5, provided

that we can establish the fact that

$$\sum_{k=1}^{\infty} E\{V_k\} < \infty \tag{7.52}$$

Lemma 7.3 (Extended Supermartingale Convergence)
Let $\{X_k\}$ and $\{V_k\}$ be given sequences of random variables and for each $k \geq 1$ let X_k and V_k be measurable with respect to $\mathcal{F}_k$, where $\mathcal{F}_1 \subset \mathcal{F}_2 \cdots$ is a monotonically increasing sequence of Borel Fields. Suppose each of the following conditions holds with probability one and for all k.

1. $X_k \geq 0, E\{X_k\} < \infty$
2. $V_k \geq 0, \ E\{V_k\} < \infty, \ \sum_{k=1}^{\infty} E\{V_k\} < \infty.$
3. $E\{X_{k+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\} \leq X_k + V_k$

Then, there exists a nonnegative (finite) random variable X such that the sequence of random variables $\{X_k\}$ converges to X with probability one.

Using Lemma 7.1 we can write

$$\begin{aligned}
\sum_{k=1}^{\infty} E\{V_k\} &= \sum_{k=1}^{\infty} E\left\{ K_2 \rho_k \beta_k \sum_{\{i|k\in\mathcal{K}^i\}} \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right| \right. \\
&\quad \left. + LA \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 E\{\|\underline{Z}_\kappa\|^2|\mathcal{F}_k\} \right\} \\
&= K_2 \sum_{k=1}^{\infty} \rho_k \beta_k \sum_{\{i|k\in\mathcal{K}^i\}} E\left\{ \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right| \right\} \\
&\quad + LA \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 E\{\|\underline{Z}_\kappa\|^2\} \\
&\leq K_2 \sum_{k=1}^{\infty} \rho_k \beta_k \sum_{\{i|k\in\mathcal{K}^i\}} E\left\{ \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right| \right\} \\
&\quad + LA \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \left(K_3 \sum_{\{i|\kappa\in\mathcal{K}^i\}} E\left\{ \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_\kappa^i) \right|^2 \right\} + K_4 \nu_\kappa \right)
\end{aligned}$$

$$
\begin{aligned}
= \; & K_2 \sum_{k=1}^{\infty} \rho_k \beta_k \sum_{\{i|k\in\mathcal{K}^i\}} E\left\{\left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|\right\} \\
& + LAK_3 \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \sum_{\{i|\kappa\in\mathcal{K}^i\}} E\left\{\left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_\kappa^i)\right|^2\right\} \\
& + LAK_4 \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \nu_\kappa
\end{aligned} \tag{7.53}
$$

Recalling Assumption 6.1, there exists a constant K_5 such that

$$
\left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta})\right| \leq \|\nabla J(\underline{\Theta})\| \leq K_5 \quad \forall \underline{\Theta} \in \Re^N \tag{7.54}
$$

which implies that

$$
E_{\underline{\Theta}}\left\{\left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta})\right|\right\} \leq K_5 \tag{7.55}
$$

Then, we can bound (7.53)

$$
\begin{aligned}
\sum_{k=1}^{\infty} E\{V_k\} \; \leq \; & K_2 \sum_{k=1}^{\infty} \rho_k \beta_k N K_5 + LAK_3 \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 N K_5^2 \\
& + LAK_4 \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \nu_\kappa \\
\leq \; & K_2 N K_5 \sum_{k=1}^{\infty} \rho_k \beta_k + LAK_3 N K_5^2 \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \\
& + LAK_4 \sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} \rho_\kappa^2 \nu_\kappa
\end{aligned} \tag{7.56}
$$

Letting

$$
K_6 = K_2 N K_5, \quad K_7 = LAK_3 N K_5^2, \quad K_8 = LAK_4 \tag{7.57}
$$

and noting that we can interchange the sums

$$
\sum_{k=1}^{\infty} \sum_{\kappa=\max\{1,k-B\}}^{k} f(\kappa) = \sum_{\kappa=1}^{\infty} \sum_{k=\kappa}^{\kappa+B} f(\kappa) \tag{7.58}
$$

then it follows from Assumption 6.4 that

$$\begin{aligned}\sum_{k=1}^{\infty} E\{V_k\} &\leq K_6 \sum_{k=1}^{\infty} \rho_k \beta_k + K_7 \sum_{\kappa=1}^{\infty} \sum_{k=\kappa}^{\kappa+B} \rho_\kappa^2 + K_8 \sum_{\kappa=1}^{\infty} \sum_{k=\kappa}^{\kappa+B} \rho_\kappa^2 \nu_\kappa \\ &= K_6 \sum_{k=1}^{\infty} \rho_k \beta_k + K_7 B \sum_{\kappa=1}^{\infty} \rho_\kappa^2 + K_8 B \sum_{\kappa=1}^{\infty} \rho_\kappa^2 \nu_\kappa \\ &\leq \infty \end{aligned} \tag{7.59}$$

Therefore, the extended martingale convergence result of Lemma 7.3 does indeed apply, and the sequence of costs $\{J(\underline{\Theta}_k)\}$ converges with probability one; the first assertion of the proposition is proved.

Now reordering (7.47) and substituting in V_k from (7.50)

$$K_1 \rho_k \alpha_k \sum_{\{i|k \in \mathcal{K}^i\}} \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right|^2 \leq J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1}) | \mathcal{F}_k\} + V_k \tag{7.60}$$

Taking unconditional expected values on both sides of the inequality and then summing

$$K_1 \sum_{k=1}^{\infty} \rho_k \alpha_k \sum_{\{i|k \in \mathcal{K}^i\}} E\left\{ \left| \frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i) \right|^2 \right\} \leq \sum_{k=1}^{\infty} E\{J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1}) | \mathcal{F}_k\}\} + \sum_{k=1}^{\infty} E\{V_k\} \tag{7.61}$$

The sum

$$\sum_{k=1}^{\infty} E\{J(\underline{\Theta}_k) - E\{J(\underline{\Theta}_{k+1}) | \mathcal{F}_k\}\} \tag{7.62}$$

is bounded since by Proposition A.2

$$\begin{aligned}\sum_{j=1}^{k} E\{J(\underline{\Theta}_j) - E\{J(\underline{\Theta}_{j+1}) | \mathcal{F}_j\}\} &= \sum_{j=1}^{k} E\{J(\underline{\Theta}_j)\} - E\{E\{J(\underline{\Theta}_{j+1}) | \mathcal{F}_j\}\} \\ &= \sum_{j=1}^{k} E\{J(\underline{\Theta}_j)\} - E\{J(\underline{\Theta}_{j+1})\} \\ &= E\{J(\underline{\Theta}_1)\} - E\{J(\underline{\Theta}_{k+1})\}, \ \forall k \end{aligned} \tag{7.63}$$

which is bounded for all k due to the boundedness of J.

Thus,

$$\sum_{k=1}^{\infty} \rho_k \alpha_k \sum_{\{i|k\in\mathcal{K}^i\}} E\left\{\left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|^2\right\} < \infty \tag{7.64}$$

which implies by Lebesgue's Monotone Convergence Theorem (Proposition A.3) that

$$E\left\{\sum_{k=1}^{\infty} \rho_k \alpha_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|^2\right\} < \infty \tag{7.65}$$

From this it follows that

$$\sum_{k=1}^{\infty} \rho_k \alpha_k \sum_{\{i|k\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k^i)\right|^2 < \infty \ \ (a.s.) \tag{7.66}$$

Since we have assumed that

$$\sum_{k=1}^{\infty} \rho_k \alpha_k = \infty \tag{7.67}$$

it must be the case that

$$\liminf_{k\to\infty} \sum_{\kappa=k}^{k+B} \sum_{\{i|\kappa\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_\kappa^i)\right|^2 = 0 \ \ (a.s.) \tag{7.68}$$

To finish the proof, we need to relate this condition, expressed in terms of the local information $\underline{\Theta}_k^i$, to the summary vector $\underline{\Theta}_k$. To do this, note that in the course of proving that $\sum_{k=1}^{\infty} E\{V_k\} < \infty$ we effectively showed that

$$\sum_{k=1}^{\infty} \rho_k^2 E\{\|\underline{Z}_k\|^2\} < \infty \tag{7.69}$$

Similarly to the argument above, this implies by the Monotone Convergence Theorem that

$$E\left\{\sum_{k=1}^{\infty} \rho_k^2 \|\underline{Z}_k\|^2\right\} < \infty \tag{7.70}$$

which gives

$$\sum_{k=1}^{\infty} \rho_k^2 \|\underline{Z}_k\|^2 < \infty \ \ (a.s.) \tag{7.71}$$

from which we can finally conclude that

$$\rho_k \|\underline{Z}_k\| \to 0 \ \ (a.s.) \tag{7.72}$$

as $k \to \infty$. Recalling Lemma 7.2,

$$\|\underline{\Theta}_k - \underline{\Theta}_k^i\| \leq \sum_{\kappa=\max\{1,k-B\}}^{k-1} \rho_\kappa \|\underline{Z}_\kappa\| \tag{7.73}$$

it follows that

$$\|\underline{\Theta}_k - \underline{\Theta}_k^i\| \to 0 \ \ (a.s.) \tag{7.74}$$

as $k \to \infty$. It also follows from iteration (7.29)

$$\underline{\Theta}_{k+1} = \underline{\Theta}_k - \rho_k \underline{Z}_k \tag{7.75}$$

that

$$\|\underline{\Theta}_{k+1} - \underline{\Theta}_k\| \leq \rho_k \|\underline{Z}_k\| \tag{7.76}$$

so that

$$\|\underline{\Theta}_{k+1} - \underline{\Theta}_k\| \to 0 \ \ (a.s.) \tag{7.77}$$

as $k \to \infty$.

Together, (7.74) and (7.77) imply that the quantity

$$\max\{\|\underline{\Theta}_k - \underline{\Theta}_\kappa^i\| \,|k \leq \kappa \leq k+B\} \tag{7.78}$$

can be made arbitrarily small for any i by taking k sufficiently large. Then, the Lipschitz continuity of ∇J from Assumption 2.4 allows us to express (7.68) in terms of of the summary vector as

$$\liminf_{k\to\infty} \sum_{\kappa=k}^{k+B} \sum_{\{i|\kappa\in\mathcal{K}^i\}} \left|\frac{\partial J}{\partial \theta_i}(\underline{\Theta}_k)\right|^2 = 0 \ \ (a.s.) \tag{7.79}$$

From the definition of B, it is clear that for every i and k there exists some time

$\kappa \in [k, k+B]$ such that $\kappa \in \mathcal{K}^i$. Thus, in the sum (7.79), the summand $|\partial J(\underline{\Theta}_k)/\partial\theta_i|$ appears at least once for each i so that we can conclude

$$\liminf_{k\to\infty} \nabla J(\underline{\Theta}_k) = 0 \ \ (a.s.) \tag{7.80}$$

and the second assertion of the proposition is proved.

■

The comments of Chapter 6 regarding the meaning of this result still apply. It also still follows directly that, with an a posteriori assumption of boundedness of the parameter sequence, stronger results can be obtained. In particular, if the cost is unimodal with a single global minimum, the result of Corollary 6.1 is immediate. That the conclusions of Corollary 6.2 continue to hold as well is not altogether obvious, although we feel this to be true. Adaptation of Kushner's bounding argument to the asynchronous case appears to be quite tedious, so we omit further consideration of this issue.

7.4 Discussion

It is interesting that under substantially weaker assumptions on the timing of updates and communications, similar conclusions regarding asymptotic convergence were obtained as for the synchronous versions of these algorithms in Chapter 6. Again, it is important to note that this does *not* imply that the rates of convergence of the algorithms should be expected to be comparable. Indeed, the effects of asynchronism on rate of convergence are at present not well understood.

Our ability to draw similar conclusions as Chapter 6 resulted from the fact that the errors due to asynchronism could be lumped into an error term of order ρ_k^2, and could then be controlled by appropriate choice of the sequence ρ_k. The additional bias and (possibly unbounded) step variance characteristic of the GSD conditions for our algorithms did not adversely impact the results obtained previously by Tsitsiklis since these simply contributed additional terms which were similarly controlled. We were

also able to exploit the boundedness of the function and its derivatives to simplify the arguments for our problem.

7.5 Numerical Experiments

In this section, we describe how to implement simple numerical experiments which capture the types of asynchronism depicted in Figures 7-1 and 7-2. We then present results of a few experiments indicating the interesting issues which can be explored with these implementations.

For the results of the previous section to be applicable to our experiments, it is necessary that the asynchronism be partial asynchronism obeying Assumption 7.1. We enforce this assumption using the schemes described below. For the purposes of performing numerical experiments, we will be liberal in our assumptions regarding the accessibility of a global clock at each processor. This is simply for ease of programming the experiments. Local clocks are taken to be the number of updates performed at each processor, so that the sequences ρ_k^i and δ_k^i will depend on the number of updates of component i which have been performed.

WIN-Type Algorithms: For WIN-Type algorithms, we can meaningfully introduce asynchronism into both the updates and communications by allowing them to occur in random order at random times according to the following scheme.

For simplicity, we assume that every processor employs an estimation phase of constant duration N_E global time iterations, and that every estimation phase is immediately followed by communication of the new estimated operating point to all processors which require it. Thus, there is a one-to-one association of estimation and communication phases. Furthermore, we assume that estimation phases, when they occur, must immediately follow update phases, i.e no lags are allowed between an update and the corresponding estimation phase. This is the scenario depicted in the timing diagram of Figure 7-1. Let the variable k_{Ulast}^i denote the last global time index at which an update at DM i occurred, and let F_E^i and F_U^i be flags indicating that processor i is currently engaged in estimation, and that processor i has just performed

an update, respectively. Specifically, let

$$F^i_{E(k)} = \begin{cases} 1 & \text{if processor } i \text{ is estimating at time } k \\ 0 & \text{else} \end{cases} \tag{7.81}$$

and

$$F^i_{U(k)} = \begin{cases} 1 & \text{if processor } i \text{ updated at time } k-1 \\ 0 & \text{else} \end{cases} \tag{7.82}$$

For each processor i, also define the (coin-flipping) random variables U^i and E^i given by

$$U^i = \begin{cases} 1 & \text{w.p. } p^i_U \\ 0 & \text{else} \end{cases} \tag{7.83}$$

and

$$E^i = \begin{cases} 1 & \text{w.p. } p^i_E \\ 0 & \text{else} \end{cases} \tag{7.84}$$

where $p^i_U \in (0,1]$ and $p^i_E \in (0,1]$ are fixed probabilities of update at processor i, and of estimation/communication at processor i.

Assume that at every instant of global time, realizations of U^i_k and E^i_k are available to processor i, and that processor i updates and estimates/communicates according to the following rules, where $B > N_E$ is the asynchronism measure.

(i) (Update) At each global time index k, processor i performs an update of its parameter θ_i according to

$$\theta_{i(k+1)} = \begin{cases} \theta_{i(k)} - \rho^i_k Z_{i(k)}(\underline{\theta}^i_k) & \text{if } (U^i_k = 1 \text{ and } F^i_E = 0\ ) \text{ or } (k - k^i_{Ulast}) = B-1 \\ \theta_{i(k)} & \text{else} \end{cases} \tag{7.85}$$

In other words, at each global time index k, processor i updates with probability p^i_U, but updates are not permitted to interrupt estimation phases in progress. Furthermore, an update is forced to occur if one has not occurred over the last B global time units. Since $B > N_E$ and estimation phases must immediately follow update phases, the preemptive update will never interrupt an estimation phase.

(ii) (Estimate/Communicate) Processor i initiates an estimation/communication phase if $F^i_{U(k)} = 1$ and $E^i_k = 1$ or if $F^i_{U(k)} = 1$ and the update was forced. In other words, following an update, processor i initiates an estimation phase with probability p^i_E, unless the update was the forced update, in which case a corresponding estimation/communication phase is also forced.

This scheme results in the updates and estimation/communications occurring in random order at random times, but ensures that Assumption 7.1 is obeyed by forcing an update and communication by every processor i at least once every interval of global time of length B. Furthermore, the scheme has significant potential for experimental variety since there are several tunable parameters, including the asynchronism measure B, and the the updating and estimation/communication probabilities p^i_U and p^i_E of each processor i. By varying these probabilities, the relative rates of update and communication activity between the processors, the resulting impact on rate of convergence can be empirically studied. Note we have chosen not to incorporate communication delays for simplicity.

KW-Type Algorithms: For the KW-Type algorithms, only asynchronism in the updates may be meaningfully introduced, as depicted in Figure 7-2. There is also no complication involving estimation and communication. Thus, processor i updates according to

$$\theta_{i(k+1)} = \begin{cases} \theta_{i(k)} - \rho^i_k Z_{i(k)} & \text{if } U^i_k = 1 \text{ or } (k - k^i_{Ulast}) = B - 1 \\ \theta_{i(k)} & \text{else} \end{cases} \tag{7.86}$$

7.5.1 Example

In this section we provide a simple illustrative example to indicate the types of studies which could be performed using the above simple schemes. We consider the two-sided KW algorithm applied to 2-Tand, for the Gaussian detection problem

$$\mu_0 = 1, \quad \mu_1 = 3, \quad \sigma^2_A = \sigma^2_B = 1, \quad p_0 = 0.75 \tag{7.87}$$

For the very simple type of asynchronism applicable in the KW setting, there are 4 adjustable parameters: the asynchronism measure B, and the updating probabilities for each component p_U^{α}, $p_U^{\beta_0}$ and $p_U^{\beta_1}$. It is possible to explore the effects of varying these parameters empirically, by examining the resulting convergence rate.

For example, Figures 7-4 and 7-5 indicate the observed rates of convergence of the parameters and cost under varying amounts of asynchronism. In particular, the solid curves represent $B = 1$, or no asynchronism, while the dashed and dotted-dashed curves represent $B = 2$ and $B = 10$, respectively.

These curves indicate the intuitive result that as the asynchronism measure is increased, the convergence of the algorithm becomes slower and slower.

However, counterintuitive effects may also be observed. For example, consider the effect of varying the relative update rates between DM A and DM B. Intuitively, since the cost is more sensitive to the settings of DM B, because it makes the final team decision, it is reasonable to expect that better performance should be obtained by allowing DM B to update more frequently. This is in fact observed for the case $B = 2$ as shown in Figure 7-6, for which convergence obtained using $p_U^{\beta_0} = p_U^{\beta_1} = 1$ and $p_U^{\alpha} = 0.5$ is clearly superior to that obtained for the reversed choice $p_U^{\beta_0} = p_U^{\beta_1} = 0.5$ and $p_U^{\alpha} = 1$.

However, if the asynchronism measure is increased to $B = 10$, the opposite effect is observed, namely it appears to result in better convergence to allow DM A to update more frequently than DM B, although the curves are now closer together. Thus, the conclusion appears to be a function of the asynchronism measure B, as it is surely also a function of starting value.

The above experiments suggest that the training framework of this report can be applied to investigate many interesting issues concerning asynchronism, and its effects on adaptation in decentralized environments.

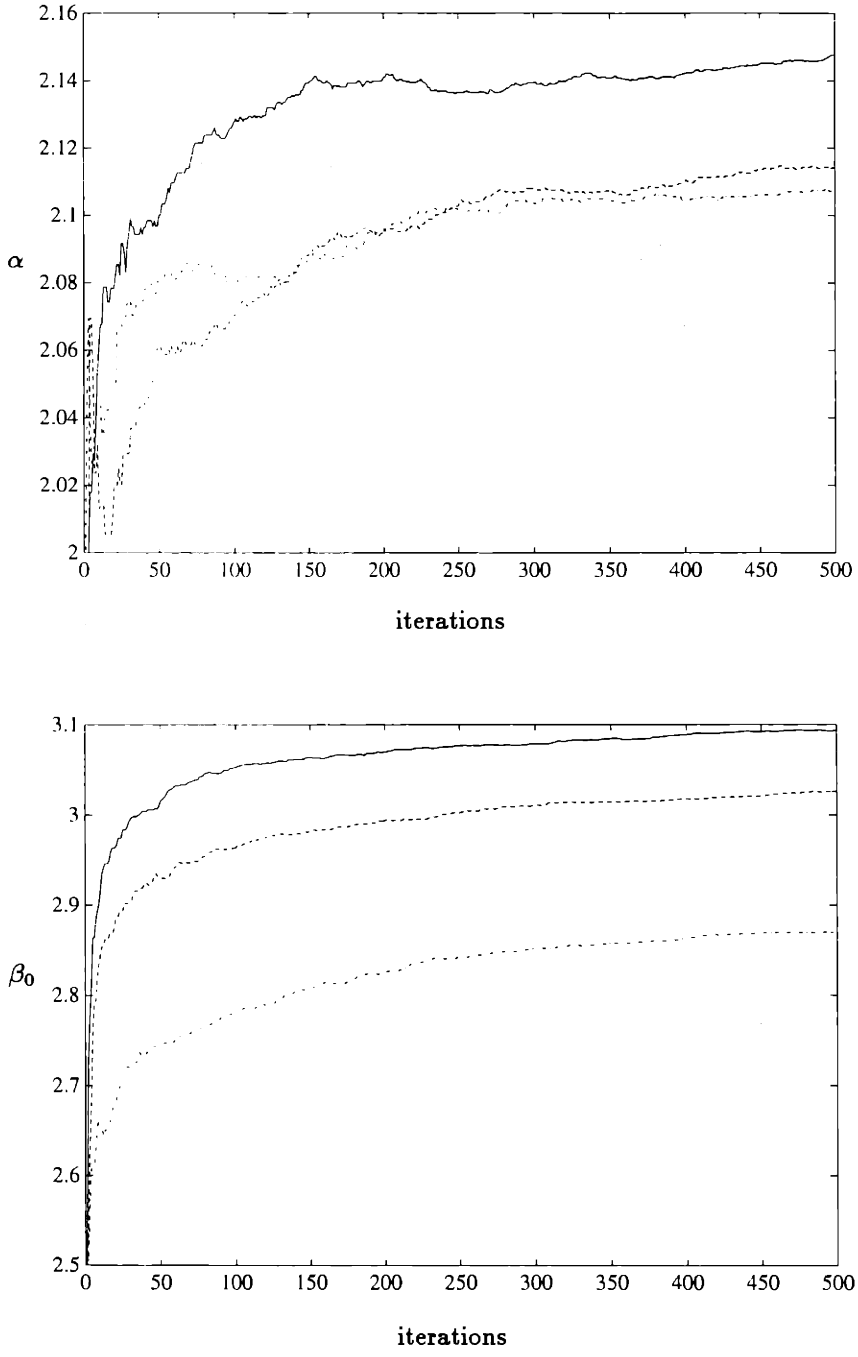
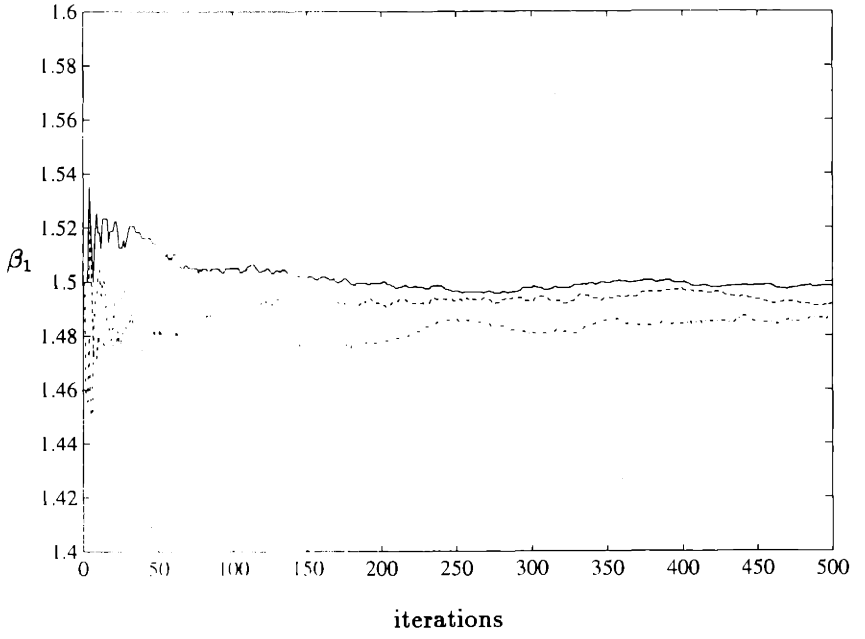


Figure 7-4: Effects of increasing asynchronism on the convergence of KW (two-sided) for the 2-Tand network. Case B=1 (solid), B=2 (dashed), and B=10 (dotted and dashed). Update probabilities are $p_U^{\alpha} = p_U^{\beta_0} = p_U^{\beta_1} = 0.5$. Sample paths are averages over 10 sample paths. Gaussian case, $\mu_0 = 1$, $\mu_1 = 3$, $\sigma_A^2 = \sigma_B^2 = 1$, $p_0 = 0.75$. Optimal values are $\alpha^* = 2.2209$, $\beta_0^* = 3.2521$.



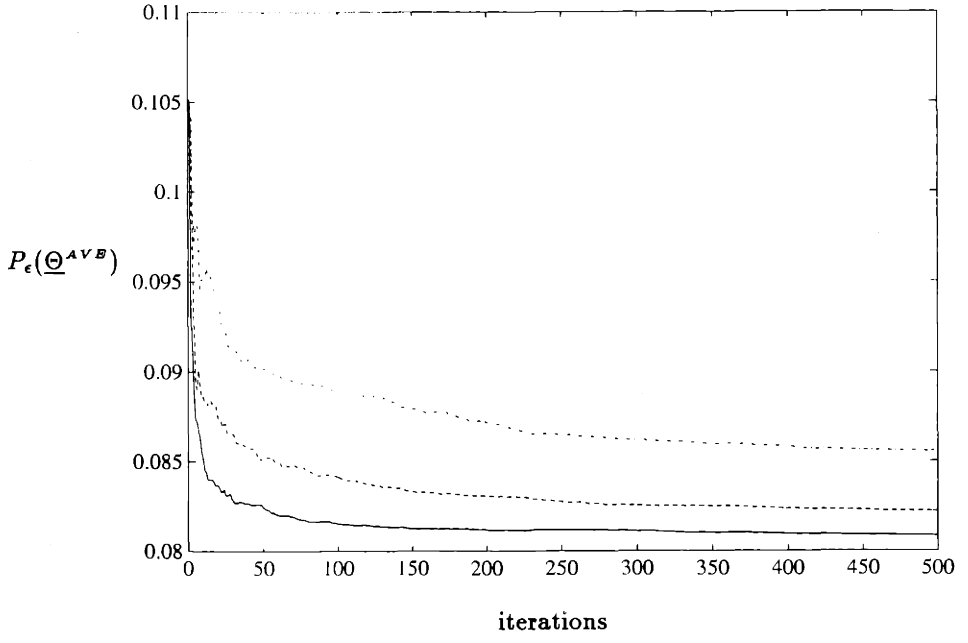


Figure 7-5: Effects of increasing asynchronism (cont'd). Case B=1 (solid), B=2 (dashed), and B=10 (dotted and dashed). Update probabilities are $p_U^{\alpha} = p_U^{\beta_0} = p_U^{\beta_1} = 0.5$. Optimal value is $\beta_1^* = 1.5734$. The lower curve is probability of error on the average sample path. Optimal value is $P_\epsilon(\underline{\Theta}^*) = 0.0794$

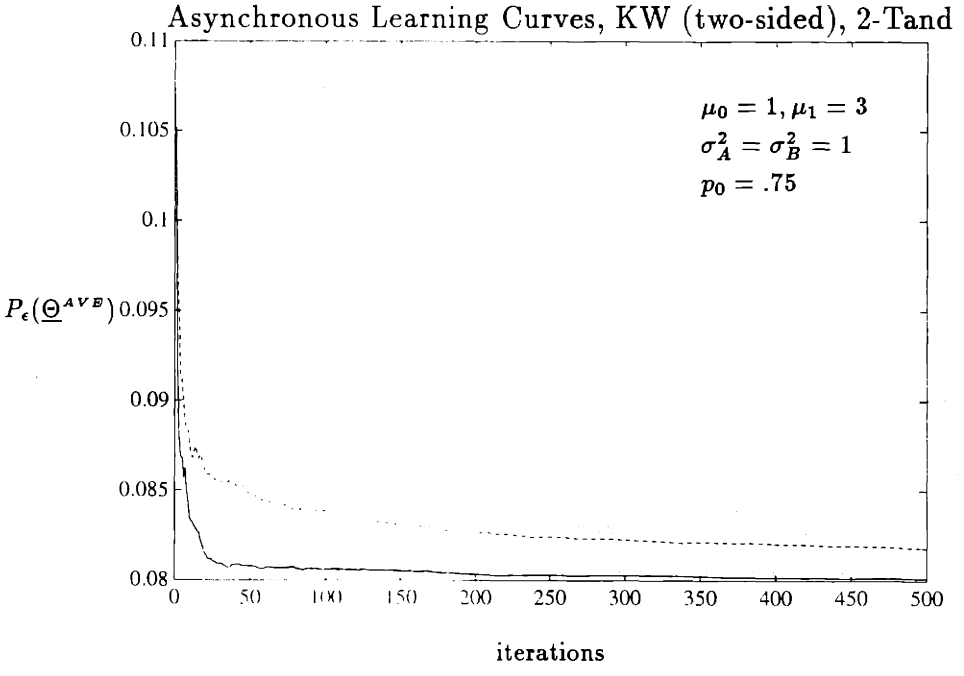


Figure 7-6: Sample Paths of $P_\epsilon(\underline{\Theta}_k^{AVE})$ for case $B = 2$. Averages computed over 20 sample paths. Case $p_U^\alpha = 0.5$, $p_U^{\beta_0} = p_U^{\beta_1} = 1$ (solid curve). Case $p_U^\alpha = 1.0$, $p_U^{\beta_0} = p_U^{\beta_1} = 0.5$ (dashed). Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$.

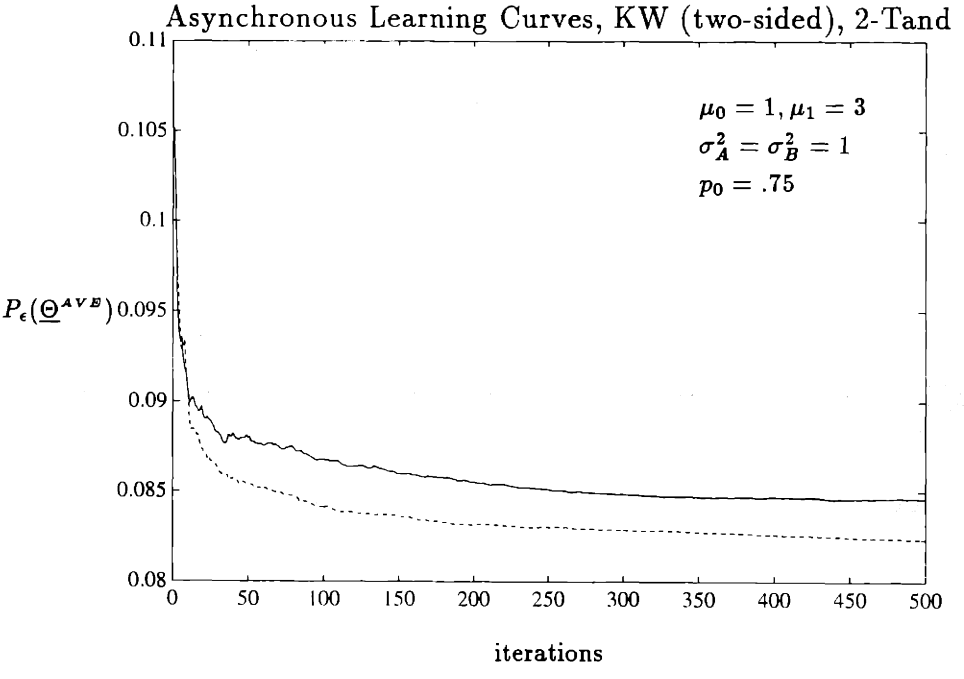


Figure 7-7: Sample Paths of $P_\epsilon(\underline{\Theta}_k^{AVE})$ for case $B = 10$. Averages computed over 20 sample paths. Case $p_U^\alpha = 0.5$, $p_U^{\beta_0} = p_U^{\beta_1} = 1$ (solid curve). Case $p_U^\alpha = 1.0$, $p_U^{\beta_0} = p_U^{\beta_1} = 0.5$ (dashed). Optimal value $P_\epsilon(\underline{\Theta}^*) = 0.0794$.

7.6 Chapter Conclusions

There are two central conclusions of this chapter: the first is that the same results obtained by Tsitsiklis [66], [67], [6] continue to hold under more general (GSD) conditions on the steps. This result extends[3] the class of stochastic gradient algorithms known to tolerate partial asynchronism to include KW and window type algorithms, both of which possess decaying bias in the gradient measurement, and for which the step variance must be allowed to become unbounded. The second conclusion is that the restrictive timing constraints imposed on the algorithms in Chapter 6 appear to have been superfluous from the point of view of asymptotic convergence, since similar results were obtained with very little structure imposed on the relative timing of updates and and communications between the processors. In particular, the asymptotic convergence of asynchronous variants of WIN and KW, as well as various timing variations of WIN-GS and KW-GS, has been established under the assumption that fixed[4] upper bounds on the infrequency of updates and the amount by which local information is allowed to become outdated exist.

The meaning of these results, in the context of our distributed nonparametric training algorithms, is that the threshold parameters may be adjusted in random order, at random times, and in the case of window algorithms, that the local models can be allowed to become out of date, so that the DMs can make adjustments based on outdated versions of the current operating points of the other members of the team, and so long as these effects are bounded, the convergence of the thresholds can still be assured[5]. Clearly, from the point of view of developing a paradigm for organizational training, the ability of the models to successfully exhibit the effects of asynchronism makes them much more interesting and realistic.

[3]For those functions obeying our stronger assumptions.

[4]It has been shown by Tsitsiklis [66], [67] that, for decreasing stepsize algorithms, B may be allowed to grow as the stepsize decreases. We have chosen not to address this issue for simplicity.

[5]We algorithmically formalized these notions in a model of distributed training.

Chapter 8

Final Remarks

In this final chapter, we summarize our results and conclusions, relate them to the goals we put forth in Chapter 1, and suggest some directions for future research.

8.1 Summary of Results

In this report, we investigated distributed nonparametric training algorithms for optimizing the decision rules in a particular class of team decision problem. Specifically, we considered the problem of applying distributed stochastic approximation (stochastic gradient) algorithms to the problem of nonparametric optimization of the thresholds in Bayesian DBHT networks. Our goal in this endeavor was to develop mathematical models suitable for studying issues of adaptation and learning in decentralized environments.

Our investigation into this problem involved several steps. We first restricted ourselves to a particular class of binary hypothesis testing network, tree-structured networks with conditionally independent observations at each node. The restriction was critical, not only for obtaining a reasonable parameterization of the decision rules, but also for demonstrating many of the properties we later required to prove convergence of the WIN training algorithms. The optimality conditions for the team decision rules came in the form of coupled systems of nonlinear equations specifying the necessary conditions for optimality. These were expressed as person-by-person

optimality conditions, specifying each DM's optimal decision rule given the decision rules of the other DMs were held fixed. The person-by-person optimal decision rules for four small examples of team hypothesis testing networks were presented in detail in order to illustrate the form of the decision rules arising in the major topological variations of tree-structured networks, and also to indicate the complexity inherent in the underlying optimization.

We settled on a parameterization of the decision rules which was simple and well-suited to nonparametric optimization, and which included the optimal rules for a problem class of interest, the decentralized Gaussian detection problem. The choice of linear threshold rules was natural for this application. We then established that the Bayes cost was sufficiently well-behaved under this parameterization to admit optimization by gradient-based techniques. In particular, we established conditions on the underlying conditional densities of the hypothesis test so that the resulting Bayes cost was twice differentiable and possessed a Lipschitz continuous gradient. The required conditions on the densities amounted to continuity, boundedness, differentiability, and boundedness of the derivative. We were unable to analytically demonstrate unimodality of the team cost, although for the team Gaussian detection problem we believe the property may hold. In the absence of such a guarantee, the most we could hope to show is convergence of the threshold parameters to a stationary point (person-by-person optimal solution). The problem of global unimodality of the team cost remains an open problem. Thus, determination of the optimal decision rules was shown to require solution of a difficult optimization problem, possessing possibly nonunique stationary points.

We then presented several alternative points of view on the optimization of the network thresholds. The sequential sample space enumeration proved extremely useful for generating parameterizations of the team error probability and its derivatives, and for establishing certain structural properties of the team cost. The optimal control formulation was also subsequently useful, as it formed the basis of a nonparametric back-propagation training algorithm.

The next step was to consider the nonparametric training solution to the sin-

gle DM (one-dimensional) problem. It was immediately discovered that while the Bayes cost does possess favorable differentiability and smoothness properties under the linear threshold parameterization, it unfortunately has a derivative which does not appear to be a regression function. No stochastic realization of the derivative was available, so Robbins-Monro type algorithms could not be used. To circumvent this difficulty, so-called window algorithms were used, which effectively generate a sequence of regression functions which converge to the true derivative, and which can be used in a modified RM type algorithm. In contrast, application of Kiefer-Wolfowitz type algorithms in our setting was straightforward and presented no difficulty. Numerical experiments with these techniques indicated that they performed extremely well.

Construction of distributed synchronous network (team) training algorithms focused on methods for ensuring that sufficient information was made available to each DM so that it could compute estimates of the partial derivative of the cost with respect to its parameter(s). The class of WIN algorithms accomplished this by requiring that each DM maintain a local representation of the rest of the network, in the form of coupling costs; this allowed local feedback to be used to train the DM. In effect, the coupling costs appropriately bias the DM's local subproblem so that the rest of the network is considered when threshold adjustments based on local information are computed. The local representations must be continually updated as the network evolves; this is accomplished by having each DM communicate estimates of its current operating point to those DMs that require it to update. In contrast, the network KW algorithms allow for the local construction of partial derivative information by requiring that the network output be observable by each DM, so that effects of perturbations in its parameter are directly measurable. The KW-type training algorithms require no local representations, and hence no communication, in order to update the thresholds.

Numerical experiments with both classes of techniques suggested that they behave reasonably, evidencing clearly an average reduction of the cost. On the negative side, most of the algorithms appeared to require many network measurements, although

no effort was made to minimize the number of required measurements. Several algorithms, in particular the ad hoc WIN-GS and KW-GS algorithms appeared to perform extremely well.

Convergence analysis of all of the training algorithms was performed using results on martingale convergence. Our strategy was to identify a common structural property of all of the algorithms, establish sufficient conditions for the convergence of algorithms with this structure, and then verify on a case-by-case basis that all of the training algorithms were indeed in this class. We were able to show that the sequence of cost realizations for each of the algorithms converges with probability one. Under an additional a posteriori assumption of boundedness of the parameter sequence, we demonstrated that all limit points of the parameter sequence are stationary (a.s.), and under unimodality of the cost we obtained (a.s.) convergence to the globally optimal solution. We could thus conclude that our synchronous gradient-based training algorithms behaved as well as could be hoped for.

We then investigated the possibility of weakening the previous stringent timing constraints on the algorithms. In particular, we investigated whether previously known results on the convergence of stochastic gradient methods under bounded asynchronism could be extended to handle our more general descent condition. We first introduced a formal algorithmic model of distributed training which was capable of representing all of our algorithms. We then demonstrated that the extension was possible; the implication is that convergence of the training algorithms continues to hold under much weaker assumptions on the relative timing and frequency of updates and communications than were previously imposed.

The majority of the effort in this report was devoted to laying technical groundwork. In this regard, the following were accomplished: a mathematically precise formulation of the problem was given, the assumptions necessary to make progress on the problem were indicated, and the resulting training algorithms were shown to be reasonable, both through analysis and simulation. Demonstration of these points provided theoretical justification of the suggested mathematical framework. The central technical conclusion of this report is that it is possible to train to the minimum

probability of error observation thresholds, subject to certain difficulties caused by the (possible) unimodality of the minimum probability of error cost function. However, in our numerical experience, unimodality in Gaussian detection problems has never been observed.

8.2 Discussion

In this section, we briefly address several aspects of the work in this report, with an eye toward relating the final conclusions to the goals originally put forth in Chapter 1 and to the questions raised in Section 1.3.1. We also indicate additional questions that may be addressed using the methods we have presented. We intend in this discussion to point out issues which can be further explored through numerical experiments, and to highlight the flexibility of the models by indicating the many different component parameters which may be varied. Then, in the following section, we address some topics for future research.

In returning to the questions originally posed in Chapter 1, we see that we are now in position to answer several of them. One such question concerns the information which must be obtained locally at each DM so that the partial derivative with respect to each parameter may be inferred in the distributed setting. We found that our algorithms could be separated into two classes based on this point, with the WIN-type algorithms requiring the maintenance of local models, and the KW-type algorithms employing direct sampling of the team cost. The key information required by a processor in the distributed setting was captured by the notion of an endogenous measurement, defined in Chapter 7.

With respect to the quality of endogenous measurement required to perform local parameter updates, it was found that the local models of the WIN algorithms could be based on operating point estimates of widely varying quality, so that the impact of deterioration in local model quality on the empirical rate of convergence, particularly as it relates to placement in the topology, or to node expertise, could be explored.

Although we have not specifically explored the idea in this report, the samples of

the team cost for KW-type methods may be obtained more accurately by averaging over several decision cycles. This reduces the variance of the sampled estimate of the cost and improves the finite difference approximation of the partial derivative. Thus averaging can be used to give some DMs more accurate feedback than others, allowing the possibility of exploring the relationship between a DM's location in the topology, its expertise, and the quality of the feedback it should receive. For example, it is expected that less observable parameters may be updated on the basis of noisier feedback without dramatically affecting the observed rate of convergence. Indeed, for both WIN and KW algorithms, we expect that better information should be supplied to those DMs controlling parameters to which the cost is maximally sensitive. Which parameters these are is, of course, a function of the location of the current iterate, as well as the network topology, and parameters external to the team such as prior probabilities and conditional densities.

With respect to "who must communicate with whom, what, when, etc." during training, a question which can be addressed in the context of WIN algorithms, it is possible to say that

- communication is, in principle, required between all network DMs, although the optimal control formulation indicates that this communication can be localized between neighboring nodes only by suitably structuring the computation
- it is sufficient to communicate estimates of the current operating points
- communication may occur at random times, and be infrequent with respect to updates, so long as the length of the intervals between communications is bounded by a constant

A different set of issues may be explored through experimentation with the choice of stepsize parameters, which we indicated in Chapters 5 and 6 could be chosen differently for each parameter. It is reasonable to assume that different choices of stepsize at the various DMs might be able to improve the rate of convergence. For example, it may hold generally true that network-wide convergence benefits from

providing nodes with greater expertise larger stepsize gain, although this remains to be explored.

Another source of variety in the models concerns the effect of asynchronism, in particular the asynchronism measure and the update and estimation/communication probabilities. As suggested in Chapter 7, a large number of interesting experiments may be performed by exploring the relationship between these quantities and their effect on the rate of convergence.

In summary, it appears that the models of this report are capable of capturing an enormous variety of behaviors, and can be used to address many interesting and complex issues.

8.3 Future Research

In this section we briefly comment on some directions for future research which come under the headings "more analysis", "more numerics", "improvements in the algorithms", and "extensions".

First and foremost is "more analysis". Significant attention has already been devoted to the fact that rate of convergence analysis and asymptotic normality studies have the potential to analytically address many of the issues discussed in the previous section. Although such analysis appears difficult, providing it would be of substantial value, both for improving the behavior of the algorithms as well as shedding insight on how various problem components, such as prior bias, expertise, and topology, affect the observed convergence of the algorithms. In addition, some outstanding theoretical issues are worthy of additional attention. In particular, the related issues of unimodality of the team cost for Gaussian detection problems, and the block contraction property of the system of necessary conditions, merit more study. In addition, extending the argument of Kushner regarding (a.s.) convergence to a stationary point to the asynchronous case merits additional study as well.

With respect to numerical experiments, we barely scratched the surface in this report. Many possible additional numerical experiments were suggested in the pre-

vious section. Through extensive experimentation, it should be possible to arrive at empirical insights concerning many of the issues of the previous section.

With respect to improving the algorithms, there are a variety of possibilities. In this report we considered only the most basic of gradient algorithms. However, well-known from deterministic nonlinear programming are the potential improvements from employing scaling matrices, and incorporating higher derivative information such as in second-order methods, which exploit Hessian information in order to speed convergence. Furthermore, we made no attempt to optimize the choice of stepsize in this report, either the time dependence or gain parameters. Indeed, from the regions of allowable stepsize exponents we simply made a choice which appeared to work well, then chose gain parameters heuristically. It would be well worth establishing optimal choices of these parameters, so that the best achievable performance of the algorithms might be identified.

Many extensions of this work are also possible, most of which require further developments in the theory, of varying degrees of difficulty. Handling dependent observations, or feedforward/feedback structures, appears to be quite difficult, although it is possible that progress can be made on problems with specific structure. A promising possibility is to employ a constant rather than decreasing stepsize to provide solutions which can adapt in response to slow time variation in the underlying hypothesis test. Extension to vector observations, M hypotheses, and multiple messages appears possible, although the value of these extensions, at least from a modeling point of view, is not readily apparent. Other possible extensions involve the measurement sequence, for example the use of finite data sets, and the study of optimal measurement sequences.

In this report we have applied ideas of distributed computation to the problem of nonparametric threshold optimization in DBHT networks; the consequence of this effort is that learning dynamics have been added to what was already an interesting and descriptive static DBHT modeling framework. It is our hope that these models will provide a useful paradigm for the study of adaptation in uncertain distributed environments, such as those characteristic of human decision making organizations,

sensor networks, and even biological neural networks.

Appendix A

Essential Probability Theory

In this appendix, we review some fundamental results from probability theory. More can be found in Papoulis [45], Drake [16], [1], and Billingsley [9] and Doob [15].

A.1 Estimating Probabilities

A problem that is fundamental to the development of minimum error training algorithms is the estimation of probabilities over repeated independent trials. This problem arises, for example, in the estimation of error rate or conditional probabilities such as probability of false alarm, detection, and so on.

Suppose we have designed a DM and wish to estimate its error rate empirically using a set of training data. That is, we wish to estimate the DM's true performance by observing its performance over a set of independent and identically distributed observation data for which the correct class is known a priori. Then we must concern ourselves with the fact that a point on the $P_\epsilon(\cdot)$ criterion function can be exactly sampled only as asymptotically many data are acquired. It is of course impractical to obtain the exact value, and so it is estimated using finite data. The most straightforward manner of obtaining the estimate is to tabulate the empirical relative frequency of errors over a "sufficiently large" number of trials. We must first make mathematical sense of this statement.

The following analysis from elementary probability theory (cf. [16],[45]) allows

one to bound the required amount of data to know the error rate within a specified degree of accuracy. Consider the following problem.

Problem A.1 *An event occurs with fixed but unknown probability p on each trial. We estimate the true value of p with the empirical relative frequency of occurrence of the event over a fixed number of independent trials N. How good is the estimate?*

Analysis: We begin by defining the random variable

$$Q_k = \begin{cases} 1 & \text{if event occurs on } k\text{th trial} \\ 0 & \text{else} \end{cases} \tag{A.1}$$

which is the indicator of an event on the kth trial. The Q_k are IID random variables, each taking the value 1 with probability p. That is, the event occurs with probability p on each trial and the Q_k are independent Bernoulli random variables. Then

$$E\{Q_k\} = p \ \forall k \tag{A.2}$$

$$E\{Q_k^2\} = p \ \forall k \tag{A.3}$$

$$E\{(Q_k - E\{Q_k\})^2\} = p(1-p) \ \forall k \tag{A.4}$$

Define the random variable

$$S_N = \sum_{k=1}^{N} Q_k \tag{A.5}$$

which represents the number of occurrences of the event over N trials, where N is given and fixed. Then we wish to take as our estimate of p the quantity

$$\hat{P}_N = \frac{S_N}{N} \tag{A.6}$$

which is the sample mean of the set $\{Q_k;\ k = 1, \ldots, N\}$, i.e., the empirical relative frequency of events which occurred.

It is clear that

$$\begin{aligned} \Pr(S_N = k) &= \Pr(k \text{ events in } N \text{ trials}) \\ &= \left(\begin{array}{c} N \\ k \end{array} \right) p^k (1-p)^{N-k} \end{aligned} \tag{A.7}$$

so that S_N is a Binomially distributed random variable. Using (A.2)-(A.4) we can compute

$$E\{\hat{P}_N\} = p \tag{A.8}$$

$$E\{\hat{P}_N^2\} = \frac{1}{N} p(1-p) + p^2 \tag{A.9}$$

$$E\{(\hat{P}_N - E\{\hat{P}_N\})^2\} = \frac{1}{N} p(1-p) \tag{A.10}$$

Note the variance $\sigma^2_{\hat{P}}$ depends on the true value of p being estimated, and attains a maximum value of $1/4$ for the values $p = 1/2$, $N = 1$, so that

$$\sigma^2_{\hat{P}_N} \leq \frac{1}{4}, \quad \forall p, N \tag{A.11}$$

and $\hat{P}_N$ is an unbiased estimator of p with bounded variance. In fact, the empirical relative frequency $\hat{P}_N$ can be shown to be the maximum likelihood estimate of p. Furthermore, it is a consistent and efficient estimator.

If N is a random variable as well, then

$$E\{\hat{P}_N\} = E_N\{E\{\hat{P}_N|N\}\} = E_N\{p\} = p \tag{A.12}$$

$$E\{\hat{P}_N^2\} = E_N\left\{\frac{1}{N} p(1-p) + p^2\right\} = E_N\left\{\frac{1}{N}\right\} p(1-p) + p^2 \tag{A.13}$$

$$E\{(\hat{P}_N - E\{\hat{P}_N\})^2\} = E_N\left\{\frac{1}{N} p(1-p)\right\} = E_N\left\{\frac{1}{N}\right\} p(1-p) \tag{A.14}$$

A.1.1 Convergence

The convergence of the estimate may be characterized in several ways. The Chebyshev inequality may be applied to demonstrate that the sample mean converges to the true mean "in probability", or "weakly" (see Section A.2). This result is the so-called Bernoulli Law of Large Numbers, or Weak Law of Large numbers. It states that

$$\lim_{k\to\infty} \Pr\left(\left|\frac{S_k}{k} - p\right| \geq \epsilon\right) = 0 \tag{A.15}$$

It should be noted from this expression that there is no value of k for which we can be sure that our estimated value $\hat{P}_N$ is within a given ϵ of p. However, the weak law states that it is increasingly unlikely, as k becomes large, for the sample mean to deviate by more than ϵ from the true mean.

The estimate may also be shown to converge in a stronger sense (again see Section A.2), known as "with probability one", or "strong" convergence, using the machinery of measure theory [1], [9]. This result states that

$$\lim_{k\to\infty} \Pr(\sup_{m\geq k} \left|\tfrac{S_k}{k} - p\right| \geq \epsilon) = 0 \tag{A.16}$$

and is also referred to as the Strong Law of Large Numbers. It is often equivalently expressed as

$$\frac{S_k}{k} \underset{k\to\infty}{\longrightarrow} kp \quad \text{w.p.1} \tag{A.17}$$

This is a stronger statement because it says something about every sample path of S_k/k, namely that as k becomes large it is increasingly unlikely that *any element* of the sequence deviates by more than ϵ from the true mean.

A.1.2 Accuracy

We now discuss two methods for characterizing the accuracy of the estimate for finite N.

Confidence Intervals

It was shown above that the random variable $S_N = \hat{P}_N N$ is binomially distributed. Using this distribution, it is possible for each value of $p \in (0,1)$ and each fixed value of N to compute a smallest interval $\mathbf{I}$ such that

$$\sum_{\hat{P}_N \in \mathbf{I}} \begin{pmatrix} N \\ \hat{P}_N N \end{pmatrix} p^{\hat{P}_N N}(1-p)^{N(1-\hat{P}_N)} \geq 0.95 \tag{A.18}$$

The interval $\mathbf{I}$ contains the estimate $\hat{P}_N$ with probability 0.95 for a fixed value of p and N. This is referred to as the *95% confidence interval* of p when the estimate $\hat{P}_N$ is computed over N trials. Figure A.1.2 plots the endpoints of the intervals $\mathbf{I}$ for various fixed values of N [57]. For a given value of $\hat{P}_N$ and N the probability is at least .95 that the true value of p lies in the interval between the upper and lower curves corresponding to that value of N. Notice that since the variance of the estimate is reduced as p approaches 0 or 1, and achieves a maximum value for $p = 0.5$, that the curves are hot-dog shape; they are fattest in the middle and squeezed at the end. It is also clear that as N is increased, the intervals rapidly shrink in size.

Example: Suppose that the estimate $\hat{P}_N = 0.4$ has been obtained over $N = 100$ trials for the error rate of our DM. Then the probability that the true value of p lies somewhere in the interval (0.3,0.5) is 0.95.

In the same way, for a given value of N we can define a *tube of confidence* over the entire error surface as shown in Figure A.1.2. The width of this tube varies with the true but unknown probability of error.

Chebyshev

An alternative approach characterizes the accuracy based on the Bernoulli or Weak Law of Large Numbers, and is a direct consequence of the Chebyshev inequality.

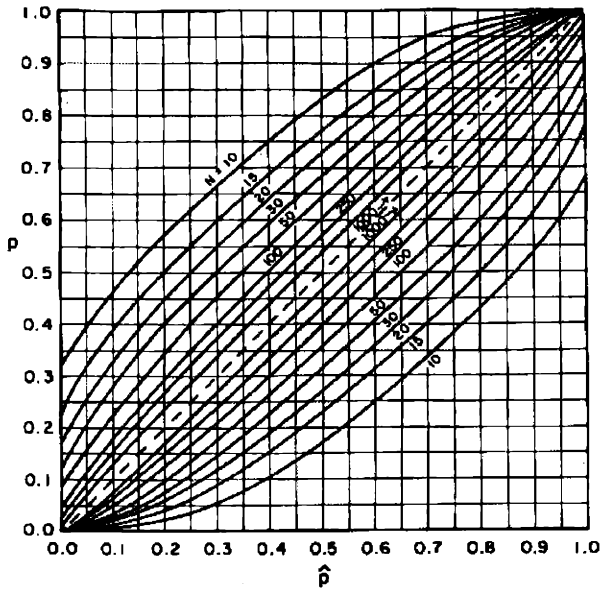


Figure A-1: 95% Confidence Intervals (Copyright 1962, American Telephone and Telegraph Co.)

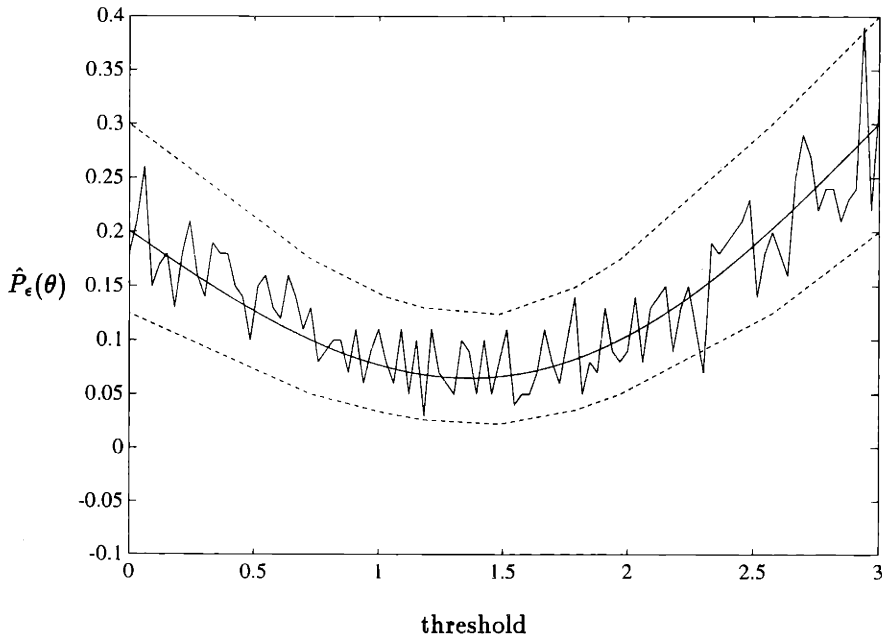


Figure A-2: 95% Confidence Tube. Solid lines, true mean and estimated surfaces. Dotted line, confidence tube. Estimated cost 100 samples per point, 9 points of confidence envelope plotted. Gaussian case, $\mu_0 = 0$, $\mu_1 = 3$, $\sigma^2 = 1$, $p_0 = 0.4$.

Proposition A.1 *If we wish to bound the probability that $\hat{P}_N$ differs from p by more than ϵ by the value C, i.e.,*

$$Pr(|\hat{P}_N - p| \geq \epsilon) \leq C \tag{A.19}$$

then an upper bound on the required number of trials is given by N such that

$$\frac{1}{4N\epsilon^2} = C \tag{A.20}$$

Proof: The proof is a straightforward application of the Chebyshev inequality. We may upper bound the variance of the estimator by 1/4 as above and directly apply the Chebyshev inequality to obtain

$$\Pr(|p - \hat{P}_N| \geq \epsilon) \leq \frac{p(1-p)}{N\epsilon^2} \leq \frac{1}{4N\epsilon^2} \tag{A.21}$$

■

This approach results in a more conservative bound on the amount of data needed to achieve a given accuracy because it neglects any information about the distribution of the estimate.

For example, suppose we wish to know how many samples it will take to ensure that the probability that $\hat{P}_N$ differs from the true value p by more than 0.1 is less than .05. Then the proposition gives the bound

$$\frac{1}{4N(.1)^2} = .05 \Rightarrow N = 500 \tag{A.22}$$

A more conservative bound was obtained by this technique, which is to be expected since the Chebyshev inequality depends only on the mean and variance of the distribution. The bound in the example is only improved to 480 from 500 if the true variance corresponding to $p = 0.4$ is used.

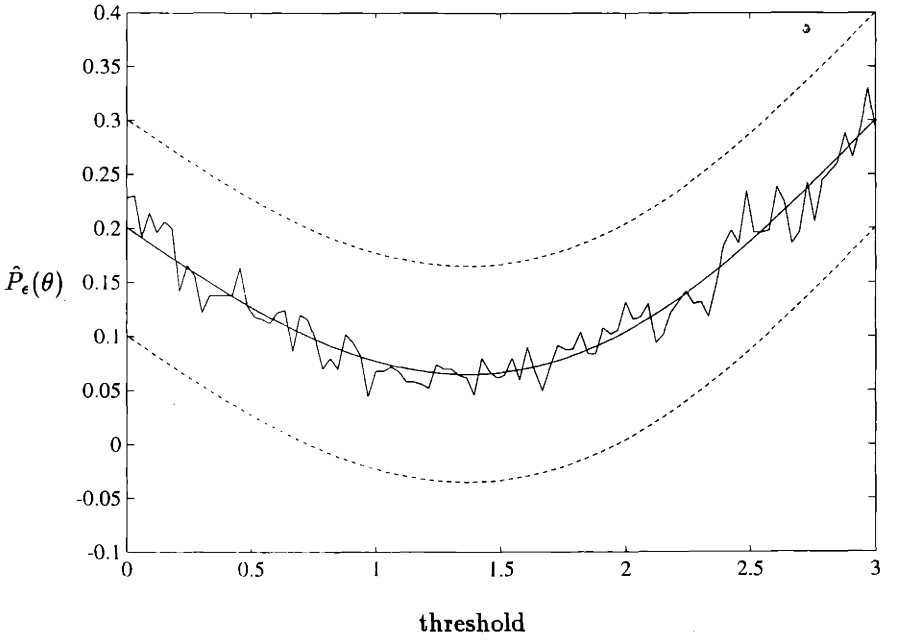


Figure A-3: 95% Probability Tube. Solid lines, true mean and estimated surfaces. Dotted line, probability tube. Estimated cost 500 samples per point; Gaussian case, $\mu_0 = 0$, $\mu_1 = 3$, $\sigma^2 = 1$, $p_0 = 0.4$.

This proposition implies that if we sample the P_ϵ surface using a fixed number of data N, then we can again specify a uniform diameter "tube" of width 2ϵ within which we know the true surface lies with probability $\left(1 - \frac{1}{4N\epsilon^2}\right)$. This is illustrated for a one dimensional example in Fig. A.1.2.

A.2 Stochastic Convergence

In this section we summarize the various notions of convergence of a sequence of random variables $\{\Theta_k\}$ to a random variable Θ^*.

Definition A.1 (Stochastic Convergence)

The sequence $\{\Theta_k\}$ *is said to converge to* Θ^*

(a) *in probability, in measure, weakly, or stochastically, if for every* $\epsilon > 0$

$$\lim_{k\to\infty} Pr(|\Theta_k - \Theta^*| \geq \epsilon) = 0 \tag{A.23}$$

or equivalently if

$$\lim_{k\to\infty} Pr(|\Theta_k - \Theta^*| \leq \epsilon) = 1 \tag{A.24}$$

(b) *in mean square, or quadratic mean, if*

$$\lim_{k\to\infty} E\{(\Theta_k - \Theta^*)^2\} = 0 \tag{A.25}$$

(c) *with probability one (w.p.1), strongly, almost surely (a.s.), or almost certainly (a.c.), if for any* $\epsilon > 0$

$$\lim_{k\to\infty} Pr(\sup_{m\geq k} |\Theta_k - \Theta^*| \leq \epsilon) = 1 \tag{A.26}$$

or equivalently

$$\lim_{k\to\infty} Pr(\sup_{m\geq k} |\Theta_k - \Theta^*| \geq \epsilon) = 0 \tag{A.27}$$

This statement is typically more concisely expressed as

$$Pr(\lim_{k\to\infty} \Theta_k = \Theta^*) = 1 \tag{A.28}$$

A sequence of random vectors $\{\underline{\Theta}_k\}$ is said to converge to a random vector $\underline{\Theta}^*$ in (a),(b), or (c) above, if *each component* Θ_i, $i = 1, \ldots, N$ converges to some random variable Θ_i^* in (a), (b), or (c), respectively.

It may be established that

$$\text{(b)} \Rightarrow \text{(a)}, \quad \text{(c)} \Rightarrow \text{(a)} \tag{A.29}$$

so that (b) and (c) are both stronger notions of stochastic convergence than (a), although it is not possible to relate (b) and (c).

A.3 Some Properties of Conditional Expectation

We present some properties of conditional expectation in the form of a proposition. Proof of these may be found in Billingsley [9].

Proposition A.2 (Properties of Conditional Expectation)
Let X, Y and $X_1, \ldots, X_M$ and $\{Y_i; i = 1, 2, \ldots\}$ be random variables with finite expectations, and define the sets $\mathcal{F} = \{X_1, \ldots, X_M\}$, $\mathcal{G} = \{X_1, \ldots, X_N\}$. Then, there holds

(a)

$$E\{E\{X|\mathcal{F}\}\} = E\{X\} \tag{A.30}$$

(b) *If $N \leq M$ (equivalently, $\mathcal{G} \subset \mathcal{F}$), then*

$$E\{E\{X|\mathcal{F}\}|\mathcal{G}\} = E\{X|\mathcal{G}\} \tag{A.31}$$

and

$$E\{E\{X|\mathcal{G}\}|\mathcal{F}\} = E\{X|\mathcal{G}\} \tag{A.32}$$

(c) *If $E\{|X|\} < \infty$, then $E\{X|F\}$ is well-defined and finite, w.p.1.*

(d) *If each $Y_i \geq 0$ and $\sum_{i=1}^{\infty} E\{Y_i\} < \infty$, then*

$$E\left\{\sum_{i=1}^{\infty} Y_i \middle| \mathcal{F}\right\} = \sum_{i=1}^{\infty} E\{Y_i|\mathcal{F}\} \tag{A.33}$$

where the infinite sums are interpreted as limits with probability one. Furthermore, both sides of equation (A.33) are finite, with probability one.

A.4 Integral Convergence

We will find the following well-known convergence theorem useful.

Proposition A.3 (Lebesgue's Monotone Convergence Theorem)
Let $\{X_i\}$ *be a sequence of nonnegative random variables, and suppose that* $\sum_{i=1}^{\infty} E\{X_i\} < \infty$. *Let* $S_k = \sum_{i=1}^{k} X_i$. *Then*

(a) *The sequence* $\{S_k\}$ *converges, w.p.1, to a finite-valued random variable* S.

(b) *There holds* $E\{S\} = \sum_{i=1}^{\infty} E\{X_i\}$

A.5 Martingale Convergence

Definition A.2 (Martingales)
Let $\{X_k;\ k = 1, 2, \ldots\}$ *be a sequence of random variables defined on a probability space* $\{\Omega, \mathcal{F}, P\}$. *and let* $\{\mathcal{F}_k\}$ *be a sequence of* σ*-fields in* $\mathcal{F}$. *If* $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, X_k *is measurable* $\mathcal{F}_k$, *and* $E\{|X_k|\} < \infty$, $k = 1, 2, \ldots$, *then the sequence* $\{(X_k, \mathcal{F}_k)\}$ *is called a*

(a) *martingale if*

$$E\{X_{k+1}|\mathcal{F}_k\} = X_k \ \ (a.s.) \tag{A.34}$$

(b) *submartingale if*

$$E\{X_{k+1}|\mathcal{F}_k\} \geq X_k \ \ (a.s.) \tag{A.35}$$

(c) *supermartingale if*

$$E\{X_{k+1}|\mathcal{F}_k\} \leq X_k \ \ (a.s.) \tag{A.36}$$

For the special choice of σ-fields $\mathcal{G}_k = \sigma(X_1, \ldots, X_k)$, for which it obviously holds that $\mathcal{G}_k \subset \mathcal{G}_{k+1}$, these inequalities are expressed

(a) martingale if

$$E\{X_{k+1}|X_1, \ldots, X_k\} = X_k \ \ (a.s.) \tag{A.37}$$

(b) submartingale if

$$E\{X_{k+1}|X_1, \ldots, X_k\} \geq X_k \ \ (a.s.) \tag{A.38}$$

(c) supermartingale if

$$E\{X_{k+1}|X_1,\ldots,X_k\} \leq X_k \ \ (a.s.) \tag{A.39}$$

We will be concerned in particular with supermartingales. A supermartingale is the stochastic generalization of a monotonically decreasing sequence since the definition implies that

$$E\{X_{k+1}\} \leq E\{X_k\} \tag{A.40}$$

The analog of the statement that a monotonically decreasing nonnegative sequence which is bounded below has a limit is a classical result from Doob [15].

Proposition A.4 (Supermartingale Convergence Theorem I)
Let $\{X_k\}$ *be a supermartingale such that* $0 \leq X_k$, $\forall k$. *Then there exists a nonnegative random variable* $0 \leq X^*$ *such that*

$$X_k \underset{k\to\infty}{\longrightarrow} X^* \ \ (a.s.) \tag{A.41}$$

Another version, also due to Doob [15], handles martingales which are not guaranteed to be nonnegative.

Proposition A.5 (Supermartingale Convergence Theorem II)

Let $\{X_k\}$ *be a supermartingale such that*

$$\sup_k E\{|X_k|\} = K < \infty \tag{A.42}$$

Then there exists a random variable X^* *such that*

$$X_k \underset{k\to\infty}{\longrightarrow} X^* \ (a.s.) \tag{A.43}$$

where X^* *satisfies*

$$E\{|X^*|\} \leq K \tag{A.44}$$

There are many extensions of the Supermartingale Convergence Theorem, two of which we require in this report. First we provide the proof of the result of Macqueen [38] employed in Chapter 6. The key conditions are the boundedness of the X_k's and the fact that the sum of the V_k's is finite.

Proposition A.6 (Extended Supermartingale Convergence: MacQueen) *Let $\{X_k\}$ and $\{V_k\}$ be given sequences of random variables and for each $k \geq 1$ let X_k and V_k be measurable with respect to $\mathcal{F}_k$, where $\mathcal{F}_1 \subset \mathcal{F}_2 \cdots$ is a monotonically increasing sequence of Borel Fields. Suppose each of the following conditions holds with probability one and for all k.*

1. $|X_k| \leq D < \infty$ *for some* $D > 0$.

2. $V_k \geq 0$ *and* $\sum_{k=1}^{\infty} V_k < \infty$.

3. $E\{X_{k+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\} \leq X_k + V_k$

Then, the sequences of random variables $\{X_k\}$ and $\{R_k\}$, where $R_0 \stackrel{\Delta}{=} 0$ and

$$R_k \stackrel{\Delta}{=} \sum_{j=1}^{k} (X_j - E\{X_{j+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_j\}), \quad k \geq 1 \tag{A.45}$$

both converge with probability one.

Proof. Let $Y_k = X_k + R_{k-1}$, so that $\{Y_k\}$ is a martingale sequence. Let $C > 0$ be a positive number, and consider the sequence $\{\tilde{Y}_k\}$ obtained by stopping Y_k[1] at the first value of k for which $Y_k \leq -C$. From condition (3.) we see that

$$Y_k \geq -\sum_{i=1}^{k-1} V_i - D \tag{A.46}$$

and since

$$Y_k - Y_{k-1} \geq 2D \tag{A.47}$$

we have

$$\tilde{Y}_k \geq \max\left\{-\sum_{i=1}^{k-1} V_i - D, -(C + 2D)\right\} \tag{A.48}$$

The sequence $\{\tilde{Y}_k\}$ is a martingale, so that $E\{\tilde{Y}_k\} = E\{\tilde{Y}_1\}, k = 1, 2, \ldots$ and being

[1] See Doob, pg. 300

bounded from below with $E\{|\tilde{Y}_1|\} \leq D$, it certainly holds that

$$\sup_k E\{|\tilde{Y}_k|\} < \infty \tag{A.49}$$

The Martingale Theorem[2] shows that $\{\tilde{Y}_k\}$ converges (a.s.). But $Y_k = \tilde{Y}_k$ on the set A_C where

$$-\sum_{k=1}^{\infty} V_k > -C - D \tag{A.50}$$

and condition (2.) implies that $\Pr(A_C) \to 1$ as $C \to \infty$. Thus, $\{Y_k\}$ converges (a.s.). This means that

$$R_{k-1} = Y_{k+1} - X_{k+1} \tag{A.51}$$

is (a.s.) bounded. Using condition (3.) we can write

$$-R_k = \sum_{i=1}^{k} V_i - \sum_{i=1}^{k} \Delta_i, \quad \Delta_i \geq 0 \tag{A.52}$$

But, since R_k and $\sum_{i=1}^{k} V_i$ are (a.s.) bounded, the sum $\sum_{i=1}^{\infty} \Delta_i$ converges (a.s.), R_k converges (a.s.), and finally, X_k converges (a.s.).

■

The second extended martingale result is from Tsitsiklis [6]. This version requires that the sum of the *expected values* of the V_k's be finite, but does not require that the X_k's be bounded.

[2]Doob, pg. 319

Proposition A.7 (Extended Supermartingale Convergence)

Let $\{X_k\}$ and $\{V_k\}$ be given sequences of random variables and for each $k \geq 1$ let X_k and V_k be measurable with respect to $\mathcal{F}_k$, where $\mathcal{F}_1 \subset \mathcal{F}_2 \cdots$ is a monotonically increasing sequence of Borel Fields. Suppose each of the following conditions holds with probability one and for all k.

1. $X_k \geq 0, E\{X_k\} < \infty$
2. $V_k \geq 0, \ E\{V_k\} < \infty, \ \sum_{k=1}^{\infty} E\{V_k\} < \infty.$
3. $E\{X_{k+1}|\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\} \leq X_k + V_k$

Then, there exists a nonnegative (finite) random variable X such that the sequence of random variables $\{X_k\}$ converges to X with probability one.

Proof.

(a) Define the random variable

$$S_k = E\left\{ \sum_{i=k}^{\infty} V_i \middle| \mathcal{F}_k \right\} \tag{A.53}$$

Then, Proposition A.2(d) states that S_k is finite, w.p.1, for every k. We define random variables W_k by

$$\begin{aligned} W_k &= X_k + S_k \\ &= X_k + E\left\{ \sum_{i=k}^{\infty} V_i \middle| \mathcal{F}_k \right\} \end{aligned} \tag{A.54}$$

Then, from Proposition A.2(b) and condition (3.), it holds that

$$\begin{aligned} E\{W_{k+1}|\mathcal{F}_k\} &= E\{X_{k+1}|\mathcal{F}_k\} + E\{S_{k+1}|\mathcal{F}_k\} \\ &= E\{X_{k+1}|\mathcal{F}_k\} + E\left\{ E\left\{ \sum_{i=k+1}^{\infty} V_i \middle| \mathcal{F}_{k+1} \right\} \middle| \mathcal{F}_k \right\} \end{aligned}$$

$$\begin{aligned} &\leq X_k + V_k + E\left\{\sum_{i=k+1}^{\infty} V_i \middle| \mathcal{F}_k\right\} \\ &= X_k + E\left\{\sum_{i=k}^{\infty} V_i \middle| \mathcal{F}_k\right\} \\ &= W_k \ (a.s.) \end{aligned} \tag{A.55}$$

Notice that each W_k is nonnegative, is measurable with respect to $\mathcal{F}_k$, and has finite expectation. Then, Proposition A.4 applies, and shows that the sequence $\{W_k\}$ converges, with probability one, to a nonnegative random variable W.

We will now prove that S_k converges to zero, with probability one. This will imply that $X_k = W_k - S_k$ converges to W, with probability one, and this will complete the proof. Using Proposition A.2(b), we have

$$\begin{aligned} E\{S_{k+1}|\mathcal{F}_k\} &= E\left\{E\left\{\sum_{i=k+1}^{\infty} V_i \middle| \mathcal{F}_{k+1}\right\} \middle| \mathcal{F}_k\right\} \\ &= E\left\{\sum_{i=k+1}^{\infty} V_i \middle| \mathcal{F}_k\right\} \\ &= S_k - E\{V_k|\mathcal{F}_k\} \\ &\leq S_k \ (a.s.) \end{aligned} \tag{A.56}$$

where the last inequality follows from the nonnegativity of V_k. We apply Proposition A.4 to the sequence $\{S_k\}$ to see that it converges to a nonnegative random variable S, with probability one. We now use the Monotone Convergence Theorem to obtain

$$E\{S_k\} = E\left\{\sum_{i=k}^{\infty} V_i\right\} = \sum_{i=k}^{\infty} E\{V_i\} \tag{A.57}$$

which converges to zero as k tends to infinity because of condition (2.). Suppose that S is nonzero with positive probability. Then, there exist some $\delta > 0$ and $\epsilon > 0$ such that $\Pr(S \geq \delta) \geq \epsilon$. On the other hand, S_k converges to S in probability and this implies that we can find some k_0 such that $\Pr(|S_k - S| \geq \delta/2) \leq \epsilon/2$ for every $k \geq k_0$.

Therefore, for $k \geq k_0$

$$\begin{aligned} \Pr(S_k \geq \delta/2) &\geq \Pr(S \geq \delta \text{ and } |S_k - S| \leq \delta/2) \\ &\geq \Pr(S \geq \delta) - \Pr(|S_k - S| \geq \delta/2) \\ &\geq \epsilon/2 \end{aligned} \tag{A.58}$$

This implies that $E\{S_k\} \geq (\delta\epsilon)/4$ for every $k \geq k_0$, which contradicts the convergence of the sequence $\{E\{S_k\}\}$ to zero, and shows that $S = 0$.

■

Appendix B

Essential Vector Calculus and Linear Algebra

In this appendix, we collect together several key results from vector calculus and linear algebra required in this report. The material is adapted from the appendices in [6], and [51], and proofs of the propositions may be found there. Additional useful material from analysis may be found in [54] and [63].

Let $J(\underline{\theta}) : \Re^N \mapsto \Re$ be a scalar-valued function of the N dimensional argument $\underline{\theta} = [\theta_1, \ldots, \theta_N]^T$. We define the *gradient* of J as the column vector

$$\nabla J(\underline{\theta}) = \left[\frac{\partial J}{\partial \theta_1}(\underline{\theta}), \frac{\partial J}{\partial \theta_2}(\underline{\theta}), \ldots, \frac{\partial J}{\partial \theta_N}(\underline{\theta})\right]^T \tag{B.1}$$

and the *Hessian* of J as the symmetric $N \times N$ matrix

$$\nabla^2 J(\underline{\theta}) = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1^2}(\underline{\theta}) & \frac{\partial^2 J}{\partial \theta_1 \theta_2}(\underline{\theta}) & \cdots & \frac{\partial^2 J}{\partial \theta_1 \theta_N}(\underline{\theta}) \\ \frac{\partial^2 J}{\partial \theta_2 \theta_1}(\underline{\theta}) & \frac{\partial^2 J}{\partial \theta_2^2}(\underline{\theta}) & \cdots & \frac{\partial^2 J}{\partial \theta_2 \theta_N}(\underline{\theta}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_N \theta_1}(\underline{\theta}) & \frac{\partial^2 J}{\partial \theta_N \theta_2}(\underline{\theta}) & \cdots & \frac{\partial^2 J}{\partial \theta_N^2}(\underline{\theta}) \end{bmatrix} \tag{B.2}$$

In the following, the notation $\|\cdot\|$ denotes the Euclidean norm.

Proposition B.1 (Mean Value Theorem)

If $J(\theta) : \Re \mapsto \Re$ is continuously differentiable, then for every pair $\theta_1, \theta_2 \in \Re$, there exists some $\theta_3 \in [\theta_1, \theta_2]$ such that

$$J(\theta_1) - J(\theta_2) = \frac{dJ}{d\theta}(\theta_3)(\theta_1 - \theta_2) \tag{B.3}$$

Corollary B.1 (Mean Value Inequality)

For J as in Proposition B.1, and under the assumption of boundedness of the first derivative of J, it holds that

$$|J(\theta_1) - J(\theta_2)| \leq L(\theta_1 - \theta_2) \tag{B.4}$$

for all $\theta_1, \theta_2 \in \Re$, where

$$L = \sup_{\theta} \frac{dJ}{d\theta}(\theta) \tag{B.5}$$

is a bounded nonnegative constant. In other words, J is Lipschitz continuous.

Proposition B.2 (First-Order Descent Lemma)

If $J(\underline{\theta}) : \Re^N \mapsto \Re$ is continuously differentiable, and its gradient satisfies the Lipschitz continuity assumption $\|\nabla J(\underline{\theta}_1) - \nabla J(\underline{\theta}_2)\| \leq L\|\underline{\theta}_1 - \underline{\theta}_2\|$ for every $\underline{\theta}_1, \underline{\theta}_2 \in \Re^N$, then

$$J(\underline{\theta}_1 + \underline{\theta}_2) \leq J(\underline{\theta}_1) + \underline{\theta}_2^T \nabla J(\underline{\theta}_1) + (L/2)\|\underline{\theta}_2\|^2 \tag{B.6}$$

or equivalently

$$|J(\underline{\theta}_1 + \underline{\theta}_2) - J(\underline{\theta}_1) - \underline{\theta}_2^T \nabla J(\underline{\theta}_1)| \leq (L/2)\|\underline{\theta}_2\|^2 \tag{B.7}$$

Proposition B.3 (Second-Order Descent Lemma [51])

If $J(\underline{\theta}) : \Re^N \mapsto \Re$ is continuously differentiable, and its Hessian satisfies the Lipschitz

continuity assumption $\|\nabla^2 J(\underline{\theta}_1) - \nabla^2 J(\underline{\theta}_2)\| \leq L\|\underline{\theta}_1 - \underline{\theta}_2\|$ *for every* $\underline{\theta}_1, \underline{\theta}_2 \in \Re^N$, *then*

$$J(\underline{\theta}_1 + \underline{\theta}_2) \leq J(\underline{\theta}_1) + \underline{\theta}_2^T \nabla J(\underline{\theta}_1) + (1/2)\underline{\theta}_2^T \nabla^2 J(\underline{\theta}_1)\underline{\theta}_2 + (L/6)\|\underline{\theta}_2\|^3 \tag{B.8}$$

or equivalently

$$|J(\underline{\theta}_1 + \underline{\theta}_2) - J(\underline{\theta}_1) - \underline{\theta}_2^T \nabla J(\underline{\theta}_1) - (1/2)\underline{\theta}_2^T \nabla^2 J(\underline{\theta}_1)\underline{\theta}_2| \leq (L/6)\|\underline{\theta}_2\|^3 \tag{B.9}$$

where

$$\|A\| = \max_{\{\underline{\theta} \in \Re^N \mid \|\underline{\theta}\|=1\}} \|A\underline{\theta}\| \tag{B.10}$$

is the maximum singular value of the matrix A.

Proposition B.4 (Schwartz Inequality)

If $\underline{\theta}_1, \underline{\theta}_2 \in \Re^N$, *then it holds that*

$$|\underline{\theta}_1^T \underline{\theta}_2| \leq \|\underline{\theta}_1\| \|\underline{\theta}_2\| \tag{B.11}$$

Bibliography

[1] Adams, M. and V. Guillemin. *Measure Theory and Probability.* Wadsworth and Brooks/Cole Advanced Books and Software, 1986.

[2] Barnard, E. and D. Casasent. A Comparison between Criterion Functions for Linear Classifiers, with an Application to Neural Nets. *IEEE Systems, Man and Cybernetics*, 19(5):1030–1041, 1989.

[3] Baudet, G.M. Asynchronous Iterative Methods for Multiprocessors. *Journal of the Assoc. for Computing Machinery*, 25(2):226–244, 1978.

[4] Benveniste, A. and M. Metivier, P. Priouret. *Adaptive Algorithms and Stochastic Approximations.* Springer-Verlag, 1990.

[5] Bertsekas, D.P. *Nonlinear Programming: Course Notes 6.252.* MIT, Cambridge MA., 1992.

[6] Bertsekas, D.P. and J.N. Tsitsiklis. *Parallel and Distributed Computation.* Prentice Hall, 1989.

[7] Bertsekas, D.P. and J.N. Tsitsiklis. Some Aspects of Parallel and Distributed Iterative Algorithms - A Survey. *Automatica*, 27(1):3–21, 1991.

[8] Bertsekas, D.P. and J.N. Tsitsiklis, M. Athans. Convergence Theories of Distributed Iterative Processes: A Survey. *MIT LIDS Report:LIDS-P-1412*, 1984.

[9] Billingsley, P. *Probability and Measure, Second Edition.* Wiley and Sons, 1986.

[10] Blum, J.A. Approximation Methods which Converge with Probability One. *Ann. Math. Stat.*, 25(2):382–386, 1954.

[11] Blum, J.A. Multidimensional Stochastic Approximation Methods. *Ann. Math. Stat.*, 25(4):737–744, 1954.

[12] Boettcher, K.L. and R.R. Tenney. Distributed Decisionmaking with Constrained Decisionmakers: A Case Study. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-16(6):813–822, 1986.

[13] Dembo, A. and T. Kailath. Model-Free Distributed Learning. *IEEE Trans. on Neural Networks*, 1(1):58–70, 1990.

[14] Do-Tu, H. and M. Installe. Learning Algorithms for Nonparametric Solution to the Minimum Error Classification Problem. *IEEE Trans. on Computers*, C-27(7):648–659, 1978.

[15] Doob, J.L. *Stochastic Processes.* John Wiley and Sons, Inc., 1953.

[16] Drake, A.W. *Fundamentals of Applied Probability Theory.* McGraw-Hill, 1967.

[17] Duda, R.O. and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[18] Dvoretzky, A. On Stochastic Approximation. In *Proc. 3rd Berkeley Symp. Math. Stat. Prob.*, volume 1, pages 39–55, 1956.

[19] Ekchian, L.K. *Optimal Design of Distributed Detection Networks.* PhD thesis, M.I.T., 1982.

[20] Fritz, J. and L. Gyorfi. On the Minimization of Classification Error Probability in Statistical Pattern Recognition. *Problems of Control and Information Theory*, pages 371–382, 1976.

[21] Fu, K.S. *Sequential Methods in Pattern Recognition and Machine Learning.* Academic Press, 1968.

[22] Fu, K.S. Learning System Theory. In Zadeh, L.A. and E. Polak, editor, *System Theory*, pages 425–463. McGraw-Hill, 1969.

[23] Gallager, R.G. *Discrete Stochastic Processes: Course Notes 6.262.* MIT, Cambridge MA., 1989.

[24] Green, D.M. and J.A. Swets. *Signal Detection Theory and Psychophysics.* Wiley, 1966.

[25] Hertz, J. and A. Krogh, R. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley Co., 1991.

[26] Ho, Y.C. and M.P. Kastner, E. Wong. Teams, Signaling, and Information Theory. *IEEE Trans. on Automatic Control*, pages 305–311, April 1978.

[27] Irving, W.W. Problems in Decentralized Detection. Master's thesis, M.I.T., 1991.

[28] Kac, M. A Note on Learning Signal Detection. *IRE Trans. on Information Theory*, pages 126–128, Feb. 1962.

[29] Kashyap, R.L. and C.C. Blaydon, K.S. Fu. Stochastic Approximation. In Mendel, J.M. and K.S. Fu, editor, *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, pages 329–355. Academic Press, 1970.

[30] Kiefer, J. and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23:462–466, 1952.

[31] Kushner, H.J. Stochastic Approximation Algorithms for the Local Optimization of Functions with Nonunique Stationary Points. *IEEE Trans. on Automatic Control*, AC-17:646–654, Oct. 1972.

[32] Kushner, H.J. and D.S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, 1978.

[33] Kushner, H.J. and G. Yin. Asymptotic Properties of Distributed and Communicating Stochastic Approximation Algorithms. *SIAM J. Control and Optimization*, 25:1266–1290, Sept. 1987.

[34] Kushner, H.J. and G. Yin. Stochastic Approximation Algorithms for Parallel and Distributed Processing. *Stochastics*, 22:219–250, 1987.

[35] Lippmann,R.P. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, pages 4–22, April 1987.

[36] Ljung, L. Analysis of Recursive Stochastic Algorithms. *IEEE Trans. on Automatic Control*, pages 551–575, August 1977.

[37] Luenberger, D.G. *Linear and Nonlinear Programming.* Addison-Wesley, 1984.

[38] MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations; On the Asymptotic Behavior of K-means. In *Proc. of the 5th Berkeley Symposium on Math. Prob. and Stats., Berkeley , CA*, 1967.

[39] Nedeljkovic', V. A Novel Multilayer Neural Networks Training Algorithm that Minimizes the Probability of Classification Error. *IEEE Trans. on Neural Networks*, pages 650–659, July 1993.

[40] Nevel'son, M.B. and R.Z. Has'minskii. *Stochastic Approximation and Recursive Approximation.* American Mathematical Society, 1976.

[41] Nilsson, N.J. *Learning Machines:Foundations of trainable pattern-classification systems.* McGraw-Hill, 1965.

[42] Ortega, J.M. and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations of Several Variables.* Academic Press, New York, 1970.

[43] Papastavrou, J. *Decentralized Decision Making in a Hypothesis Testing Environment.* PhD thesis, M.I.T., 1990.

[44] Papastavrou, J. and M. Athans. On Optimal Distributed Decision Architectures in a Hypothesis Testing Environment. *IEEE Trans. on Automatic Control*, AC-37(8):1154–1169, 1992.

[45] Papoulis, A. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1984.

[46] Pete, A. and K.R. Pattipati, D.L. Kleinman. Optimal Team and Individual Decision Rules in Uncertain Dichotomous Choice Situations. *Public Choice*, 75:205–230, 1993.

[47] Pete, A. and K.R. Pattipati, D.L. Kleinman. Team Relative Operating Characteristic: A Normative-Descriptive Model of Team Decisionmaking. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-23(6):1626–1648, 1993.

[48] Pete, A. D.L. Kleinman, K.R. Pattipati. Tasks and Organizations: A Signal Detection Model of Organizational Decisionmaking. *International Journal of Intelligent Systems in Accounting, Finance, and Management*, 2:289–303, 1993.

[49] Polyak, B. T. Convergence and Convergence Rate of Iterative Stochastic Algorithms I. General Case. *Automation and Remote Control*, 37:1858–1868, 1976.

[50] Polyak, B. T. and Y. Z. Tsypkin. Pseudogradient Adaptation and Training Algorithms. *Automation and Remote Control*, pages 377–397, 1972.

[51] Polyak, B.T. *Introduction to Optimization.* Optimization Software, Inc., 1987.

[52] Pothiawala, J. Analysis of a Two-Sensor Tandem Distributed Detection Network. Master's thesis, M.I.T., 1989.

[53] Robbins, H. and S. Monro. A Stochastic Approximation Method. *Ann. Mathematical Statistics*, 22:400–407, 1951.

[54] Rudin, W. *Principles of Mathematical Analysis.* McGraw-Hill, 1976.

[55] Rumelhart, D.E. and G.E. Hinton, R.J. Williams. Learning internal representations by error propagation. In Rumelhart, D.E. and J.L. McClelland, editor, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362. MIT press, Cambridge,MA, 1986.

[56] Sakrison, D.J. Stochastic Approximation: A Recursive Method for Solving Regression Problems. In Balakrishnan, A.V., editor, *Advances in Communication Systems, vol.2*, pages 51-106. New York: Academic Press, 1966.

[57] Sklansky, J. and G. Wassel. *Pattern Classifiers and Trainable Machines.* Springer-Verlag, 1981.

[58] Spall, J.C. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Trans. on Automatic Control*, AC-37(3):332–341, 1992.

[59] Tang, Z.B. *Optimization of Detection Networks.* PhD thesis, Univ. of Connecticut, 1990.

[60] Tang, Z.B. and K.R. Pattipati, D.L. Kleinman. An Algorithm for Determining the Decision Thresholds in a Distributed Detection Problem. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-21(1):231–237, 1991.

[61] Tang, Z.B. and K.R. Pattipati, D.L. Kleinman. Optimization of Detection Networks: Part I-Tandem Structures. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-21(5):1044–1059, 1991.

[62] Tang, Z.B. and K.R. Pattipati, D.L. Kleinman. Optimization of Detection Networks: Part II- Generalized Tree Structures. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-23:211–221, 1993.

[63] Taylor, A.E. and W.R. Mann. *Advanced Calculus.* John Wiley and Sons, 1983.

[64] Tenney, R.R. and N.R. Sandell, Jr. Detection with Distributed Sensors. *IEEE Trans. on Aerospace and Electronic Systems*, AES-17(4):501–510, 1981.

[65] Tsitsiklis, J.N. Decentralized Detection (to appear). In Poor, H.V. and J.B. Thomas, editor, *Advances in Statistical Signal Processing, vol.2: Signal Detection.*

[66] Tsitsiklis, J.N. *Problems in Decentralized Decision Making and Computation.* PhD thesis, M.I.T., 1984.

[67] Tsitsiklis, J.N. and D.P. Bertsekas, M. Athans. Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms. *IEEE Trans. on Automatic Control*, AC-31(9):803–812, 1986.

[68] Tsitsiklis, J.N. and M. Athans. On the Complexity of Decentralized Decision Making and Detection Problems. *IEEE Trans. on Automatic Control*, AC-30(5):440–446, 1985.

[69] Tsypkin, YA. Z. *Adaptation and Learning in Automatic Systems.* Academic Press, 1971.

[70] Tsypkin, YA. Z. *Foundations of the Theory of Learning Systems.* Academic Press, 1973.

[71] Van Trees, H.L. *Detection, Estimation, and Modulation Theory*, volume 1. J. Wiley, New York, 1968.

[72] Wasan, M.T. *Stochastic Approximation.* Cambridge University Press, 1969.

[73] Wassel, G. and J. Sklansky. Training a One-Dimensional Classifier to Minimize the Probability of Error. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-2(4), Sept. 1972.

[74] Wassel, G.N. *Training a Linear Classifier to Optimize the Error Probability.* PhD thesis, Univ. of California, Irvine, 1972.

[75] Wilde, D. *Optimum Seeking Methods.* Prentice-Hall, Inc., 1964.

[76] Willsky, A.S. and J.H. Shapiro. *Stochastic Processes, Detection, and Estimation: Course Notes 6.432.* MIT, Cambridge MA., 1988.

[77] Wissinger, J.W. and M. Athans. A Nonparametric Training Algorithm for Decentralized Binary Hypothesis Testing Networks. In *Proceedings of the 1993 American Control Conference, San Francisco, CA*, 1993.