Problems in Decentralized Detection

by

William W. Irving

S.B., Electrical Engineering, Massachusetts Institute of Technology
(1987)

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

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

Massachusetts Institute of Technology

May, 1991

Signature of Author _____
Department of Electrical Engineering and Computer Science
May, 1991

Certified by _____
John N. Tsitsiklis
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chair, Department Committee on Graduate Students

PROBLEMS IN DECENTRALIZED DETECTION

by

William W. Irving

Submitted to the Department of Electrical Engineering
on May 1, 1991 in partial fulfillment of the
requirements for the Degree of Master of Science in
Electrical Engineering

ABSTRACT

A decentralized Bayesian hypothesis testing problem is considered. It is analytically demonstrated that for the binary hypothesis problem, when there are two peripheral sensors with statistically independent Gaussian-distributed observations (conditioned on the true hypothesis), then there is no loss in optimality in using identical peripheral sensor decision rules.

A conditionally dependent binary hypothesis Gaussian problem is analyzed; strong evidence is supplied that suggests that a threshold strategy is optimal. A conditionally dependent Erlang problem is also analyzed, for which the same conclusion (as for the Gaussian problem) is drawn.

The nonlinear Gauss-Seidel algorithm is tested on a conditionally dependent, binary hypothesis testing problem. Another numerical method, the Chernoff exponent minimizer, is also described

A conditionally independent parameter estimation problem is analyzed. A version of the Cramer-Rao bound is developed. Three heuristically motivated numerical methods are tested on the parameter estimation problem.

Thesis Supervisor: Dr. John N. Tsitsiklis
Title: Associate Professor of Electrical Engineering

Contents

List of Figures

Chapter 1

Introduction

This work is concerned with decentralized detection theory. To impart some appreciation of why this theory is useful, we begin by explaining a possible application. As the application will show, the general problem is multidisciplinary. Thus, to yield a tractable theory, abstraction of many of the problem's details is necessary. We will discuss that we do this abstraction by analyzing simple *probabilistic* models of real physical systems. With respect to these models, decentralized detection theory is very similar to the much more established discipline of centralized detection theory.

Although centralized and decentralized detection theory do both analyze similar mathematical models, there are important nontrivial differences between the two theories. We highlight these differences, and point out the focuses of the decentralized theory. Finally, we discuss the contributions of this thesis.

1.1 Ingredients of a Decentralized Detection Problem

The following scenario motivates the need for a decentralized detection theory. Following the scenario is a summary of the ingredients that are common to all of the detection problems that we will analyze.

1.1.1 Military Application

Somewhere in the battlefield, buried under camouflage, there is an hostile military target. Trying to ascertain the presence or absence of the target is a detection system. The system consists of three radar sensors, each geographically separated from the others, and a fusion center. The fusion center is inside of a manned tank; it is the centralized location where the information from the three radar sensors is combined and processed into a final answer to the question: *is the enemy present or not?* At this point, the scenario divides into two possibilities.

Centralized Detector

In a *centralized* detection scheme, each sensor transmits its entire observation to the fusion center. There, an onboard computer merges all of the sensors' measurements, and somehow processes them all into a final decision.

Decentralized Detector

Practical difficulties may prohibit the use of the centralized scheme. Suppose, for example, that there are space constraints in the tank. Suppose, in fact, that the situation is so tight that the tank operator must act in lieu of an onboard computer as the processor for the fusion center. Clearly, that operator would be flooded with more information than she could handle if the sensors just dumped all of their data on her.

Other difficulties might arise with the centralized scheme, even if there were a sophisticated central computer processor. A likely possibility is that bandwidth limitations preclude transmission of the bulk raw observations from sensors to fusion center.

One eminently reasonable way to handle these difficulties is to equip the radar sensors with local processing capabilities. Then, each sensor boils down its own observation to its essence– down to something that can be captured with only a few bits of information. These few bits are transmitted to the fusion center (requiring much less communication bandwidth), and a human fusion center can fairly readily assimilate the summary information into a final decision.

This alternative is a decentralized detection scheme. It has arisen rather naturally from two needs. First is the need to "share the work," so that no single central location is burdened with an overwhelming amount of processing. Second is the need to limit the the communication burden at the peripheral sensor locations.

1.1.2 More General View

Of course, this detection paradigm is not restricted to the purview of military surveillance. In general terms, we can summarize the problem as follows.

Summary of problem

There is a source that generates an output. The output is transformed (i.e., usually meaning that it is corrupted), and once transformed, it is observed by a collection of sensors. The reason for the sensors is that there is some interesting source attribute whose value we wish to know.

In a centralized scheme, the sensors act as a dummy interface between the observations and the central location. All of the measurements from the sensors are sent to the fusion center, where they are combined to reconstruct the value of the interesting source attribute.

In a decentralized scheme, each sensor locally processes it own observation, thereby capturing some aspect of the observation in the form of a single letter

from a D-valued alphabet. In this study, D will be very small (typically 2, 3, or 4); in practice, D could be much larger. The sensor messages are all sent to the fusion center, where they are combined to reconstruct the value of the interesting source attribute.

Differences between centralized and decentralized systems

There are several important differences between centralized and decentralized detection systems. For one, the decentralized system fusion center must make a final decision on the basis of less information than the centralized fusion center. This follows because the peripheral sensors (in the decentralized system) are only sending summaries to the fusion center; information can only be lost, not gained, in this summarizing process.

Second, the decentralized system has much simpler communication requirements than the centralized system. Thus, in light of the first difference, there is performance tradeoff between accuracy of final decision (favoring the centralized system) versus bandwidth requirements (favoring the decentralized system).

Third, the real-time computation requirements at the fusion center are much simpler in the decentralized system. Typically the decentralized detection fusion rule will be a simple look-up table; in one column is each combination of bits that the fusion center receives as messages from the peripheral sensors, while in the other column is a final decision corresponding to each bit combination. This sort of look-up table would clearly not be reasonable in a centralized scheme; continuous-valued waveform observations are not susceptible to being organized into a table.

Lastly is the issue of the off-line computation needed to find the optimal processors. As we will see, finding the optimal decentralized detection processor (i.e., the summarizing decision rules to be used at the peripheral sensors, plus the look-up table at the fusion center) is much more difficult than finding the optimal processor in the centralized system. It is not at all obvious why the decentralized system is so much more difficult to design, and will only become clear as the theory unfolds.

1.2 Decentralized Detection Theory

Decentralized detection theory provides insights into the tradeoffs in the design of real detection systems. It is a mathematical theory for analyzing mathematical models of detection scenarios. In what follows, we give an example of a decentralized detection problem in the prototypical form of all of the problems that we analyze. We discuss the reason that we have posed the problem as we have, and then we discuss the specific issues that the theory addresses.

1.2.1 Example Problem

> The random variable X models the presence or absence of a target. With probability $1/8$, $X = 1$ (target present), while with probability $7/8$, $X = 0$ (target not present). The value of X is conveyed to two

sensors through the environment which is modeled as a noisy channel. The random variables Y_1, Y_2 model the observation at sensors 1 and 2, respectively. In particular, $Y_i = X + W_i$ $(i = 1, 2)$, where W_1, W_2 are jointly Gaussian random variables with known means and covariance. Each sensor independently processes its own observation into a one bit message, which is sent to the fusion center. Our challenge is this: what summarizing rules should we use at the sensors and what rule do we use to combine the messages (into a final decision) at the fusion center?

1.2.2 Comments on Formulation

There are several noteworthy features of this example.

Probabilistic model

Most importantly, the detection problem is modeled probabilistically. In all of our work, we assume that conditional on the true value of the source, there is a known probability distribution for the observations at the sensors. A probabilistic model is certainly not meaningful or appropriate for all situations. This is not our concern, for no theory is all-encompassing. For many problems, probabilistic models have been used with remarkable success, and it is in these cases that the theory is most useful.

Focus of theory

As the example suggests, the central focus of decentralized detection theory is to find, for a given probabilistic model, the optimal decision rules for the peripheral sensors and for the fusion center. Formally, this set of rules is known as a *strategy*. A strategy is said to be optimal if it minimizes the expected value of some given cost criterion. In hypothesis testing, we use the probability of error cost criterion, and in parameter estimation, we use the mean-square error cost criterion.

The next chapter will show that there exist necessary conditions for optimality of a strategy. These conditions take the form of a set of coupled nonlinear functional equations, which can sometimes be solved numerically, but can virtually never be solved analytically. Since in general it is hopeless to find an optimal strategy, much of decentralized detection theory focuses on finding good, though suboptimal strategies.

Some of the questions that theory attempts to answer are as follows. Is it easy to find suboptimal strategies that perform reasonably well? Can one show that asymptotically, as the number of sensors goes to infinity, that some suboptimal alternative performs only negligibly worse than the optimal? What are efficient techniques for numerically finding strategies? What is the tradeoff between number of sensors and number of bits sent per sensor? For example, is it better to have 12 sensors transmitting 1 bit messages each or 4 sensors transmitting 3 bit messages each? In one guise or another, we will examine all of these issues.

Simplicity of probabilistic model

Note that no justification has been given for the probabilistic model that is used. For that matter, the model is very simple; it is doubtful that it corresponds to some real physical situation with any high degree of accuracy. We might have alternatively begun as follows:

> We want to analyze the problem of detecting some tactical (i.e., military) target that is deployed beneath heavy foliage. We begin the analysis with a treatment of the electromagnetic interaction between radar, foliage and target. From that analysis, we conclude that a deterministic treatment is too difficult. Instead, we should model the electromagnetic signature of the tank as a random process with known first and second order statistics, while everything else (receiver noise, foliage interference and attenuation) is lumped together into a second random process that corrupts the first process. Then, we have two hypotheses: (1) there is no tank present–the observed waveform is a sample of only the second random process; (2) there is a tank present– the observed waveform is a sample of the combined process.

Though this latter formulation is reasonable for a complete treatment of some specific detection problem, it is not appropriate for developing theory. The difficulty is that the mathematical complexity of most realistic models obscures the issues that interest us. This tractability issue is particularly true with decentralized detection theory, where we will see that even the simplest of operational situations (such as the first version of the target detection problem), is challenging to analyze.

In conclusion, we analyze probabilistic models without very much concern for the model's correspondence to physical reality. Usually a given model is motivated by some important aspect of a real situation, but we do not dwell on the analogy. Also, to reiterate, we always assume that part of the model is a probability distribution for the observations at the sensors.

1.3 Contributions of this Thesis

The decentralized detection problem was introduced a decade ago in [TS81] (many more references will be given in the next chapter). From the discipline's inception, most research has focused on the binary hypothesis testing problem for the special case of independent sensor observations, conditional on the true hypothesis. In the context of our radar detection problem, for example, the conditional independence restriction means that the noise corrupting the electromagnetic signature observed by any given sensor is statistically independent of the noise corrupting the signature received by any other sensor.

As we will see in the next chapter, the conditional independence assumption leads to significant simplifications to the search for optimal decision rules for the sensors. The reason is that with no loss of optimality, each sensor can use a likelihood ratio test to yield its decision. However, while the assumption yields com-

putational advantages, it does not always make physical sense. For example, if two radar sensors are in close proximity, then they are quite likely to have correlated noise corruption.

A primary focus of this work will be the binary hypothesis testing problem with conditionally *dependent* observations. For the general binary hypothesis, conditionally dependent problem, the optimal peripheral sensor rule will NOT be a likelihood ratio test. However, we will see that for certain problems, the likelihood ratio test is optimal. The complete argument of this result is analytically intractable. However, we can carry the analysis far enough to motivate the result, at which point we can finish the analysis with numerical methods.

The next focus is on numerical methods for finding good, though possibly suboptimal, strategies for the hypothesis testing problem. We set up two explicit decentralized detection problems, and analyze them with different numerical methods. The methods that we try are not new, but what is novel is that we bring the methods together so that a direct performance comparison can be made. This is in contrast to much of the literature, where only a single technique is tried on a given problem. In general, the results of the numerical experiments provide some comfort that good (though suboptimal) strategies can be found without excessive computation.

Third, we consider the parameter estimation problem. We derive some general bounds on performance. Then, we set up a particular problem. For the given problem, we use different search techniques to find good, but suboptimal, strategies. We compare the strategies found by the different techniques.

Chapter 2

Preliminaries

In this chapter, we introduce many definitions and concepts that will be useful throughout the rest of the work. Most of this material is well-known in the decentralized detection field, and so many references will be given. Here is an outline of this chapter.

First, there is a comment on notation convention. Our notation conforms fairly closely with standard practice, and will be followed throughout the thesis.

Next is a formal statement of our version of the Bayesian hypothesis testing problem. Roughly speaking, the scenario is as follows. There is a finite set of mutually exclusive events (formally known as hypotheses), each with a known prior probability of occurring. One of these events occurs, and its occurrence manifests itself as some observable quantity at a collection sensors. Each sensor transmits a summary of its observation to a central location. At that location, all the summaries are combined to make a (hopefully good) guess of which event really occurred. The problem is to coordinate all the decision rules so as to get the best possible guess.

Following the problem formulation is the definition of a strategy. This is a formal term for the collection of decision rules used by the sensors. We discuss both deterministic and randomized strategies. Then, we develop necessary conditions for optimality of a strategy.

As will become apparent, there is (in general) no closed-form analytic expression for an optimal strategy. Thus, numerical methods are needed. One numerical method is a sort of exhaustive search over all strategies. We discuss the computational complexity of this brute force approach, and thereby demonstrate the combinatorial complexity of the problem.

Lastly, there is a brief discussion of the Neyman-Pearson variant of the decentralized detection problem. This variant is useful when either it is not meaningful or is not possible to assign prior probabilities to the hypotheses.

2.1 Notation Convention

This is a summary of the notation convention that is used throughout this paper. First, with a few exceptions as noted, upper-case letters are used to denote random

variables, while lower-case letters are used to denote realizations of random variables. For example, $\Pr(X = x)$ is the probability that the random variable X takes on the particular value x.

The primary exception to this rule is that H_j is used, as opposed to h_j, to denote the occurrence of the jth hypothesis. This exception is made in order to conform to standard usage. Another exception is that upper-case letters are used to denote certain important constants and parameters. For example, M is the number of hypotheses, and N is the number of peripheral sensors. Context will hopefully eliminate any possible confusion about whether a variable is random or not.

The unconditional marginal distribution of continuous random variable Y is denoted by $f_Y(y)$, while its conditional marginal distribution (conditioned on, say, H_1) is denoted by $f_{Y|H_1}(y|H_1)$. The unconditional joint distribution of continuous random variables Y_1, ..., Y_N is denoted by $f_{Y_1,...,Y_N}(y_1,...,y_N)$, while their joint conditional distribution (conditioned on H_1) is denoted by $f_{Y_1,...,Y_N|H_1}(y_1,...,y_N|H_1)$.

A different notation is used for the distribution of discrete random variables. The unconditional, joint probability that discrete random variables U_1, ..., U_N are equal to u_1, ..., u_N, respectively, is denoted by $\Pr[U_1 = u_1, ..., U_N = u_N]$, or sometimes more succinctly by $\Pr[u_1, ..., u_N]$. The notation for the conditional, marginal probability for U, and for all the other permutations, should now be self-evident and so we belabor the point no further.

When there is no ambiguity, the limits on integrals and summations are omitted. For multiple integrals, we define

$$\int g(y_1,...,y_N)\, dy_1...dy_N \equiv \int_{l_1}^{h_1} \cdots \int_{l_N}^{h_N} g(y_1,...,y_N)\, dy_1...dy_N, \tag{2.1}$$

where l_1, h_1, ..., l_N, h_N are limits whose values are supplied by context. Similarly, for multiple summations, we define

$$\sum_{u_1,...,u_N} g(u_1, ..., u_N) \equiv \sum_{u_1=l_1}^{h_1} \cdots \sum_{u_N=l_N}^{h_N} g(u_1, ..., u_N). \tag{2.2}$$

The relation

$$W \sim N(a, b)$$

means that W is a real scalar Gaussian random variable with mean a and variance b; that is,

$$f_W(w) = \frac{\exp[-(w-a)^2/2b]}{\sqrt{2\pi b}}.$$

The notation

$$f(u) \sim g(u), \quad u \to \infty \tag{2.3}$$

means that in the limit of u going to infinity, $f(u)$ is asymptotically equal to $g(u)$; formally, this is

$$\lim_{u\to\infty} \frac{f(u)}{g(u)} = 1. \tag{2.4}$$

Note that our symbol for asymptotic equality is the same symbol used to denote Gaussian random variables. Context will always resolve this ambiguity.

We define

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-t^2/2)\, dt. \tag{2.5}$$

This is the cumulative distribution function for a zero-mean, unit variance Gaussian random variable. There is no closed-form solution to the integral, but its values are widely tabulated. We will exclusively use the $\Phi(\cdot)$ notation for this integral.

There are some important parameters that are defined in Section 2.2. For completeness, we also define them here: N is the number of peripheral sensors; M is the number of hypotheses; D is the alphabet size of the messages at the peripheral sensors.

The notation

$$y_i \underset{H_1}{\overset{H_2}{\gtrless}} T \tag{2.6}$$

is used throughout much of this thesis to denote a particular type of decision rule. The definition of (2.6) is deferred to Section 3.2, where the notation is first used.

This concludes a fairly complete summary of our notation convention. There are some notational quirks that we have not mentioned, but these are all defined at the time of their use, and additionally, their use is confined to a single section.

2.2 Formulation of Bayesian Hypothesis Testing Problem

We formulate the Bayesian version of the decentralized hypothesis testing problem. Then we make some comments about the particular way that we have made our formulation.

2.2.1 Formulation

We have an environment that is in one of M ($M \geq 2$) possible states. Each state of the environment corresponds to one hypothesis; when the environment is in state i ($1 \leq i \leq M$), we say that hypothesis H_i has *occurred* or that H_i is *true*. The hypotheses occur with positive a priori probabilities $\Pr(H_1)$, ..., $\Pr(H_M)$.

Detecting the state of the environment are N ($N \geq 1$) *peripheral sensors*. Sensor i ($1 \leq i \leq N$) receives scalar, real-valued environment observation y_i. We assume that conditioned on hypothesis H, the realizations of the sensor observations obey the known joint probability density function $f_{Y_1,...,Y_N|H}(y_1, ..., y_N|H)$.

Each peripheral sensor, upon receiving its observation, evaluates a message, $u_i = \gamma_i(y_i) \in \{1, ..., D\}$, where $\gamma_i(\cdot)$ is deterministic, and D is typically small (typically,

D is 2, 3, or 4). Each peripheral sensor sends its message to the *fusion center*, which we label as the zeroth sensor. We assume that we have noiseless channels for the transmission of these messages. The fusion center uses the N peripheral sensor messages to evaluate a final decision, $u_0 = \gamma_0(u_1, ..., u_N) \in \{1, ..., M\}$.

When the true hypothesis is H, the cost of a fusion center decision of u_0 is $C(u_0, H)$, where

$$C(i,j) = \begin{cases} 0, & i = j, \\ 1, & i \neq j. \end{cases} \tag{2.7}$$

This is known as the *probability of error* cost criterion.

The M-ary hypothesis decentralized detection problem can now be summarized as follows:

Given

- a priori probabilities, $\Pr(H_1)$, ..., $\Pr(H_M)$, and
- joint probability density function, $f_{y_1,...,y_N|H}(y_1, ..., y_N|H)$ for each hypothesis,

we want to minimize

$$\mathrm{E}[C(U_0, H)] = \sum_{j=1}^{M} \Pr(H_j) \Pr[\gamma_0(U_1, ..., U_N) \neq j | H_j], \tag{2.8}$$

where

$$\Pr[\gamma_0(U_1, ..., U_N) \neq H | H] = \int C[\gamma_0(\gamma_1(y_1), .., .\gamma_N(y_N)), H]\ f_{y_1,...,y_N|H}(y_1, ..., y_N|H)\ dy_1 ... dy_N \tag{2.9}$$

In (2.8) and (2.9), we have explicitly shown the dependence of the expected cost ($E[C(U_0, H)]$) on each of the decision rules.

What choice of $\gamma_0(\cdot), \gamma_1(\cdot), ..., \gamma_N(\cdot)$ will minimize this expectation?

The decentralized Bayesian hypothesis testing problem was first formulated in [TS81]. We will have more to say about their groundbreaking work, and also about the work of other early contributors, after we discuss person-by-person optimality conditions in Section 2.4.

2.2.2 Comments on the Formulation

We comment on aspects of our formulation that could be generalized or modified.

Our restriction to scalar observations at the peripheral sensors is made mostly to simplify our numerical examples in later chapters. In an analytical sense, our results are easily extensible to the case of vector observations at the peripheral sensors. In [LS82], authors go even farther, as they consider waveform observations at the peripheral sensors. Even this case is closely related to the results that we

will obtain. In particular, if it is possible to break each waveform observation into a set of Karhunen-Loeve coefficients, then we are led to a vector of coefficients at each peripheral sensor, which, in turn, takes us back to our comment about vector observations. Let it simply be said that the scalar observation version of the problem will present more than enough difficulty.

We have assumed that the sensor observations have a probability density. This is mostly for notational convenience. Most of the results that we obtain extend readily to the case of discrete sensor observations. In fact, many of our numerical examples will be with discrete problems.

In some of the formulations in the literature, the fusion center receives not only the N peripheral sensor messages, but also an observation, Y_0, of its own. There are some complications that arise in conjunction with this more generalized formulation. We defer consideration of these issues to future work.

Some authors consider a more general cost criterion in which the only restriction is that

$$C(i,j) > C(i,i), \quad i \neq j.$$

This generalization renders some of our results more algebraically burdensome without adding insight. For this reason, we avoid the generalization. For the same reason, we avoid the generalization of letting the number of messages, D, be a sensor-dependent quantity.

2.3 Strategies

We define a deterministic *strategy* to be any collection of decision rules, $\gamma = (\gamma_0, ..., \gamma_N)$. We define Γ_i to be the set of all deterministic decision rules for sensor i. Then, the set of all deterministic strategies, Γ, is given by $\Gamma = \Gamma_0 \times ... \times \Gamma_N$.

As a generalization of deterministic strategies, we also consider randomized strategies. Associated with any randomized strategy is a finite collection of deterministic strategies, γ^1, ..., γ^r, $r \geq 1$, and a random variable C with probability mass function $\Pr[C = i] = P_i$, $1 \leq i \leq r$. Random variable C is independent of everything else in the detection problem. When $C = i$, then deterministic strategy γ^i, $1 \leq i \leq r$ is used. We define the set of all randomized strategies by Γ^*.

In the Bayesian formulation of the last section (Section 2.2), we explicitly restricted attention to deterministic strategies. The next proposition shows that that restriction is harmless.

Proposition 1

$$\inf_{\gamma \in \Gamma} J(\gamma) = \inf_{\gamma \in \Gamma^*} J(\gamma). \tag{2.10}$$

Proof: From the definitions of Γ, and Γ^*, it is obvious that $\Gamma \subset \Gamma^*$. But from elementary probability, we also know that if $\gamma^* \in \Gamma^*$, then

$$J(\gamma^*) = \sum_j P_j J(\gamma^j),$$

for deterministic strategies $\gamma^j \in \Gamma$. Thus, we easily conclude that (2.10) holds. **QED**. This result is from [T89].

When we later consider the Neyman-Pearson version of the problem, we will need the added generality of randomized strategies.

2.4 Bayesian Person-by-Person Optimality Conditions

We develop necessary conditions for an optimal strategy. These conditions are very important, and in one guise or another, they will be used in almost every section that follows.

The conditions for the fusion rule are different from the conditions for the peripheral sensor rules, and so we treat the two types of rules separately.

2.4.1 Fusion Rule

For the fusion rule, we have the following proposition.

Proposition 2 *With respect to fixed peripheral sensor decision rules* $\gamma_1(\cdot), ..., \gamma_N(\cdot)$, *the optimal fusion rule is*

$$\gamma_0(u_1, ..., u_N) = \underset{j=1,...,M}{arg\ max} \{Pr(H_j)\ Pr[u_1, ..., u_N|H_j]\}. \quad (2.11)$$

Proof: We first expand (2.8) to the convenient form

$$\mathrm{E[C(}U_0\mathrm{,H)]} = \sum_{u_1,...,u_N} \Pr[u_1, ..., u_N]\ \Pr[\gamma_0(u_1, ..., u_N) \neq H|u_1, ..., u_N]. \quad (2.12)$$

The choice of $\gamma_0(\cdot)$ that will minimize $\mathrm{E[C}(U_0, H)]$ is the same as the choice of $\gamma_0(\cdot)$ that will minimize each of the summands on the right-hand side of (2.12). This is so because each summand corresponds to a different set $\{u_1, ..., u_N\}$ of messages at the fusion center. Each summand is minimized via the fusion rule

$$\begin{aligned}
\gamma_0(u_1, ..., u_N) &= \underset{d=1,...,M}{\arg\min} \left\{ \Pr[u_1, ..., u_N] \sum_{j=1}^{M} \Pr[H_j|u_1, ..., u_N] C(d, H_j) \right\} \\
&= \underset{d=1,...,M}{\arg\min} \left\{ \sum_{j=1}^{M} \Pr[u_1, ..., u_N|H_j] \Pr(H_j) C(d, H_j) \right\} \\
&= \underset{j=1,...,M}{\arg\max} \{\Pr[u_1, ..., u_N|H_j] \Pr(H_j)\} \qquad (2.13)
\end{aligned}$$

From the first line to the second line, we used Bayes rule,

$$\Pr[H|u_1, ..., u_N] \Pr[u_1, ..., u_N] = \Pr[H]\ \Pr[u_1, ..., u_N|H].$$

Then the next equality follows easily from the form of $C(u_0, H)$), and so the result is proved. **QED.**

The rule in (2.11) is familiar from classical centralized hypothesis testing theory as the maximum aposteriori probability (MAP) rule. This should not be surprising. We can view the set of peripheral sensor messages as a vector of observations that the fusion center receives (this interpretation of the peripheral sensor messages was first made in [CV86]). Once the probability mass function (PMF) is computed via

$$\Pr[U_1 = u_1, \ ..., \ U_N = u_N|H] = \int_{\gamma_1^{-1}(u_1)} \cdots \int_{\gamma_N^{-1}(u_N)} f_{Y_1 \ldots Y_N|H}(y_1, ..., y_N|H) dy_1 \cdots dy_N,$$

(where $\gamma_i^{-1}(u)$ is the portion of the real axis for which $\gamma_i(y) = u$), then the centralized interpretation becomes complete.

Though the form of the fusion rule is familiar, the decentralization of our problem still adds complications. Most importantly, an optimal strategy's fusion rule can only be explicitly stated when optimal peripheral sensor rules are known. But, as we will see in the next section, the optimal peripheral sensor rules are mutually dependent and are also dependent on the optimal fusion rule. Everything is tightly intertwined.

2.4.2 Peripheral Sensor Rules

The following proposition gives necessary conditions for optimal peripheral sensor decision rules.

Proposition 3 *With respect to fixed sensor decision rules*

$$\gamma_0(\cdot), \ ..., \ \gamma_{i-1}(\cdot), \ \gamma_{i+1}(\cdot), \ ..., \ \gamma_N(\cdot),$$

the optimal peripheral sensor rule, $\gamma_i(\cdot)$ is

$$\gamma_i(y_i) = \underset{d=1,...,D}{arg\ min} \sum_{j=1}^{M} f_{Y_i|H_j}(y_i|H_j) Pr(H_j) a_i(d, H_j, y_i), \qquad (2.14)$$

where

$$a_i(d, H, y_i) = Pr[\gamma_0(U_1, ..., U_{i-1}, d, U_{i+1}, ..., U_N) \neq H|H, y_i]. \qquad (2.15)$$

Proof: To keep the notation less burdensome, we derive (2.14) for the particular case, $\gamma_1(\cdot)$. The generalization to $\gamma_i(\cdot)$ will then follow easily.

To find the optimal $\gamma_1(\cdot)$, we expand (2.8) to the convenient form,

$$E[C(U_0, H)] = \int E\{C[\gamma_0(\gamma_1(y_1), U_2, ..., U_N), H]|y_1\} \ f_{Y_1}(y_1) \ dy_1 \qquad (2.16)$$

The choice of $\gamma_1(\cdot)$ that will minimize $E[C(U_0, H)]$ is the same as the choice of $\gamma_1(\cdot)$ that will minimize the integrand in (2.16). Hence, for each value of Y_1, we have the optimal $\gamma_1(\cdot)$ relation

$$\gamma_1(y_1) = \arg\min_{d=1,...,D} f_{Y_1}(y_1) E\{C[\gamma_0(d, U_2, ..., U_N), H] | y_1\} \tag{2.17}$$

$$= \arg\min_{d=1,...,D} f_{Y_1}(y_1) \sum_{j=1}^{M} \Pr[H_j|y_1] \Pr[\gamma_0(d, U_2, ..., U_N) \neq H_j | H_j, y_1] \tag{2.18}$$

In (2.18), we expanded the expectation in (2.17). But then, by Bayes rule,

$$f_{Y_1}(y_1) \Pr[H_j|y_1] = \Pr[H_j] \, f_{Y_1|H_j}(y_1|H_j),$$

and so (2.18) is equivalent to (2.14). **QED**

We see that for each peripheral sensor rule, the optimality conditions are completely specified once the a-coefficients are known; however, the a-coefficients are a function of all of the other sensor rules. Once again, we see a tight interdependence of the optimality conditions.

2.4.3 Necessity and Sufficiency: Person-by-person Optimality

We introduce an important definition.

> A strategy is *person-by-person optimal* (pbp-optimal) if for $0 \leq i \leq N$, and for fixed $\gamma_j(\cdot)$ $(0 \leq j \leq N, j \neq i)$, we have $\gamma_i(\cdot)$ satisfying (2.11) when $i = 0$ or satisfying (2.14) when $i > 0$.

Conditions (2.11) and (2.14) are pbp-optimality conditions. All globally optimal strategies are pbp-optimal, but the converse is not true. The following example shows that a suboptimal strategy can be pbp-optimal.

- **Example**

  We have a source that transmits a signal, S, that is either zero or one. In particular,

$$S = \begin{cases} 0 & \text{under hypothesis } H_1 \\ 1 & \text{under hypothesis } H_2 \end{cases},$$

  with $\Pr(H_1) = \Pr(H_2) = 1/2$.

  Receiving the signal are two peripheral sensors. The signal observations are contaminated by Gaussian noise,

$$Y_i = S + W_i, \quad W_i \sim \mathrm{N}(0, 1), \quad i = 1, 2$$

with W_1 and W_2 independent.

Finally, $\gamma_i(y_i) \in \{1,2\}, i = 1,2$. Our goal is to find a strategy, γ that minimizes the probability of error.

Surprisingly, one pbp-optimal strategy is the rule

$$\gamma_1(y) = \gamma_2(y) = 1, \quad \forall y,$$

for the peripheral sensors, and the rule

$$\gamma_0(u_1, u_2) = 1, \quad \forall u_1, u_2.$$

for the fusion center. The pbp-optimality of this can be directly checked by evaluating (2.11) and (2.14).

This strategy has a probability of error of 0.5. If it is the optimal strategy, then the discipline of decentralized detection is doomed. Actually, the optimal strategy has a cost of 0.271; the issue of finding a strategy with that optimal cost will be explored in Chapter 4.

Note that a pbp-optimal strategy is not necessarily even locally optimal; while perturbing no single sensor rule can improve performance, pbp-optimality says nothing about the simultaneous perturbation of many sensor rules.

2.4.4 Rapprochement: General Pbp-Optimality Conditions

So far, we have considered the pbp-optimality conditions for the general case of any number of hypotheses and for any sort of observation probability density function. The form of the conditions (a set of $(N+1)$ coupled nonlinear functional equations) is not particularly insightful. There is no known closed-form solution to the equations. In fact, for many problems, it is more realistic to do an exhaustive search of all strategies (in some discretized version of the observation space Y_1, ..., Y_N) than it is to solve the pbp-optimality conditions.

In the next section, we consider the binary hypothesis problem with mutually independent observations, conditioned on either hypothesis. We will see that for this special case, likelihood ratio tests can optimally be used for the peripheral sensor rules. Thus, the situation is not as uniformly bleak as it might now appear.

2.5 Binary Hypothesis Problem with Conditionally Independent Observations

If the peripheral sensor observations are mutually independent, conditioned on the true hypothesis, then

$$f_{Y_1,\ldots,Y_N|H}(y_1, \ldots, y_N|H) = f_{Y_1|H}(y_1|H) \cdot \ldots \cdot f_{Y_N|H}(y_N|H). \tag{2.19}$$

In the sequel, when (2.19) holds, and there is no ambiguity, we will succinctly refer to (2.19) as the conditional independence property.

Conditional independence is a very restrictive property. Equation 2.19 is violated in at least two important cases:

- Detection of a known signal in sensor-by-sensor correlated noise,
- Detection of an unknown signal in noise.

If, for example, we are modeling a system with geographically distributed sensors, then closely situated sensors are likely to have correlated noise in their observations. Conditional independence is thus not only restrictive in a pure mathematical sense, but is also restrictive in a practical, modeling sense.

2.5.1 Threshold Strategies

On the positive side, though, conditional independence leads to considerably more structure in optimal strategies for the binary hypothesis testing problem. To precisely describe this structure, we need some definitions; the following is patterned after [T89].

> Suppose that there are thresholds $t_1, ..., t_{D-1}$ satisfying $0 \leq t_1 \leq ... \leq t_{D-1}$, and that there are intervals $I_1 = [0, t_1], I_2 = [t_1, t_2], ..., I_D = [t_{D-1}, \infty]$. Then, decision rule $\gamma_i(\cdot)$ is called a *monotone threshold rule* if
>
> $$\gamma_i(y_i) = d \quad \text{only if} \quad L_i(y_i) \in I_d,$$
>
> where
>
> $$L_i(y_i) = \frac{f_{Y_i|H_2}(y_i|H_2)}{f_{Y_i|H_1}(y_i|H_1)}.$$
>
> More generally, decision rule $\gamma_i(\cdot)$ is called a *threshold rule* if there exists a permutation mapping $\sigma : \{1, ..., D\} \rightarrow \{1, ..., D\}$ such that $\sigma \circ \gamma_i$ is a monotone threshold rule.
>
> Finally, we say that any strategy is a *threshold strategy* if all of the peripheral sensors use threshold rules.

2.5.2 When threshold strategies are optimal

The following proposition states when threshold strategies are optimal.

> **Proposition 4** *Assume that an optimal strategy exists. Also, let $M = 2$ (i.e., binary hypothesis testing problem) and let (2.19) hold. Then, there exists an optimal strategy that is a deterministic monotone threshold strategy.*

This is one of the most significant results in decentralized detection theory. It was first established (for a special case) in [TS81], and since then has been refined by various authors. Since the proof can be found in many places, we only outline the portion that is insightful to work later in the thesis.

Outline of Proof: We manipulate (2.14) and (2.15) in order to prove the result. First, note that in the conditionally independent case, $a(d, H, y_i)$ (defined in Equation 2.15) is independent of y_i; hence we can write $a(d, H)$.

The simplification of the a-coefficients, together with the fact that $M = 2$, means that (2.14) can be rearranged to

$$\gamma_i(y_i) = \arg\min_{d=1,\ldots,D} [b_i(d,1) + b_i(d,2)L(y_i)], \tag{2.20}$$

where $b_i(d,j) = \Pr(H_j)a_i(d,H_j)$ and the likelihood ratio is assumed to be well-defined. The key to proving this result is to note that for fixed d, the expression in parentheses in (2.20) is a line (as a function of L). Thus, there are D lines corresponding to the D different messages, and any given message is optimal only for the portion of the L-axis for which the corresponding line is the minimum. But, it is easy to see that the minimizing range of L (corresponding to a given message) can be expressed as the interval $T' \leq L < T''$, for some T', T''. There will never be two (or more) disjoint regions in L-space for which a given message is uniquely optimal. Thus, we have established that there exists an optimal strategy that is a deterministic threshold strategy. We omit the proof that there exists an optimal determinisitic *monotone* threshold strategy. **QED**

This proof has been based on [T89], where the geometric argument is carried out more fully, and where the proof covers the monotone threshold strategy case.

When the conditional independence assumption is relaxed, there is no longer a guarantee of existence of an optimal strategy within the class of threshold strategies. Thus, the conditional independence / binary hypothesis combination is a powerful one. In particular, as we will see in sections to follow, it allows for drastic reduction in the number of strategies that must be examined in the search for an optimal strategy.

2.5.3 Historical Notes

As mentioned earlier, the Bayesian decentralized detection problem was first considered in [TS81]. They only examined the binary hypothesis, two sensor, binary message case. In that work, the fusion center was always fixed a priori; they did not note the fusion center pbp-optimality conditions. However, their work did contain all of the ingredients necessary to prove Proposition 3 in its generality here.

In [S86], the author attempted to extend the pbp-optimality conditions to the $M > 2$, $N > 2$ case. Unfortunately, his derivation is vague, and is quite difficult to follow. As in [TS81], the author does not consider data fusion. In [CV86], the fusion center pbp-optimality conditions were finally proved.

The work in [RN87] was the first to explicitly show the interplay between the optimal fusion center and the optimal peripheral sensor rules. Unlike earlier works, which assumed one fixed (fusion center or set of peripheral sensors) and then optimized with respect to the other, [RN87] demonstrated the need to simultaneously

optimize with respect to both the fusion rule and with respect to the peripheral sensor rules.

Our derivations are most closely patterned after [T89], which provides an excellent overview of the whole field of decentralized detection.

2.6 Strategies with Identical Peripheral Sensor Rules

This section continues the examination of the structure of optimal strategies. So far, we have shown that for the binary hypothesis, conditionally independent problem, an optimal strategy can always be chosen to be a threshold strategy. Here, we consider the binary hypothesis testing problem with peripheral sensor observations Y_1, ..., Y_N that are identically distributed conditional on either hypothesis.

By symmetry, one might expect that there exists an optimal deterministic strategy in which all of the peripheral sensors use identical decision rules. If this were true, then it would drastically reduce the computational complexity of searching for an optimal strategy.

Unfortunately, in spite of the intuitive appeal of the conjecture, it is not true. We separately examine the issue further in the conditionally independent and dependent cases. The reason for treating the cases separately will be evident as the discussion unfolds.

2.6.1 Conditionally Independent Observations

Consider the binary hypothesis testing problem with Y_1, ..., Y_N identically distributed and conditionally independent, given either hypothesis. There is an example in [T88] of an instance of such a problem in which the peripheral sensor rules *cannot* be identical in the optimal strategy.

The identical sensor rule does not always hold for an arbitrary number of sensors, N, (as the example in [T88] proves) but it does hold asymptotically as $N \to \infty$ for the probability of error cost criterion. This was shown in [T88] by using a particular form of Chernoff bounds from [SGB67].

Returning to the finite N case, one might wonder about worst case reduction in performance when all of the peripheral sensors are constrained to use the same decision rule. Limited results are available. In [PA90], the authors define the *relative deterioration* of performance by

$$\text{relative deterioration} = \frac{P^{error}_{identical} - P^{error}_{optimal}}{P^{error}_{optimal}}.$$

For the case of $N = 2$, the authors show that the least upper bound on the relative deterioration is 1. Their analysis is not readily extensible to the $N > 2$ case; no worst case deterioration results are available for more than two peripheral sensors.

Despite the above caveats, experimental results in both [SR83] and [RN87] show that for many particular problem instances with $N = 2$ or 3, the optimal strategy

does use identical peripheral sensor rules. Also, in Chapter three, we analytically demonstrate that for the Gaussian problem in the example of Section 2.4.3, there is no loss in optimality in using identical peripheral sensor rules.

2.6.2 Conditionally Dependent Observations

Now consider the binary hypothesis testing problem with Y_1, ..., Y_N identically distributed (given either hypothesis), but conditionally dependent.

The following example shows that if identical deterministic decision rules are used at the peripheral sensors, then

1. Performance can be far from optimal, even if an infinite number of peripheral sensors are used,

2. The relative deterioration can be infinite when two peripheral sensors are used.

Both of these results are in contrast to the properties of the conditionally independent problem, as discussed in the previous section.

- **Example**

  We have a source that transmits a signal, S, to a set of N peripheral sensors. The observation model is

$$\begin{aligned} \Pr[W = j] &= 1/2, \quad j = 0, 1; \\ \Pr[S = k] &= 1/3, \quad k = 0, 1, 2, \\ Y_i &= S + W. \end{aligned}$$

  We define H_1 to be true when $W = 0$ (i.e., the channel between the source and sensors is not corrupted), and we define H_2 to be true when $W = 1$ (i.e., the channel is corrupted). We want to minimize the probability of error.

  Suppose that the fusion center is colocated with the source (so that the fusion center always knows what the source transmitted). Also, let $D = 2$, and let $N = 2$. We want to find a strategy that minimizes the probability of error.

  If we are not constrained to use identical rules, then the following strategy will clearly have an error of 0:

$$\gamma_1(y_1) = \begin{cases} 0 & \text{for} \quad y_1 = 0, 1 \\ 1 & \text{for} \quad y_1 = 2, 3 \end{cases},$$

$$\gamma_2(y_2) = \begin{cases} 0 & \text{for} \quad y_2 = 0, 2 \\ 1 & \text{for} \quad y_2 = 1, 3 \end{cases},$$

$$\gamma_0(u_1, u_2) = \begin{cases} 0 & \text{iff } 2u_1 + u_2 = s \\ 1 & \text{otherwise} \end{cases}$$

On the other hand, for deterministic identical sensor rules, it is impossible to achieve a probability of error of zero, no matter how many sensors we employ. To see this, note that since $Y_1 = Y_2 = ... = Y_N$ (under either hypothesis), it follows that with identical rules, $U_1 = U_2 = ... = U_N$. Therefore, with no loss of information, we need only examine U_1, which represents only a single bit of information. Clearly, a single bit does not convey enough information to the fusion center for determining whether the peripheral sensor observations have been corrupted. Thus, optimal performance cannot be achieved with deterministic identical rules. Also, note that for $N = 2$, the relative deterioration is infinite.

Despite the caveat of this example, we will see through numerical experimentation in later chapters that identical decision rule strategies often perform at or just slightly below optimality. The example is a worst-case result, but is probably not typical.

2.7 Computational Complexity of Binary Hypothesis Problem

Suppose that the observation space at each peripheral sensor is discrete with cardinality A. Then, as an alternative to solving the pbp-optimality conditions for finding an optimal strategy, there is the option of doing an exhaustive search. That is, the alternative is to exhaustively consider the cost of all deterministic strategies, and to choose one that has minimum cost. This alternative is borne from the perpective that the pbp-optimality conditions are too difficult to solve, and so the problem of searching for an optimal strategy should be treated as a purely combinatorial one.

Here is a summary of what follows. After introducing some special notation and addressing some secondary details, we count the number of monotone threshold strategies that are possible in a binary hypothesis testing problem. This number is relevant to the conditionally independent problem. Next, we count the number of general strategies that are possible in a binary hypothesis testing problem; this number is relevant to the conditionally dependent problem.

Finally, we count the number of identical rule general strategies and identical rule monotone threshold strategies. From the discussion in Section 2.6, we know that these identical rule strategies are not guaranteed to be optimal, but (as we will see) the reduction in the number of strategies over which we must search may more than compensate for the resulting suboptimal performance.

2.7.1 General Considerations

The following are some general comments that pertain to our entire computational complexity discussion.

Notation

As we discuss exhaustive searches over strategies, we need a shorthand notation to describe particular decision rules. To describe our notation, first suppose that $A = 5$ and $D = 3$. Then, we define the following equivalence

$$\text{Peripheral sensor } i \text{ uses the rule 11223} \iff \gamma_i(y_i) = \begin{cases} 1 & \\ 2 & \text{iff} \\ 3 & \end{cases} \begin{array}{ll} y_i = 1,2, \\ y_i = 3,4 \\ y_i = 5. \end{array}$$

Equivalent Peripheral Decision Rules

The following definition describes when two peripheral sensor decision rules are equivalent.

> Two peripheral sensor rules, $\gamma(\cdot)$ and $\gamma'(\cdot)$, are said to be *equivalent* if there exists a permutation mapping $\sigma : \{1, ..., D\} \rightarrow \{1, ..., D\}$ such that $\gamma(y) = \sigma(\gamma'(y))$, $\forall y$.
>
> The set of all decision rules that are equivalent is called an *equivalent class.*

We describe the above relationship as one of equivalence, because two equivalent decision rules convey identical information to the fusion center. If we are exhaustively searching for an optimal strategy, then there is no reason to examine more than one decision rule from each equivalent class. Note that if all D messages are used in a decision rule, then there are $D!$ rules in the given rule's equivalent class.

Insistence that all messages be used

The following proposition will help us yield modest computational savings when A is small.

> **Proposition 5** *Assuming that $A \geq D$, an optimal strategy can always be found that uses all D message values in all of the peripheral sensor rules.*

Proof (Outline): This result is essentially obvious. An example will convey why it must be so. Suppose that $D = 3$, $A = 5$, and as part of an optimal strategy, peripheral sensor i uses the rule 11222. Then, peripheral sensor i can also use the rule 11223. The fusion center merely has to compensate by treating both 2s and 3s from peripheral sensor i as 2s. **QED**.

Fusion

As we systematically search over different strategies, we need only vary the peripheral sensor rules. For each set of peripheral sensor rules the corresponding optimal fusion rule follows automatically from Proposition 2.11.

Complexity of computing strategy cost

The probability of error for any given strategy can be computed with $O(NA + D^N)$ arithmetic operations [T89]. The first term accounts for the effort needed to read the given strategy. The second term accounts for the computation needed to consider each combination of messages that the fusion center receives.

2.7.2 Conditionally Independent, Binary Hypothesis

If the conditional independence assumption (2.19) holds, then by Proposition 4, we can restrict attention to the class of threshold strategies. With this in mind, we count the number of threshold rules that must be considered at any one peripheral sensor when $A \geq D$. From this number, the total number of strategies that we must consider will follow easily.

Suppose that we insist that all D messages be used in any decision rule (see Proposition 5). Then it is easy to see that the number of different threshold rules (per peripheral sensor) that we must consider is the same as the number of solutions to

$$x_1 + ... + x_D = A,$$

where $x_1, ..., x_D$ are constrained to be positive integers. Call this number $C(D, A)$. Then, it is clear that it satisfies the recursive formula

$$\begin{aligned} C(D, A) &= \sum_{k=1}^{A-D+1} C(D-1, A-k) \\ &= \sum_{k=D-1}^{A-1} C(D-1, k), \end{aligned}$$

with the initialization $C(D = 1, A) = 1$. Thus, for $D = 2$, there are $A - 1$ possible decision rules per sensor; for $D = 3$, there are $1/2(A^2 - 3A + 2)$ decision rules per sensor, etc.

In general, then, we need to consider $C^N(D, A)$ different strategies at a cost of $O[(NA + D^N)(C^N(D, A))]$ arithmetic operations.

2.7.3 Conditionally Dependent, Binary Hypothesis

In the conditionally-dependent case, the computational complexity becomes much more formidable. In general, we can place no a priori structure on the form of optimal peripheral sensor rules. Thus, for each peripheral sensor rule, we must examine one rule from each of equivalent class.

Non-skipping rules

To help us systematically restrict our search to one rule from each of equivalent class, we introduce the following concept.

A decision rule is said to be a *non-skipping* decision rule iff the following holds:

$$\inf\{y : \gamma(y) = i\} \quad < \quad \inf\{y : \gamma(y) = j\}, \quad 1 \leq i \leq D, \quad j > i.$$

A strategy composed of non-skipping peripheral sensor rules is said to be a non-skipping strategy.

Thus, 1123123 is a non-skipping rule, while 1132132 is not.

We now can prove the following proposition.

Proposition 6 *With no loss of optimality, we can restrict attention to non-skipping strategies.*

Proof: We show that every decision rule is equivalent to some rule within the class of non-skipping rules. Thus, we can restrict attention to non-skipping rules (and hence, non-skipping strategies) with no loss of optimality.

To show that every rule is equivalent to some rule within the class of non-skipping rules, consider the arbitrary rule $\gamma(\cdot)$. Let $x_1 = \min\{y : \gamma(y) > 1\}$ (the minimum exists, since Y is discrete, with a finite number of possible values). If $\gamma(x) \neq 2$, then perform a permutation mapping on $\gamma()$, in which $\gamma(x_1) \rightarrow 2$, $2 \rightarrow \gamma(x_1)$, and everything else stays the same. Label the resulting decision rule $\gamma^1(\cdot)$.

Now, let $x_2 = \min\{y : \gamma^1(y) > 2\}$. If $\gamma^1(x_2) \neq 3$, then perform a permutation mapping on $\gamma^1(\cdot)$ in which $\gamma(x_2) \rightarrow 3$, $3 \rightarrow \gamma(x_2)$, and everything else stays the same. Label the resulting decision rule $\gamma^3(\cdot)$. Note that this mapping will not disrupt the locations of the decision regions for which the message is 1 or 2.

We continue this procedure until all D message values have been accounted for. Since D is finite, we will eventually arrive at a non-skipping rule. Since the rule at every intermediate step is equivalent to the starting rule, we conclude that the final rule is equivalent to the starting rule. **QED**.

Operation count

We can again develop a recursive equation for the number of decision rules per peripheral sensor that must be considered. Again, we insist that all D message values be used in any rule. Then

$$C(D, A) = \frac{1}{D!}\left[D^A - \sum_{k=1}^{D-1} \binom{D}{k} C(k, A)\right] \tag{2.21}$$

with the initialization $C(D = 1, A) = 1$. The term in brackets corresponds to the number of decision rules that use the maximum number of symbols (D). The division by $D!$ accounts for the number of decision rules in each equivalent class for rules that use all D symbols. From (2.21), it follows that for $D = 2$, there are $2^{A-1} - 1$ decision rules per sensor, for $D = 3$, there are

$$\frac{1}{2}\left[3^{A-1} - 2^A + 1\right]$$

decision rules, etc. When A is large, there is only a $(D!)^N$ savings of limiting the search to non-skipping strategies, as opposed to searching every possible strategy. However, when A is small, the savings can be greater.

Overall, the complexity is $O(C^N(D,A)(NA+D^N))$.

NP-Completeness

One might wonder whether we can somehow cleverly reduce the computation of finding an optimal strategy in the conditionally dependent case. Unfortunately, a main result of [TA85] would indicate that this possibility is very unlikely.

In [TA85], the authors consider the following problem:

> We are given finite sets Y_1, Y_2; a rational number K; a rational probability mass function p: $Y_1 \times Y_2 \rightarrow Q$; and a partition $\{A_0, A_1\}$ of $Y_1 \times Y_2$. Do there exist $\gamma_i : Y_i \rightarrow \{0,1\}$, $i = 1,2$ such that $J(\gamma_1,\gamma_2) \leq K$? Here,
>
> $$J(\gamma_1,\gamma_2) = \sum_{(y_1,y_2)\in A_0} p(y_1,y_2)\gamma_1(y_1)\gamma_2(y_2) + \sum_{(y_1,y_2)\in A_1} p(y_1,y_2)[1-\gamma_1(y_1)\gamma_2(y_2)].$$

This problem can be shown to be NP-complete.

In a sense, the above NP-complete problem is the simplest possible decentralized detection problem that has conditionally dependent observations. This simplicity is in the sense that a centralized detector for the problem would trivially have zero probability of error. Thus, all difficulties in finding an optimal strategy are solely the result of decentralizing the problem.

Any NP-completeness result really only refers to the worst case. Thus, there may well be structured subclasses of conditionally dependent problems for which the optimal strategy can be found efficiently. In general, though, the NP-completeness result indicates that research effort should be dedicated to search algorithms for *approximately* optimal strategies.

In the next chapter, we will examine some very specific subclasses, where optimal strategies can be found efficiently. Then, in the chapters that follow, we will consider heuristically-based search algorithms for approximately optimal strategies.

2.8 Neyman-Pearson Variant of Binary Hypothesis Testing

Consider now a variant of the binary hypothesis testing problem in which it is not reasonable to assign a priori probabilities to the two hypotheses. In that case, we solve

$$\begin{aligned} &\text{Maximize} && J^D(\gamma) \\ &\text{Subject to} && J^F(\gamma) = \alpha, \end{aligned}$$

where $J^D(\gamma) = \Pr[\gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_N)) = 2|H_2]$ (the probability of detection), and $J^F(\gamma) = \Pr[\gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_N)) = 2|H_1]$ (the probability of false alarm). This is known as the Neyman-Pearson variant of the binary hypothesis testing problem.

We now prove the following proposition, which will help us solve the Neyman-Pearson problem in a later chapter.

Proposition 7 *Suppose that deterministic strategy γ is optimal for the Bayesian problem when the a priori hypothesis probabilities are*

$$Pr(H_1), \quad Pr(H_2).$$

Then, γ is also optimal for the Neyman-Pearson problem corresponding to $\alpha = J^F(\gamma)$.

Proof: First, note that by Proposition 2.10, γ is optimal not just over Γ, but also over Γ^*. Now, suppose by way of contradiction, that

$$\exists\, \gamma^* \in \Gamma^* \text{ s.t. } J^F(\gamma^*) = \alpha, \quad J^D(\gamma^*) > J^D(\gamma).$$

If this were true, then we would have

$$\begin{aligned} \Pr(H_1)(1 - J^F(\gamma^*)) + \Pr(H_2)(1 - J^D(\gamma^*)) < \\ \Pr(H_1)(1 - J^F(\gamma)) + \Pr(H_2)(1 - J^D(\gamma)), \end{aligned}$$

which would mean that γ^* has a lower probability of error than γ for the Bayesian problem. Since this contradicts the optimality of γ over all Γ^* for the Bayesian problem, we conclude that no such γ^* exists. Thus, the optimality of γ for the Neyman-Pearson problem when $\alpha = J^F(\gamma)$ is established. **QED.**

Unfortunately, the converse of Proposition 7 is not true. That is, there is no guarantee that for some given α, there exist $\Pr(H_1)$, $\Pr(H_2)$ such that the corresponding Bayesian optimum strategy satisfies $J^F(\gamma) = \alpha$. Under stronger conditions, though, the converse of Proposition 7 is true, as is summarized in the following.

Proposition 8 *Suppose that the 2-dimensional set*

$$Q = \left\{(J^F(\gamma), J^D(\gamma)) | \gamma \in \Gamma\right\}$$

is convex. Then, for any given $\alpha, 0 \leq \alpha \leq 1$, there exists some choice of $Pr(H_1)$, $Pr(H_2)$ such that the corresponding Bayesian problem has a deterministic optimum strategy γ satisfying $J^F(\gamma) = \alpha$.

The following proof is very closely based on the proof of a related result in [T89]. We could just refer to that proof, but the details are just slightly different enough to justify inclusion of the proof here. So, for completeness sake, here is the proof.

Proof: Note that our optimization problem is equivalent to maximizing q_2 subject to $q_1 = \alpha$, with $(q_1, q_2) \in Q$. Let $q_2' \in [0, 1]$ be the maximum of q_2 subject to these constraints. Clearly, the point (α, q_2') lies at the upper-boundary of Q. Then,

because Q is convex, the supporting hyperplane theorem guarantees that there exist scalars c_1, c_2 such that (α, q_2') maximizes $c_1 q_1 + c_2 q_2$ over the set Q.

Now, note that the set Q is contained in the unit rectangle and the set includes the points $(0,0)$, $(1,1)$; hence, by simple geometry, we must have $c_1 \leq 0$, $c_2 > 0$. For this reason, the point (α, q_2') must maximize $-cq_1 + q_2$, where $c = -c_1/c_2$ (and so consequently $c \geq 0$). With no loss of generality, we can let $c = \Pr(H_1)/(1 - \Pr(H_1))$ for some choice of $\Pr(H_1)$. But using this last expression for c, we see that (α, q_2') also minimizes $\Pr(H_1)q_1 - \Pr(H_2)q_2$, which is exactly the probability of error in the Bayesian problem. Hence, we have shown by construction that there exist a priori probabilities, $\Pr(H_1)$, $\Pr(H_2)$ such that the corresponding Bayesian problem has an optimal strategy satisfying $J^F = \alpha$. **QED.**

Note that the proof of this proposition depends heavily on the convexity of Q. Unfortunately, an example in [R87] shows that the set Q is not necessarily convex, and so care must be taken in applying Proposition 8.

There are many aspects of the decentralized Neyman-Pearson theory that we have not touched. Excellent discussions of the subject can be found in [T89], [T89a]. Our proof of Proposition 7 is a simplified rearrangement of the proof in [T89a].

Chapter 3

Case Studies

3.1 Introduction

The last chapter began with a fairly general viewpoint toward the decentralized hypothesis testing problem. In that general setting, we derived the person-by-person optimality conditions; those general conditions did not provide very much insight into the structure of optimal strategies. We then focused on the binary hypothesis, conditionally independent observation problem. In that specialized setting, we noted that optimal strategies can always be found within the class of threshold strategies. We found no such powerful result for the conditionally dependent observation problem. Nor did we find any powerful results for the M-ary ($M > 2$) hypothesis testing problem.

In this chapter, we will further specialize our analysis. Through four case studies, we will show that for particular problems, we can find optimal strategies with aesthetically pleasing structure.

3.2 Case Study 1: Conditionally Independent Gaussian Problem

Consider the following signal model, wherein the observation at each peripheral sensor is a signal immersed in Gaussian noise:

$$\begin{aligned} \Pr[S = s_j] = \Pr(H_j) > 0, \qquad & j = 1,2, \quad s_2 > s_1 \\ Y_i = S + W_i, \qquad & i = 1,2, \\ W_i \sim \mathrm{N}(0,\sigma^2), \qquad & 0 < \sigma^2 < \infty, \end{aligned} \tag{3.1}$$

with the Gaussian random variables W_1, W_2 statistically independent. For $D = 2$, what is the optimal strategy?

Note that conditional on the true hypothesis, Y_1 and Y_2 are independent and identically distributed. As discussed in Chapter two, there is no guarantee (in general) that the optimal strategy will use the same decision rules at both peripheral sensors. However, for this particular problem, we have the following proposition:

Proposition 9 *Consider the detection problem described in (3.1). For any fixed s_1, s_2, σ, $Pr[H_j], (j = 1, 2)$, with*

$$s_2 > s_1, \quad \sigma > 0, \quad Pr[H_j] > 0, \ j = 1, 2,$$

there exists an optimal deterministic monotone threshold strategy that uses peripheral sensor rules

$$L(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} \exp\left[\frac{s_2 - s_1}{\sigma^2} T_i + \frac{s_1^2 - s_2^2}{2\sigma^2}\right] \iff y_i \underset{H_1}{\overset{H_2}{\gtrless}} T_i, \tag{3.2}$$

with

$$T_1 = T_2 = T, \tag{3.3}$$

for some finite T.

In the above proposition, we define

$$y_i \underset{H_1}{\overset{H_2}{\gtrless}} T_i \iff \gamma_i(y_i) = \begin{cases} 1 & y_i < T_i \\ 2 & y_i \geq T_i \end{cases} \tag{3.4}$$

We will use this notation frequently throughout the rest of the thesis.

The result in the proposition is not particularly surprising; many published numerical experiments have suggested its truth. The distinction here is that the optimality result is *analytically* demonstrated. We begin with a heuristic argument which motivates the result. Then we give the full argument, in detail.

3.2.1 Outline of Proof

The following is an outline of the proof of the above proposition. This outline will show the simplicity of the key components of the proof.

First, because the peripheral sensor observations are conditionally independent, it follows that Proposition 4 is valid. Thus, there exists an optimal deterministic monotone threshold strategy of the form (3.2). However, Proposition 4 does not guarantee that (3.3) is satisfied, and proving the validity of this condition is the primary challenge before us.

For any given fixed fusion rule, the peripheral sensor pbp-optimality conditions can be expressed as a pair of coupled equations that relate T_1, T_2. These equations have the form

$$T_1 = f(T_2), \quad T_2 = f(T_1), \tag{3.5}$$

where $f(\cdot)$ depends on the particular fusion rule.

It turns out that if the fusion rule is the OR rule,

$$\gamma_0(u_1, u_2) = \begin{cases} 1 & u_1 = u_2 = 1, \\ 2 & \text{otherwise}, \end{cases} \tag{3.6}$$

or the AND rule,

$$\gamma_0(u_1, u_2) = \begin{cases} 2 & u_1 = u_2 = 2, \\ 1 & \text{otherwise}, \end{cases} \tag{3.7}$$

then the corresponding $f(\cdot)$ has a very nice property. In particular, for either of these fusion rules, the corresponding $f(\cdot)$ satisfies

$$-1 < \frac{df(t)}{dt} < 0, \quad -\infty < t < \infty. \tag{3.8}$$

The significance of (3.8) will readily follow from two observations:

1. From (3.5), it follows that any pbp-optimal pair of thresholds T_1, T_2 satisfy

$$T_2 - T_1 = f(T_1) - f(T_2).$$

2. From (3.8) it follows that any pair of finite thresholds T_1, T_2 satisfy

$$f(T_1) - f(T_2) \leq T_2 - T_1,$$

with equality if and only if $T_1 = T_2$.

Combining points 1 and 2, we conclude that for AND and OR fusion rules, all corresponding pbp-optimal threshold strategies (3.2) have equal thresholds (i.e, $T_1 = T_2$).

It turns out that there always exists a globally optimal strategy that uses AND or OR fusion. Thus, since all globally optimal strategies are pbp-optimal, we conclude that there always exists a globally optimal threshold strategy (3.2) that has equal thresholds.

In the next section, we will tie all of these ideas together in detail. The trickiest part will be proving the validity of (3.8).

3.2.2 Proof

In this section, we give a detailed proof of Proposition 9. We break the proof into a number of steps.

Canonical form

Suppose that in the context of (3.1),

$$s_1 = k_1, \quad s_2 = k_2, \quad \sigma = \theta,$$

and that the optimal threshold strategy (3.2) uses thresholds

$$T_1 = t_1, \quad T_2 = t_2.$$

Then, it is easy to see that for the scaled parameter values

$$s_1 = 0, \quad s_2 = \frac{k_2 - k_1}{\theta}, \quad \sigma = 1,$$

the optimal threshold strategy (3.2) can use the scaled thresholds

$$T_1 = \frac{t_1 - k_1}{\theta}, \quad T_2 = \frac{t_2 - k_1}{\theta}.$$

Therefore, if Proposition 9 is true for

$$s_1 = 0, \quad s_2 > 0, \quad \sigma = 1, \tag{3.9}$$

then the proposition must also be true for arbitrary s_1, s_2, σ (with $s_2 > s_1, \quad \sigma > 0$).

We will prove the proposition for the less notationally burdensome case (3.9), and as this discussion shows, we can do this with no loss of generality.

Fusion rules: AND and OR

We prove the following lemma.

> **Lemma:** Consider the detection problem described in (3.1). With no loss of optimality, the fusion rule can be constrained to the class of OR (3.6) and AND (3.7) fusion rules.

Proof: We prove this by exhaustive enumeration of the other possible fusion rules.

There are only six possible deterministic fusion rules. They are

u_1	u_2	u_0
1	1	1
1	2	1
2	1	1
2	2	2

(i),

u_1	u_2	u_0
1	1	1
1	2	2
2	1	2
2	2	2

(ii),

u_1	u_2	u_0
1	1	1
1	2	1
2	1	2
2	2	2

(iii),

u_1	u_2	u_0
1	1	1
1	2	2
2	1	1
2	2	2

(iv),

plus the two "degenerate" rules

u_1	u_2	u_0
1	1	1
1	2	1
2	1	1
2	2	1

(v),

u_1	u_2	u_0
1	1	2
1	2	2
2	1	2
2	2	2.

(vi)

Rules (i) and (ii) are AND and OR fusion, respectively; we now systematically eliminate the need to ever use any of the other fusion rules.

First, consider rule (iii); suppose that it is an optimal rule. There is no loss in optimality in computing (iii) as $\gamma_0(u_1, u_2) = u_1$. When computed this way, it is clear that rule is independent of u_2. Thus, without loss of optimality, $\gamma_2(\cdot)$ can be changed to any arbitrary rule. Suppose that it is changed to the rule

$$\gamma_2(\cdot) = \gamma_1(\cdot).$$

Then, with this new $\gamma_2(\cdot)$, it is clear that

$$\Pr[U_1 = 1, U_2 = 2|H_j] = \Pr[U_1 = 2, U_2 = 1|H_j], \quad j = 1, 2.$$

Thus, MAP fusion satisfies

$$\gamma_0(U_1 = 1, U_2 = 2) = \gamma_0(U_1 = 2, U_2 = 1).$$

Hence, since MAP fusion is optimal, we can substitute fusion rule (i) or (ii) for fusion rule (iii), without loss of optimality. We conclude that we never have to use the fusion rule (iii).

A virtually identical argument (which we omit) shows that we never have to use fusion rule (iv).

Next, consider the degenerate fusion rule, (vi). Let γ be some strategy that uses (vi). Then, it is clear that

$$J(\gamma) = \Pr[H_1].$$

We show that γ is not optimal by constructing a strategy with lower probability of error.

To wit, consider the strategy, γ' in which

1. $\gamma_1'(\cdot)$ is

$$L(y_1) \underset{H_1}{\overset{H_2}{\gtrless}} \frac{\Pr[H_1]}{\Pr[H_2]}, \tag{3.10}$$

2. $\gamma_2'(\cdot)$ is arbitrary, and

3. $\gamma_0'(u_1, u_2) = u_1$.

Then,

$$\begin{aligned} J(\gamma') &= \Pr(H_1)\Pr[U_0 = 2|H_1] + \Pr(H_2)\Pr[U_0 = 1|H_2] \\ &= \Pr(H_1) + \Pr(H_2)\Pr[U_0 = 1|H_2] - \Pr(H_1)\Pr[U_0 = 1|H_1] \\ &= \Pr(H_1) + \int_{\gamma_1^{-1}(U_1=1)} \left[\Pr(H_2) f_{Y_1|H_2}(y_1|H_2) - \Pr(H_1) f_{Y_1|H_1}(y_1|H_1)\right] \, dy_1 \\ &< \Pr(H_1). \end{aligned}$$

The inequality in the last line is a consequence of two facts. First is the form of the rule in (3.10). Second is the positiveness of $\Pr[H_1]$, $\Pr[H_2]$, which we have assumed throughout this development. This inequality proves that the optimal fusion rule can never be (vi) when the prior probabilities are positive.

A virtually identical argument (which we omit) shows that we never have to use the fusion rule (v). Thus, we can now conclude that there is no loss in optimality in restricting attention to AND and OR fusion. The lemma is proved. **QED.**

Pbp-optimality condition for thresholds

In Section 3.2.1, we stated that for any fixed fusion rule, the peripheral sensor pbp-optimality conditions can be expressed as a pair of coupled equations that relate T_1, T_2; see (3.5). This section gives a derivation of $f(\cdot)$ for OR and AND fusion.

The key is to tailor the peripheral sensor pbp-optimality condition (see Proposition 3) to this particular problem. We expand (2.14) to

$$\begin{array}{l} f_{Y_i|H_2}(y_i|H_2)\Pr(H_2)[a_i(2,H_2) - a_i(1,H_2)] \begin{array}{c} H_2 \\ \leq \\ > \\ H_1 \end{array} \\ \qquad f_{Y_i|H_1}(y_i|H_1)\Pr(H_1)[a_1(1,H_1) - a_1(2,H_1)]. \end{array} \tag{3.11}$$

Then, using the identity (for $D = 2$),

$$\begin{aligned} a_i(d,H_j) &= \Pr[U_0 \neq j | U_i = d, H_j] \\ &= \Pr[U_0 = 3 - j | U_i = d, H_j], \quad j = 1,2, \end{aligned}$$

(3.11) becomes

$$\begin{array}{l} f_{Y_i|H_2}(y_i|H_2)\Pr(H_2)[\Pr(U_0 = 2|U_i = 2, H_2) - \Pr(U_0 = 2|U_i = 1, H_2)] \begin{array}{c} H_2 \\ \geq \\ < \\ H_1 \end{array} \\ \qquad f_{Y_i|H_1}(y_i|H_1)\Pr(H_1)[\Pr(U_0 = 2|U_i = 2, H_1) - \Pr(U_0 = 2|U_i = 1, H_1)]. \end{array} \tag{3.12}$$

It is readily verified that for OR fusion,

$$\Pr[U_0 = 2|U_i = 2, H_j] - \Pr[U_0 = 2|U_i = 1, H_j] = \Pr[U_{3-i} = 1|H_j], \tag{3.13}$$

while for AND fusion,

$$\Pr[U_0 = 2|U_i = 2, H_j] - \Pr[U_0 = 2|U_i = 1, H_j] = \Pr[U_{3-i} = 2|H_j]. \tag{3.14}$$

Hence, assuming that

$$\Pr[U_i = d|H_j] > 0, \quad d,i,j = 1,2, \tag{3.15}$$

it follows that (3.12) can be expressed as

$$L_i(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} \alpha_i,$$

where

$$\alpha_i = \frac{\Pr(H_1)}{\Pr(H_2)} \cdot \frac{\Pr[U_0 = 2|U_i = 2, H_1] - \Pr[U_0 = 2|U_i = 1, H_1]}{\Pr[U_0 = 2|U_i = 2, H_2] - \Pr[U_0 = 2|U_i = 1, H_2]}. \tag{3.16}$$

Or, alternatively, (3.12) can be expressed as

$$y_i \underset{H_1}{\overset{H_2}{\gtrless}} T_i, \quad i = 1, 2, \tag{3.17}$$

where

$$T_i = \frac{\sigma^2}{s_2 - s_1} \left[\frac{s_2^2 - s_1^2}{2\sigma^2} + \log \alpha_i \right]. \tag{3.18}$$

As a note, one can show that if $\Pr(H_1) > 0$, $\Pr(H_2) > 0$, then there always exists an optimal threshold strategy that satisfies (3.15). The details (which we omit) are similar to those in the discussion of why "degenerate" fusion rules are never needed.

In terms of the threshold rule in (3.17) and (3.18), our expression for α_i (3.16) can be simplified. For OR fusion,

$$\begin{aligned} \Pr[U_0 = 2|U_i = 2, H_j] - \Pr[U_0 = 2|U_i = 1, H_j] &= \Pr[U_{3-i} = 1|H_j] \\ &= \Pr[Y_{3-i} < T_{3-i}|H_j] \\ &= \Phi(T_{3-i} - s_j), \end{aligned} \tag{3.19}$$

for $i = 1, 2$, and $j = 1, 2$. The first line follows from (3.13); the second line follows from (3.17), and the third line follows from the probability distribution of Y_i (see (3.1)). Similarly, for AND fusion,

$$\begin{aligned} \Pr[U_0 = 2|U_i = 2, H_j] - \Pr[U_0 = 2|U_i = 1, H_j] &= \Pr[U_{3-i} = 2|H_j] \\ &= \Pr[Y_{3-i} \geq T_{3-i}|H_j] \\ &= \Phi(s_j - T_{3-i}), \end{aligned} \tag{3.20}$$

for $i = 1, 2$ and $j = 1, 2$.

The following lemma summarizes this discussion, and casts the results in a form that will be useful for what follows.

Lemma: Consider the detection problem described in (3.1) with $s_1 = 0, s_2 > 0, \sigma = 1$. With no loss in optimality, the peripheral sensor rules can be constrained to the form (3.17). If OR fusion is used, then

$$T_1 = f_{or}(T_2), \quad T_2 = f_{or}(T_1),$$

where

$$f_{or}(t)) = \frac{1}{s_2}\left\{\frac{s_2^2}{2} + \ln\left[\frac{\Pr(H_1)}{\Pr(H_2)}\right] + \ln\left[\frac{\Phi(t)}{\Phi(t-s_2)}\right]\right\}. \tag{3.21}$$

If AND fusion is used, then

$$T_1 = f_{and}(T_2), \quad T_2 = f_{and}(T_1),$$

where

$$f_{and}(t) = \frac{1}{s_2}\left\{\frac{s_2^2}{2} + \ln\left[\frac{\Pr(H_1)}{\Pr(H_2)}\right] + \ln\left[\frac{\Phi(-t)}{\Phi(s_2-t)}\right]\right\}. \tag{3.22}$$

Bounds on derivative of $f(\cdot)$

We prove the following lemma.

Lemma:

$$-1 < \frac{df(t)}{dt} < 0, \quad -\infty < t < \infty, \tag{3.23}$$

for both OR and AND fusion, where $f_{or}(\cdot)$ is defined in (3.21) and $f_{and}(\cdot)$ is defined in (3.22).

Proof: We readily compute

$$\frac{df_{or}(t)}{dt} = \frac{1}{s_2\sqrt{2\pi}}\left[g(t) - g(t-s_2)\right], \tag{3.24}$$

and

$$\frac{df_{and}(t)}{dt} = \frac{1}{s_2\sqrt{2\pi}}\left[g(s_2-t) - g(-t)\right],$$

where

$$g(t) = \frac{\exp(-t^2/2)}{\Phi(t)}. \tag{3.25}$$

Note that

$$\frac{df_{and}(u)}{du}|_{u=t} = \frac{df_{or}(u)}{du}|_{u=s_2-t}.$$

Hence, if (3.23) is valid for OR fusion, then it is also valid for AND fusion. Thus, we concentrate on characterizing the derivative of $f_{or}(\cdot)$.

Our approach is to show that

$$-\sqrt{2\pi} < \frac{dg(u)}{du} < 0, \tag{3.26}$$

where $g(\cdot)$ is defined in (3.25). This is useful because it implies that

$$-s_2\sqrt{2\pi} < g(t) - g(t - s_2) < 0,$$

which, in light of (3.24), implies the validity of (3.23).

To establish (3.26), first note that

$$\frac{dg(u)}{du} = -g(u)\left[u + \frac{1}{\sqrt{2\pi}}g(u)\right]. \tag{3.27}$$

Now, we bound this derivative by separately considering negative and nonnegative values of u.

For negative u, we exploit the bound [AS64]

$$\sqrt{\frac{2}{\pi}}\frac{\exp(-u^2/2)}{|u| + \sqrt{u^2+4}} < \Phi(u) \leq \sqrt{\frac{2}{\pi}}\frac{-\exp(-u^2/2)}{|u| + \sqrt{u^2+8/\pi}}, \qquad u \leq 0.$$

This bound implies that

$$\sqrt{\frac{\pi}{2}}\left(|u| + \sqrt{u^2+8/\pi}\right) \leq g(u) < \sqrt{\frac{\pi}{2}}\left(|u| + \sqrt{u^2+4}\right), \qquad u \leq 0, \tag{3.28}$$

and

$$\frac{1}{2}\left(u + \sqrt{u^2+8/\pi}\right) \leq u + \frac{1}{\sqrt{2\pi}}g(u) < \frac{1}{2}\left(u + \sqrt{u^2+4}\right), \qquad u \leq 0. \tag{3.29}$$

Thus, combining (3.27), (3.28) and (3.29), we obtain the surprisingly simple bound

$$-\sqrt{2\pi} < \frac{dg(u)}{du} \leq -\sqrt{\frac{8}{\pi}}, \quad u \leq 0.$$

Now we bound the derivative of $g(\cdot)$ for nonnegative u. This turns out to be much simpler than for negative u. Proceeding,

$$\begin{aligned} ug(u) &= \frac{u\exp(-u^2)}{\Phi(u)} \\ &< \frac{1\exp(-1)}{\Phi(0)} \\ &= 2\exp(-1), \qquad u \geq 0. \end{aligned}$$

Also,

$$\begin{aligned}\frac{1}{\sqrt{2\pi}}g^2(u) &\leq \frac{1}{\sqrt{2\pi}}\left(\frac{\exp(0)}{\Phi(0)}\right)^2 \\ &= \frac{4}{\sqrt{2\pi}}, \qquad u \geq 0.\end{aligned}$$

Combining these last two results, we conclude that

$$\begin{aligned}\frac{dg(u)}{du} &> -2\exp(-1) - \frac{4}{\sqrt{2\pi}} \\ &> -\sqrt{2\pi}, \qquad u \geq 0.\end{aligned}$$

Also, it is easy to see that

$$\frac{dg(u)}{du} < 0, \qquad u \geq 0.$$

In summary, we can combine all of the above results to see that for all u, (3.26) is valid. Thus, from the discussion immediately after (3.26), we conclude that (3.24) is valid. Thus, the lemma is proved. **QED**.

Final step

A review of the discussion in Section 3.2.1 will show that all of the components of the proof have now been established. Thus, the proposition is proved. **QED**.

3.2.3 Generalizations

The above analysis raises at least two questions. First, does Proposition 9 hold when there are more than two peripheral sensors? Second, does the proposition hold for a more general class of distributions for the peripheral sensor observations (i.e., for non-Gaussian distributions)? In this section, we examine both of these issues. The discussion has a twofold purpose. It acts both as a reformulation of the central ideas in this Case study, and also as a guideline for future research.

Generalization to arbitrary number of sensors

Consider the binary hypothesis detection problem in which the peripheral sensor observations $Y_1, ..., Y_N$ are independent and identically distributed, conditional on the true hypothesis, and in which $D = 2$.

Suppose that there exists an optimal strategy with the following properties:

1. The peripheral sensor rules are of the form

$$L(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} \alpha_i.$$

2. The inequality

$$\Pr[U_0 = 2|U_i = 2, H_j] > \Pr[U_0 = 2|U_i = 1, H_j], \quad j = 1, 2$$

holds for $1 \leq i \leq N$.

3. The optimal fusion rule is of the form

$$\gamma_0(K = k) = 2, \quad \text{iff} \quad k \geq k^*, \tag{3.30}$$

where k^* is an integer, $0 < k^* \leq N$, and where K is the random variable

$$K = \sum_{k=1}^{N} (U_k - 1). \tag{3.31}$$

This is known as a *k-out-of-N rule.*

As we now show, these three properties lead to the possibility of a conceptually straightforward extension to Proposition 9.

Proceeding, properties 1 and 2 allow us to express the pbp-optimality conditions by the set of N coupled equations

$$\alpha_i = f_i(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_N), \quad 1 \leq i \leq N, \tag{3.32}$$

where $f_i(\cdot)$ is implicitly defined by the right-hand side of (3.16). This set of equations is dependent on the fusion rule.

For k-out-of-N fusion rules, it is easy to see that (3.16) becomes

$$\alpha_i = \frac{\Pr(H_1)}{\Pr(H_2)} \frac{\Pr[K_i = k^* - 1|H_1]}{\Pr[K_i = k^* - 1|H_2]}, \tag{3.33}$$

where

$$K_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} (U_k - 1). \tag{3.34}$$

By symmetry, the right-hand side of (3.33) is independent of i, and hence, Property 3 allows us to express (3.32) as

$$\alpha_i = f(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_N), \quad 1 \leq i \leq N, \tag{3.35}$$

where $f(\cdot)$ is now implicitly defined by the right-hand side of (3.33). Note, also, that by symmetry,

$$f(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_N) = f(\beta_1, ..., \beta_{N-1}), \tag{3.36}$$

where $\beta_1, ..., \beta_{N-1}$ is any permutation of the values $\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, \alpha_N$.

This last symmetry result, renders it straightforward to show that a sufficient condition to guarantee that $\alpha_1, ..., \alpha_N$ are identical in an optimal strategy is that

$$-1 < \frac{\partial}{\partial \beta_1} f(\beta_1, ..., \beta_{N-1} < 1, \quad 0 < \beta_1, ..., \beta_{N-1} < \infty.) \tag{3.37}$$

Actually, this condition is sufficient to guarantee that $\alpha_1, ..., \alpha_N$ are identical in any pbp-optimal strategy.

To demonstrate this sufficiency, note that for any α_i and α_j $(i \neq j)$ in a pbp-optimal strategy, we must have (by definition)

$$\alpha_i - \alpha_j = f(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_N) - f(\alpha_1, ..., \alpha_{j-1}, \alpha_{j+1}, ..., \alpha_N). \tag{3.38}$$

But, also, by the sufficiency condition and by the symmetry expressed in (3.36),

$$|\alpha_i - \alpha_j| \geq |f(\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_N) - f(\alpha_1, ..., \alpha_{j-1}, \alpha_{j+1}, ..., \alpha_N)|, \tag{3.39}$$

with equality if and only if $\alpha_i = \alpha_j$. Thus, we conclude that when properties 1-3 hold, and also (3.37) holds, then all pbp-optimal strategies have

$$\alpha_1 = \cdots = \alpha_N.$$

This result is aesthetically pleasing, but is not practically useful. There are at least two difficulties. First, there is no known way to show that property 3 (i.e., optimality of k-out-N fusion rules) holds. Second, there is no way to analytically demonstrate that (3.37) holds for $N > 2$.

Generalization to non-Gaussian distributions

There is no apparent way to extricate the dependence of the proof of Proposition 9 from the details of the Gaussian distribution. It would seem that a completely different proof would be needed, and it is unknown whether such a proof exists.

Nevertheless, the given proof does provide an outline of one approach to a test whether a identical rule strategies are optimal in any given problem. For the two peripheral sensor case, the key is to construct (for each possible fusion rule) the coupling function, $f(\cdot)$, and to show that

$$-1 < \frac{d}{dt} f(t) < 1, \quad -\infty < t < \infty. \tag{3.40}$$

3.2.4 Conclusion

The key result of this section is Proposition 9. The result is new. The motivation for it was the nonlinear Gauss-Seidel algorithm (see next chapter).

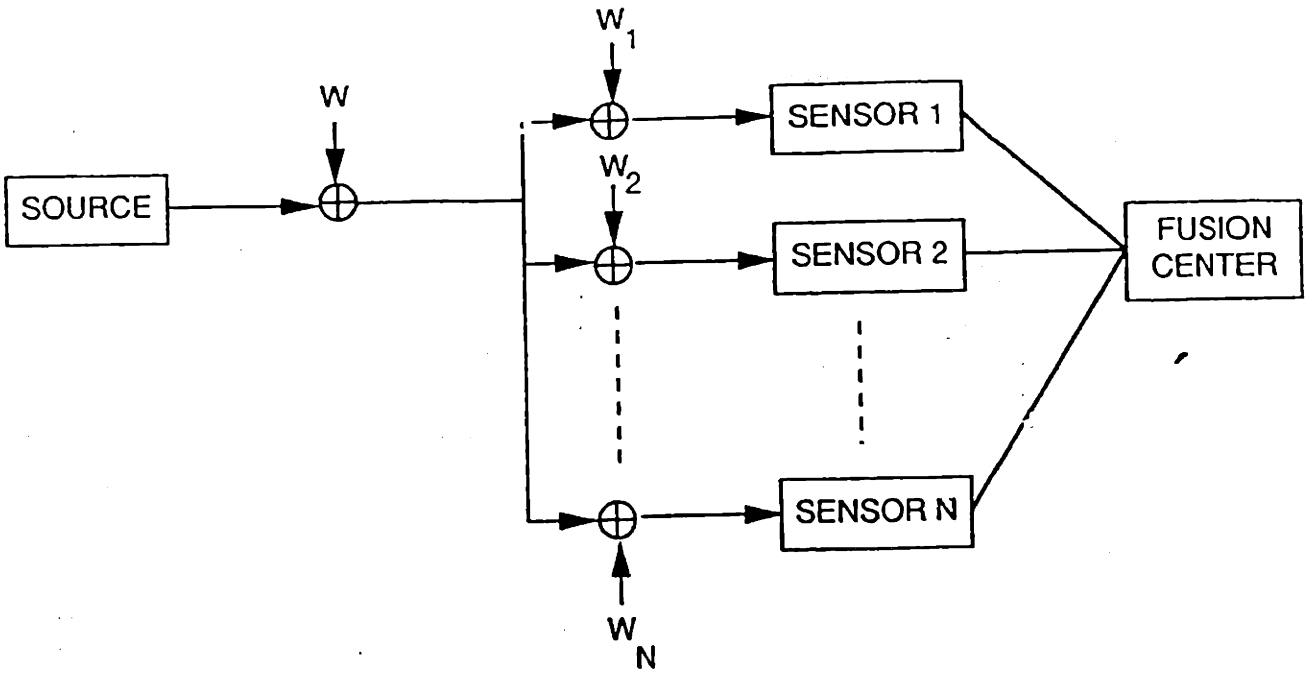

Figure 3.1: Observation model for Gaussian problem

3.3 Case Study 2: Conditionally-Dependent Gaussian Problem

Consider the following signal model, wherein the observation at each peripheral sensor is a signal immersed in Gaussian noise:

$$\begin{aligned} \Pr[S = j-1] = \Pr(H_j), &\qquad j = 1,2, \\ Y_i = S + W + W_i, &\qquad 1 \le i \le N, \\ W \sim \mathrm{N}(0,\rho), \quad W_i \sim \mathrm{N}(0,(1-\rho)), &\qquad 0 \le \rho < 1. \end{aligned} \tag{3.41}$$

with all of the Gaussian random variables $W, W_1, ..., W_N$ mutually independent. Note that conditional on the true value of the binary discrete random variable, S, the peripheral sensor observations, $Y_1, ..., Y_N$ are still correlated; they all share the common noise component, W. Also note that conditioned on either hypothesis, the correlation coefficient between Y_i, Y_j, $i \neq j$, is ρ. This observation model is pictorially represented in Figure 3.1.

When $D = 2$ (i.e., code alphabet size of two at each peripheral sensor), what is the optimal strategy?

3.3.1 Motivation

To motivate the following analysis, consider the above model for $\rho = 0$. In that case, the peripheral sensor observations become conditionally independent, and so from Proposition 4, there exists an optimal deterministic monotone threshold strategy, γ'. Now, suppose that ρ is infinitesimally small, but nonzero. Then, Proposition 4 no longer holds. Nevertheless, one could reasonably conjecture that as ρ smoothly increases from 0, there exists an optimal strategy that smoothly (in some sense, to be determined) moves away from γ'.

Along these lines, we argue as follows. We conjecture that there exists a pbp-optimal strategy in which the peripheral sensors all use the identical rule

$$L(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} \exp(T(\rho) - \frac{1}{2}) \iff y_i \underset{H_1}{\overset{H_2}{\gtrless}} T(\rho), \tag{3.42}$$

where $T(\rho)$ depends only on the correlation coefficient, ρ, and where the direct thresholding form of the test follows from the definition of the likelihood ratio. Next, for fixed ρ, we find a threshold, T such that the resulting strategy (threshold rule at all of the peripheral sensors and MAP rule at the fusion center) is indeed pbp-optimal. Finally, we note that for the case of $\rho = 0$, we have proved (for the case $N = 2$) that the corresponding identical threshold strategy is globally optimal. Also, for the $N > 2$, there is experimental evidence that the identical threshold strategy is globally optimal. Thus, it is likely that the strategy corresponding to $T(\rho)$, $\rho > 0$ is also globally optimal.

Somewhat disappointingly, we will not be able to obtain analytical results at every stage of this development. At the analytically insurmountable points of the development, we will turn to numerical experimentation. Then, at the end, we will summarize our findings.

3.3.2 Recasting of Pbp-Optimality Conditions for $D = 2$ Case

Our restriction that $D = 2$ allows us to rearrange the peripheral sensor pbp-optimality condition (2.14) to a more insightful form. Under a critical assumption that will be discussed below, the results in this section are general; they will be used for our Gaussian problem and also for the Erlang problem to be introduced in the second Case study.

We can expand (2.14) to

$$f_{Y_i|H_2}(y_i|H_2)\Pr(H_2)[a_i(2,H_2,y_i) - a_i(1,H_2,y_i)] \underset{H_1}{\overset{H_2}{\lessgtr}} f_{Y_i|H_1}(y_i|H_1)\Pr(H_1)[a_1(1,H_1,y_i) - a_1(2,H_1,y_i)]. \tag{3.43}$$

Then, using the identity (for $D = 2$),

$$\begin{aligned} a_i(d, H_j, y_i) &= \Pr[U_0 \neq j | U_i = d, y_i, H_j] \\ &= \Pr[U_0 = 3 - j | U_i = d, y_i, H_j], \quad (j = 1, 2), \end{aligned}$$

(3.43) becomes

$$f_{Y_i|H_2}(y_i|H_2)\Pr(H_2)[\Pr(U_0 = 2|U_i = 2, y_i, H_2) - \Pr(U_0 = 2|U_1 = 1, y_i, H_2)] \underset{H_1}{\overset{H_2}{\gtrless}}$$

$$f_{Y_i|H_1}(y_i|H_1)\Pr(H_1)[\Pr(U_0 = 2|U_i = 2, y_i, H_1) - \Pr(U_0 = 2|U_i = 1, y_i, H_1)] \quad (3.44)$$

Now, under the critical assumption that

$$\Pr[U_0 = 2|U_i = 2, y_i, H_2] > \Pr[U_0 = 2|U_i = 1, y_i, H_2], \quad 1 \leq i \leq N, \ \forall y_i \quad (3.45)$$

we can rearrange (3.44) to

$$L_i(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} \alpha_i(y_i), \quad (3.46)$$

where

$$\alpha_i(y_i) = \frac{\Pr(H_1)}{\Pr(H_2)} \cdot \frac{\Pr[U_0 = 2|U_i = 2, y_i, H_1] - \Pr[U_0 = 2|U_i = 1, y_i, H_1]}{\Pr[U_0 = 2|U_i = 2, y_i, H_2] - \Pr[U_0 = 2|U_i = 1, y_i, H_2]}. \quad (3.47)$$

Disregarding for a moment our perhaps tenuous assumption (3.45), our result (3.46) is quite interesting. We have shown that in the $D = 2$ case, we can express the pbp-optimal sensor rule as a data-dependent likelihood ratio test. In the next subsection, we exploit this result in our Gaussian problem. For that specific problem, we show that for a properly-selected threshold, T, the strategy described by (3.42) is pbp-optimal.

3.3.3 Finding Threshold for Pbp-Optimal Strategy

Consider the strategy described by (3.42). The following steps are directed toward showing that for special values of T in (3.42), the corresponding strategy is pbp-optimal. Unfortunately, it does not seem feasible to carry out the full demonstration analytically. Thus, the following argument is really only a detailed motivation of the idea. The full validity of the argument for special cases will be carried out numerically in Section 3.3.6.

1. If the rule (3.42) is used (regardless of its optimality), then by symmetry, the number of 2-messages that the fusion center receives (denoted by k),

$$K = \sum_{k=1}^{N} (U_k - 1) \quad (3.48)$$

is a sufficient statistic for the fusion rule.

Also, by symmetry, the statistic

$$K_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} (U_k - 1) \tag{3.49}$$

is meaningful; we will find this statistic useful in the what follows. In words, K_i is the total number of 2-messages that the fusion center receives from all sensors excluding sensor i.

2. When the rule (3.42) is used, the fusion rule becomes monotonic with respect to the k-statistic introduced in point 1. In other words, the pbp-optimal fusion rule becomes

$$\gamma_0(k) = 2 \quad \text{iff} \quad k \geq k^*,$$

for some integer k^*, $0 \leq k^* \leq N$; that is, it becomes a k-out-of-N rule.

To see this result, we first express our original fusion pbp-optimality condition (2.11) as a likelihood ratio test,

$$\gamma_0(\cdot): \quad \frac{\Pr[u_1, ..., u_N | H_2]}{\Pr[u_1, ..., u_N | H_1]} \underset{H_1}{\overset{H_2}{\gtrless}} \frac{\Pr(H_1)}{\Pr(H_2)}. \tag{3.50}$$

Since for our problem, K is a sufficient statistic for the fusion rule, it follows that (3.50) can be simplified to

$$\gamma_0(k): \quad \frac{\Pr[K = k | H_2]}{\Pr[K = k | H_1]} \underset{H_1}{\overset{H_2}{\gtrless}} \frac{\Pr(H_1)}{\Pr(H_2)}. \tag{3.51}$$

Hence, by showing that the left-hand side of (3.51) is nondecreasing with k, we will establish that the k-out-of-N rule is pbp-optimal. We defer the proof until Section 3.3.4; the details are tedious, and we do not wish to interrupt the flow of the present argument.

For now, we treat the value of k^* as a parameter. At the end of this section, we will show how to select the optimal k^*.

3. Again, suppose that the rule (3.42) is used. Then, we can use the fusion monotonicity result in point 2 to show that (3.45) holds. In particular, k-out-of-N fusion means that

$$\Pr[U_0 = 2 | U_i = d, y_i, H_2] = \sum_{k=k^*+1-d}^{N-1} \Pr[K_i = k | y_i, H_2], \quad d = 1, 2, \tag{3.52}$$

and hence, for $k^* > 0$, it follows that

$$\begin{aligned} &\Pr[U_0 = 2|U_i = 2, y_i, H_2] - \Pr[U_0 = 2|U_i = 1, y_i, H_2] \\ &\quad = \Pr[K_i = k^* - 1|y_i, H_2] \\ &\quad > 0, \end{aligned} \tag{3.53}$$

thus establishing (3.45) for the case of $k^* > 0$. This result will allow us, in the next step, to use the special form of the pbp-optimality condition expressed by (3.46).

4. Suppose that peripheral sensors $2, ..., N$ are using the rule (3.42), that the fusion center is using a k-out-of-N rule, and that we want to find the corresponding pbp-optimal rule for sensor 1, $\gamma_1(\cdot)$. By the result of point 3, we can use (3.46). We denote this form of the decision rule with the special notation

$$L_i(y_1) \underset{H_1}{\overset{H_2}{\gtrless}} \alpha_1'(y_1, k^*, T),$$

thus explicitly showing the dependence of the data-dependent threshold on k^* (fusion rule) and on T (other peripheral sensor rules).

5. Now comes the most critical step. We attempt to find a threshold T, such that the rule in point 4 satisfies the equivalence

$$L_i(y_i) \geq \alpha_i'(y_i, k^*, T) \iff y_i \geq T. \tag{3.54}$$

If such a threshold exists, then the pbp-optimal rule for sensor 1 is to use a direct thresholding rule with threshold T. But then, by symmetry, threshold rule T is pbp-optimal for *all* of the peripheral sensors, and so we have an identical-rule pbp-optimal strategy.

We need to prove that there exists a T that satisfies (3.54). Toward that end, we first simplify our general expression for $\alpha_i'(\cdot)$ (see Equation 3.47); by the result in point 3, it follows that

$$\alpha_i'(y_i, k^*, T) = \frac{\Pr[H_1]}{\Pr[H_2]} \frac{\Pr[K_i = k^* - 1|y_i, H_1, T]}{\Pr[K_i = k^* - 1|y_i, H_2, T]}. \tag{3.55}$$

For our Gaussian problem,

$$\begin{aligned} \Pr[K_i = k - 1|y_i, H_j, T] \;=\; & \\ & \binom{N-1}{k-1} \int_{-\infty}^{\infty} \frac{\exp\{-[w - (s_j + \rho(y - s_j))]^2/2\rho(1-\rho)\}}{\sqrt{2\pi\rho(1-\rho)}} \cdot \\ & \left[\Phi\left(\frac{w - T}{\sqrt{1-\rho}}\right)\right]^{k-1} \left[\Phi\left(\frac{T - w}{\sqrt{1-\rho}}\right)\right]^{N-k} dw, \end{aligned} \tag{3.56}$$

with $s_j = j - 1, \quad j = 1, 2$. The messy details of establishing this identity are relegated to Section 3.3.4

The expression in (3.56) is not at all pleasant, and as a result, there is no apparent way to prove that (3.54) has a solution. It is somewhat helpful to note that the only values of T that can possibly satisfy (3.54) are solutions to

$$L_i(y_i) = \alpha_i'(y_i, k^*, T)|_{T=y_i}. \tag{3.57}$$

Unfortunately, in general, proving existence of a T that satisfies (3.57) is also intractable. Thus, we content ourselves with an asymptotic analysis and with numerical experimentation.

First, we show that for a very special symmetric case, (3.57) has a solution that can be seen by inspection. We then use asymptotic techniques to show that as $\rho \to 0$, this threshold solution to (3.57) also behaves correctly in (3.54) as $|y_i| \to \infty$.

Finally, we pursue the solution to (3.57) and to (3.54) numerically. For the cases that we consider, we will see that a pbp-optimal identical threshold strategy exists, and can be found routinely.

As a final note, the optimal value of k^* is not known apriori. Hence, for each possible value of k^*, we solve for the optimal T (assuming that it exists), and then we compute

$$J(\gamma) = \Pr[H_2] \sum_{k=0}^{k^*-1} \Pr[K = k|H_2, T] \; + \; \Pr[H_1] \sum_{k=k^*}^{N} \Pr[K = k|H_1, T]$$

The value of k^* that minimizes this expression is the optimal value of k^*.

In the next section is a proof that the fusion rule is monotonic. Then, the development outlined above begins in Section 3.3.5.

3.3.4 Mathematical Details

In this section, we fill in two of the missing details from the last section. First, we prove that if the peripheral sensors all use the decision rule (3.42), then the fusion rule is monotonic. Second, we prove the identity (3.56).

Proof that fusion rule is monotonic

The following proposition establishes the monotonicity of the fusion rule.

> **Proposition 10** *Suppose that for some threshold T, the rule in (3.42) is used at all of the peripheral sensors. Then, regardless of the value of T, the ratio*
>
> $$\frac{Pr(K = k|H_2)}{Pr(K = k|H_1)}, \tag{3.58}$$
>
> *is nondecreasing. In this expression, K is defined in (3.48).*

The monotonicity of (3.58) can be expressed explicitly by

$$\frac{\int_{-\infty}^{\infty} e^{-(n-s_1)^2/2\epsilon^2} \left[1-\Phi\left(\frac{T-n}{\theta}\right)\right]^k \left[\Phi\left(\frac{T-n}{\theta}\right)\right]^{N-k} dn}{\int_{-\infty}^{\infty} e^{-(n-s_0)^2/2\epsilon^2} \left[1-\Phi\left(\frac{T-n}{\theta}\right)\right]^k \left[\Phi\left(\frac{T-n}{\theta}\right)\right]^{N-k} dn} \leq$$
$$\frac{\int_{-\infty}^{\infty} e^{-(n-s_1)^2/2\epsilon^2} \left[1-\Phi\left(\frac{T-n}{\theta}\right)\right]^{k+1} \left[\Phi\left(\frac{T-n}{\theta}\right)\right]^{N-k-1} dn}{\int_{-\infty}^{\infty} e^{-(n-s_0)^2/2\epsilon^2} \left[1-\Phi\left(\frac{T-n}{\theta}\right)\right]^{k+1} \left[\Phi\left(\frac{T-n}{\theta}\right)\right]^{N-k-1} dn}, \tag{3.59}$$

which must hold for $0 \leq k \leq N-1$, and $\forall T$. In the context of (3.41), $\epsilon = \sqrt{\rho}$, and $\theta = \sqrt{1-\rho}$.

The following lemma is needed for the proof.

Lemma Let $g(x)$ be any smooth, monotonically decreasing function. Let f(x) be any nonnegative, smooth function. Then, the ratio

$$\frac{\int_0^1 x\, f(x)\, e^{c\, g(x)}\, dx}{\int_0^1 f(x)\, e^{c\, g(x)}\, dx} \tag{3.60}$$

monotonically decreases as the scalar c increases, provided that the ratio is finite.

Proof: For $c_2 \geq c_1$ and $x_2 \geq x_1 \geq 0$, it follows that

$$\begin{aligned} c_1[g(x_1)-g(x_2)] &\leq c_2[g(x_1)-g(x_2)] \implies \\ x_1 e^{c_1 g(x_2)+c_2 g(x_1)} + x_2 e^{c_1 g(x_1)+c_2 g(x_2)} &\leq x_1 e^{c_1 g(x_1)+c_2 g(x_2)} + x_2 e^{c_1 g(x_2)+c_2 g(x_1)} \end{aligned} \tag{3.61}$$

where we have used the monotonicity of $g(x)$. From (3.61) it follows that

$$\int_0^1 \int_{x_1}^1 f(x_1)\, f(x_2)\, [x_1\, e^{c_1 g(x_2)+c_2 g(x_1)} + x_2\, e^{c_1 g(x_1)+c_2 g(x_2)}]\, dx_2\, dx_1 \leq$$
$$\int_0^1 \int_{x_1}^1 f(x_1)\, f(x_2)\, [x_1\, e^{c_1 g(x_1)+c_2 g(x_2)} + x_2\, e^{c_1 g(x_2)+c_2 g(x_1)}]\, dx_2\, dx_1,$$

for the nonnegative $f(x)$ in the lemma statement. But, using the factorization identity

$$\int_0^1 \int_{x_1}^1 a(x_1)\, b(x_2) + a(x_2)\, b(x_1)\, dx_1\, dx_2 = \left[\int_0^1 a(x)\, dx\right] \left[\int_0^1 b(x)\, dx\right],$$

with

$$\begin{aligned} a(x) &= x\, f(x)\, e^{c_2 g(x)}, \\ b(x) &= f(x)\, e^{c_1 g(x)}, \end{aligned}$$

it follows that

$$\left[\int_0^1 x\, f(x)\, e^{c_2 g(x)}\, dx\right] \left[\int_0^1 f(x)\, e^{c_1 g(x)}\, dx\right] \leq$$
$$\left[\int_0^1 x\, f(x)\, e^{c_1 g(x)}\, dx\right] \left[\int_0^1 f(x)\, e^{c_2 g(x)}\, dx\right],$$

or,

$$\frac{\int_0^1 x\ f(x)\ e^{c_2 g(x)}\ dx}{\int_0^1 f(x)\ e^{c_2 g(x)}\ dx} \leq \frac{\int_0^1 x\ f(x)\ e^{c_1 g(x)}\ dx}{\int_0^1 f(x)\ e^{c_1 g(x)}\ dx},$$

which is the desired result. **QED.**

The rest of the proof is a tedious application of algebra and elementary calculus. Proceeding, we let

$$s_0' = \frac{s_0 - T}{\theta}, \quad s_1' = \frac{s_1 - T}{\theta}, \quad \sigma = \frac{\epsilon}{\theta}, \tag{3.62}$$

where $s_0, s_1, \epsilon, \theta, T$ all defined above. Also let

$$c_0 = \frac{s_0'}{\sigma^2}, \quad c_1 = \frac{s_1'}{\sigma^2}.$$

With the identification

$$f(x) = \exp\{[\Phi^{-1}(x)]^2[\frac{1}{2} - \frac{1}{2\sigma^2}]\}\ (1-x)^k\ x^{N-k-1},$$

(a positive function over the $[0,1]$ interval), and

$$g(x) = -\Phi^{-1}(x)$$

(a strictly decreasing function), and with the change of variables $m = \Phi^{-1}(x)$, the inequality (3.60) in the lemma becomes

$$\frac{\int_{-\infty}^{\infty} e^{-(m+s_0')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k-1}\ \Phi(m)\ dm}{\int_{-\infty}^{\infty} e^{-(m+s_0')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k-1}\ dm} \geq \frac{\int_{-\infty}^{\infty} e^{-(m+s_1')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k-1}\ \Phi(m)\ dm}{\int_{-\infty}^{\infty} e^{-(m+s_1')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k-1}\ dm}. \tag{3.63}$$

Taking the reciprocals of both sides, reversing the inequality, and subtracting unity from both sides, (3.63) becomes

$$\frac{\int_{-\infty}^{\infty} e^{-(m+s_0')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k-1}\ dm}{\int_{-\infty}^{\infty} e^{-(m+s_0')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k}\ dm} - 1 \geq \frac{\int_{-\infty}^{\infty} e^{-(m+s_1')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k-1}\ dm}{\int_{-\infty}^{\infty} e^{-(m+s_1')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m))]^{N-k}\ dm} - 1. \tag{3.64}$$

With slight algebraic rearrangement, (3.64) becomes

$$\frac{\int_{-\infty}^{\infty} e^{-(m+s_1')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k}\ dm}{\int_{-\infty}^{\infty} e^{-(m+s_0')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k}\ dm} \geq \frac{\int_{-\infty}^{\infty} e^{-(m+s_1')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m)]^{N-k}\ \left[\frac{1}{\Phi(m)} - 1\right]\ dm}{\int_{-\infty}^{\infty} e^{-(m+s_0')^2/2\sigma^2}\ [\Phi(-m)]^k\ [\Phi(m))]^{N-k}\ \left[\frac{1}{\Phi(m)} - 1\right]\ dm}. \tag{3.65}$$

But substituting (3.62) into (3.65) yields (3.59), which is the desired result. **QED.**

Proof of Equation 3.56

We prove that if all of the peripheral sensors use the decision rule in (3.42), then the identity (3.56) is valid.

We will find it most convenient to establish (3.56) for the special case of $i = N$. However, note that by symmetry, $\Pr[K_i = k-1|Y_i = y, H_j, T]$ is independent of i, and so once the result is established for $i = N$, the result is also established for all i.

By conditioning on the true value of the common noise component, W (see (3.41)), we obtain

$$\Pr[K_N = k-1|y_N, H_j, T] = \int_{-\infty}^{\infty} f_{W|Y_N,H_j,T}(w|y_N, H_j, T) \Pr[K_N = k-1|w, y_N, H_j, T]\, dw. \quad (3.66)$$

In this expression,

$$f_{W|Y_i,H_j,T}(w|y_i, H_j, T) \sim N(\rho(y - s_j), \rho(1-\rho)). \quad (3.67)$$

Also,

$$\begin{aligned}
&\Pr[K_N = k-1|w, H_j, T] \\
&= \binom{N-1}{k-1} \Pr[U_1 = 2, ..., U_{k-1} = 2, U_k = 1, ..., U_{N-1} = 1|w, H_j, T] \\
&= \binom{N-1}{k-1} \left(\prod_{i=1}^{k-1} \Pr[U_i = 2|w, H_j, T]\right) \left(\prod_{i=k}^{N-1} \Pr[U_i = 1|w, H_j, T]\right) \\
&= \binom{N-1}{k-1} (\Pr[U_1 = 2|w, H_j, T])^{k-1} (\Pr[U_1 = 1|w, H_j, T])^{N-k} \\
&= \binom{N-1}{k-1} (\Pr[W_1 \geq T - s - w])^{k-1} (\Pr[W_1 < T - s - w])^{N-k} \\
&= \binom{N-1}{k-1} \left[\Phi\left(\frac{w + s - T}{\sqrt{1-\rho}}\right)\right]^{k-1} \left[\Phi\left(\frac{T - s - w}{\sqrt{1-\rho}}\right)\right]^{N-k}. \quad (3.68)
\end{aligned}$$

The second line follows by symmetry; all of the possible permutations of $U_1, ..., U_{N-1}$ that yield $K_N = k-1$ are equally likely. The third line follows from the independence of $U_1, ..., U_N$, conditional on W. The fourth line follows from symmetry, and the fifth line follows from the form of the decision rule (3.42) used at the peripheral sensors. Finally, the last line follows from the particular form of the probability distributions, as described in (3.41).

Finally, substituting (3.67) and (3.68) into (3.66) establishes the desired result. **QED**.

3.3.5 Asymptotic Analysis

Ideally, we would like to analytically show that for any ρ, $0 \leq \rho < 1$, (3.57) has a solution, T, such that (3.54) is satisfied. Unfortunately, there is no apparent way to

carry out that full analysis; in lieu of it, we offer the following asymptotic analysis, for the case of $\Pr(H_1) = \Pr(H_2) = 1/2$.

In general, our expression for $\alpha_i'(\cdot)$ in (3.55) must be evaluated numerically. However, note that for N odd, $k = \left\lceil \frac{N}{2} \right\rceil$, and $y = 1/2$,

$$\begin{aligned} & \Pr[K_i = k-1 | H_1, y_i, T]_{T=y=1/2} = \Pr[K_i = k-1 | H_2, y_i, T]_{T=y=1/2} \\ \Longrightarrow \quad & \alpha_i'(y_i, k, T)|_{T=y=1/2} = 1. \end{aligned} \tag{3.69}$$

Since $L(y_i)|_{y=1/2} = 1$, we conclude that $T = y = 1/2$ is a solution to (3.57).

Now, we want to determine if $T = 1/2$ also satisfies (3.54). The full behavior of $\alpha_i'(\cdot)$ can only be handled numerically, but as $\rho \to 0$, we can show that (3.54) is satisfied for $y \to -\infty$, and $y \to \infty$.

The Gaussian PDF, in the integrand of (3.56), becomes an impulse as $\rho \to 0$; formally, one can show [BO78] that

$$\Pr[K_i = k-1 | H_j, y_i, T] \sim \left(\begin{array}{c} N-1 \\ k-1 \end{array} \right) [\Phi(s_j(y_i))]^{k-1} [\Phi(-s_j(y_i))]^{N-k}, \quad \rho \to 0,$$

and so

$$\alpha_i'[k, y_i, T] \sim \frac{[\Phi(s_1(y))]^{k-1} [\Phi(-s_1(y))]^{N-k}}{[\Phi(s_2(y))]^{k-1} [\Phi(-s_2(y))]^{N-k}}, \quad \rho \to 0,$$

where

$$s_j(y) = \frac{1}{\sqrt{1-\rho}} [s_j + \rho(y - s_j) - T], \quad j = 1, 2.$$

Then, since [BO78]

$$\Phi(x) \sim \left\{ \begin{array}{ll} \frac{1}{\sqrt{2\pi}} \frac{1}{|x|} \exp(-x^2/2), & x \to -\infty, \\ 1 - \frac{1}{\sqrt{2\pi}} \frac{1}{x} \exp(-x^2/2), & x \to \infty, \end{array} \right. ,$$

it follows that

$$\alpha_i' \left[k = \left\lceil \frac{N}{2} \right\rceil, y_i, T = 1/2 \right] \sim \exp(k\rho y_i), \quad y \to -\infty, \quad \text{or} \quad y \to \infty,$$

and so for $k\rho < 2$,

$$\begin{aligned} \alpha_i' \left[k = \left\lceil \frac{N}{2} \right\rceil, y_i, T = 1/2 \right] & < L(y_i), \quad y \to -\infty, \\ \alpha_i' \left[k = \left\lceil \frac{N}{2} \right\rceil, y_i, T = 1/2, \right] & > L(y_i), \quad y \to \infty. \end{aligned} \tag{3.70}$$

Equation 3.70 shows that if the threshold computed in (3.69) is used at all of the peripheral sensors, then the pbp-optimality condition (3.54) is satisfied for $\rho \to 0, |y| \to \infty$.

From a practical point of view, this analysis is not very useful. We have not been able to characterize the behavior of $\alpha_i'(y)$ for either moderate values of ρ or for moderate values of y. The numerical approach that we pursue in the next section will be more fruitful.

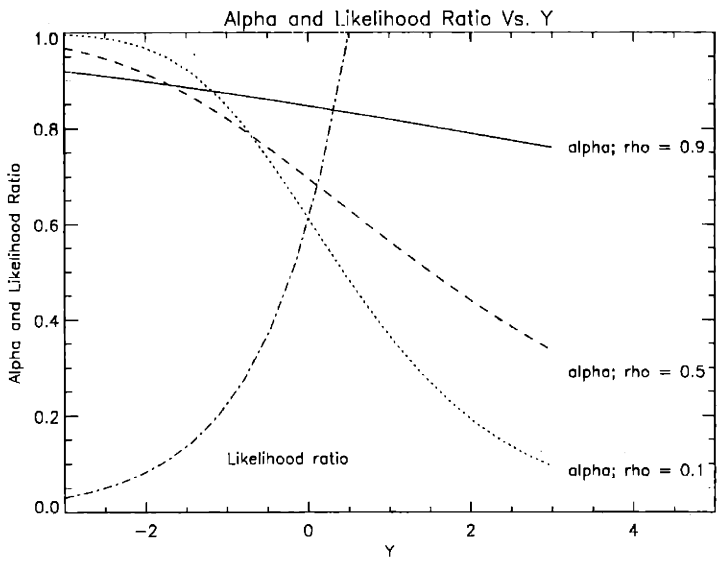

Figure 3.2: $\alpha_i'(y_i, k^*, T)|_{T=y_i}$ as a function of y for various values of ρ.

3.3.6 Numerical Results

In Section 3.3.3, we established that the key to finding a pbp-optimal, identical threshold strategy is to solve (3.54) for T. As we have shown through asymptotic analysis, this is intractable analytically. Here, we *numerically* search for a solution for the specific case of $N = 2$, $k^* = 2$, $\Pr(H_1) = \Pr(H_2) = 1/2$. As it will turn out, a numerical solution to (3.54) is straightforward (and, more importantly, exists), thus lending additional credence to the idea of a pbp-optimal identical threshold rule strategy.

We break the solution of (3.54) into two steps. First, we solve (3.57), and second we show that our solutions to (3.57) are also solutions to (3.54).

Figure 3.2 shows a numerical solution to (3.57) for three different values of ρ ($\rho = 0.1, 0.5, 0.9$). Note that for each of these three ρ values, the (decreasing) $\alpha_i'(y)$ curve intersects the (increasing) $L_i(y)$ curve only once. Thus, for these cases, (3.57) has a unique solution.

Note that a threshold strategy satisfying (3.57) is not guaranteed to also satisfy (3.54). To confirm satisfaction of the latter, we plot both sides of (3.54) in Figure 3.3 for the cases $\rho = 0.1, 0.9$. Note that for the limited range of y plotted, both sides of (3.54) are monotonically increasing, but $L(y)$ increases more quickly of the two. Thus, for these cases, (3.54) has a unique solution that can be found by solving (3.57).

In Table 3.3.6, we show the dependence of T^* on ρ. Note that as $\rho \to 1$, $T^* \to 1/2$. This is an intuitively pleasing result. In the limit of $\rho = 1$, the observations at the two peripheral sensors become identical; since this is only a binary hypothesis testing problem, the fusion center need only use one peripheral sensor to make its final decision. But then, the final decision is essentially made at the active peripheral sensor, which itself can perform a classical centralized likelihood ratio

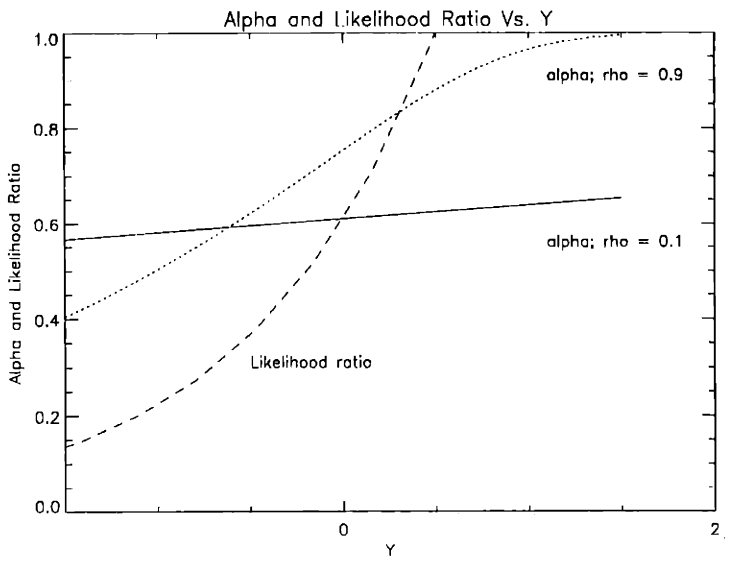

Figure 3.3: $\alpha_i'(y_i, k^*, T^*)$ as a function of y for various values of ρ.

correlation coefficient	optimum threshold
0.0	-0.0135
0.1	0.00615
0.5	0.116
0.9	0.323
0.99	0.444
0.999	0.482

Figure 3.4: Relation between correlation coefficient and optimum threshold in threshold strategy.

test. The optimal threshold for that centralized test is $T = 1/2$ (since the two hypotheses are equally likely), in agreement with the trend in Table 3.3.6.

3.3.7 Conclusion

The analysis of this section is new. It was motivated by the method used in [RN87] to find pbp-optimal strategies (with identical thresholds at all peripheral sensors) for the conditionally *indpendent* problem.

The method of analysis is not limited to just the Gaussian problem. It is intuitively plausible that for any problem whose observations can be modeled as in Figure 3.1, an identical threshold strategy could be optimal.

The following summarizes the sufficient conditions for optimality of an identical threshold strategy.

1. When the rule

$$L_i(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} T \tag{3.71}$$

is used at the peripheral sensors, the inequality in (3.45) is satisfied.

2. When condition 1, holds, we can use the special form of the peripheral sensor pbp-optimality condition, as expressed in (3.46) and (3.47). Then, if there exists a threshold T such that

$$L_i(y_i) \geq \alpha_i(y_i) \iff L_i(y_i) \geq T, \quad 1 \leq i \leq N, \tag{3.72}$$

then the threshold strategy in (3.71) (with the corresponding MAP rule at the fusion center) is pbp-optimal.

For our Gaussian problem the symmetry of the observation model (Figure?) made it fairly easy to establish condition 1. Condition 2 was analytically intractable, but for specific cases the condition could be numerically established.

In the next section, we will analyze another problem in which the peripheral sensor observations can be modeled by Figure 3.1. We will find that the analysis method used in this section is again useful.

3.4 Case Study 3: Conditionally Dependent Erlang Problem

Consider the following signal model wherein the observation at each peripheral sensor is a sum of two exponential random variables. The λ-parameter is always the same for both summands, but is different by hypothesis. We have

$$\begin{aligned} \Pr[\Lambda = \lambda_j] = \Pr(H_j), &\qquad j = 1,2, \quad \lambda_1 > \lambda_2, \\ Y_i = W + W_i, &\qquad 1 \leq i \leq n_s, \\ f_{W|H_j}(w|H_j) = \lambda_j \exp(-\lambda_j w), &\qquad w \geq 0, \end{aligned}$$

with all of the exponential random variables $W, W_1, ..., W_N$ mutually independent. When $D = 2$, what is the optimal strategy?

For reference, note that

$$\begin{aligned} f_{Y_i|H_j}(y_i|H_j) &= \lambda_j^2 y_i \exp(-\lambda_j y_i), \\ L(y_i) &= \left(\frac{\lambda_2}{\lambda_1}\right)^2 \exp[(\lambda_1 - \lambda_2)y_i]. \end{aligned} \tag{3.73}$$

This observation model is pictorially represented in Figure 3.1. Note that this same figure was used for the Gaussian problem observation model (in the last section). Thus, although the probability distribution of observations is certainly different between the Gaussian problem and this problem, the two observation models do have a common underlying structure.

3.4.1 Motivation

In the sense of Figure 3.1, this problem has the same structure as the Gaussian problem. Since threshold strategies were found to be optimal in that problem, we are motivated to try threshold strategies for this problem. In particular, we try to find a pbp-optimal strategy in which the peripheral sensors all use the identical rule

$$L(y_i) \underset{H_1}{\overset{H_2}{\gtrless}} \left(\frac{\lambda_2}{\lambda_1}\right)^2 \exp[(\lambda_1 - \lambda_2)T] \iff y_i \underset{H_1}{\overset{H_2}{\gtrless}} T. \tag{3.74}$$

Recall that in Section 3.3.3, we provided a pair of sufficient conditions for the optimality of the strategy (3.74). For the Gaussian problem, we analytically demonstrated that the first of the two conditions is always satisfied, and we numerically demonstrated that the second condition is satisfied. Here, we test the validity of the two conditions for our Erlang problem.

As it will turn out, the mathematics is a bit more tractable here than it was in the Gaussian problem, and so we can get closer than we could there to a full analytical proof of the validity of the two conditions. Still, we cannot complete the analytical proof, and so we ultimately resort to numerical experimentation.

There is really nothing conceptually new in this section. The analysis here is a streamlined reprise of the analysis in the last section, and as such, that section should certainly be read first.

3.4.2 Verification of First Condition

To establish the validity of Condition 1 (see Equation 3.45), we use exactly the same line of reasoning that we used for the Gaussian problem. To reiterate, we note that if the peripheral sensor rule (3.74) is used (regardless of its optimality) then, by symmetry, k (defined in Equation 3.48) becomes a sufficient statistic for the fusion rule. Also, as we prove below, if the peripheral sensor rule in (3.74) is used, then the fusion rule becomes monotonic with respect to the k statistic. In turn, this fusion monotonicity property means that Equations 3.52 and 3.53 hold, thus establishing the validity of (3.45). Thus, once we prove that the fusion rule is monotonic, Condition 1 will be verified.

The following Proposition demonstrates the monotonicity of the fusion rule.

Proposition 11 *Suppose that for some threshold T, the rule in (3.74) is used at all of the peripheral sensors. Then, regardless of the value of T, the ratio*

$$\frac{Pr(K = k|H_2)}{Pr(K = k|H_1)}, \tag{3.75}$$

is nondecreasing. In this expression, K is defined in (3.48).

Proof: For notational convenience, we define

$$E_\lambda(x) = \begin{cases} \int_0^x \lambda \exp(-\lambda t)\, dt, & x > 0 \\ 0, & x \le 0. \end{cases} \tag{3.76}$$

In words, $E_\lambda(x)$ is the CDF for an exponential random variable. Now, the monotonicity of (3.75) can be expressed explicitly by the inequality

$$\frac{\int_0^\infty \lambda_2 \exp(-\lambda_2 r)[1 - E_{\lambda_2}(T-r)]^{k+1}[E_{\lambda_2}(T-r)]^{N-k-1}\, dr}{\int_0^\infty \lambda_1 \exp(-\lambda_1 r)[1 - E_{\lambda_1}(T-r)]^{k+1}[E_{\lambda_1}(T-r)]^{N-k-1}\, dr} \quad \ge \quad \frac{\int_0^\infty \lambda_2 \exp(-\lambda_2 r)[1 - E_{\lambda_2}(T-r)]^{k}[E_{\lambda_2}(T-r)]^{N-k}\, dr}{\int_0^\infty \lambda_1 \exp(-\lambda_1 r)[1 - E_{\lambda_1}(T-r)]^{k}[E_{\lambda_1}(T-r)]^{N-k}\, dr} \tag{3.77}$$

which must hold for $0 \le k \le N-1$.

To prove (3.77), we use the following lemma.

Lemma: *The inequality*

$$\frac{\int_0^{L_1} x\, m(x)\, dx}{\int_0^{L_1} m(x)\, dx} \le \frac{\int_0^{L_2} x\, m(x)\, dx}{\int_0^{L_2} m(x)\, dx},$$

holds for all $L_2 > L_1$, $m(x) \ge 0$, *and* $\int_0^{L_1} m(x)dx > 0$.

Proof: First, note that

$$0 \le \frac{\int_0^{L_1} x\, m(x)\, dx}{\int_0^{L_1} m(x)\, dx} \le L_1, \quad L_1 \le \frac{\int_{L_1}^{L_2} x\, m(x)\, dx}{\int_{L_1}^{L_2} m(x)\, dx} \le L_2,$$

and so

$$\frac{\int_0^{L_1} x\, m(x)\, dx}{\int_0^{L_1} m(x)\, dx} \le \frac{\int_{L_1}^{L_2} x\, m(x)\, dx}{\int_{L_1}^{L_2} m(x)\, dx}. \tag{3.78}$$

But from the simple algebraic relationship

$$\frac{a}{b} \le \frac{c}{d} \implies \frac{a}{b} \le \frac{a+c}{b+d}, \quad \text{for } a, b, c, d > 0, \tag{3.79}$$

it follows that (3.78) implies

$$\frac{\int_0^{L_1} x\, m(x)\, dx}{\int_0^{L_1} m(x)\, dx} \le \frac{\int_0^{L_2} x\, m(x)\, dx}{\int_0^{L_2} m(x)\, dx}, \tag{3.80}$$

where we have used (3.79) with the identification

$$\begin{aligned} a &= \int_0^{L_1} x\, m(x)\, dx, \quad b = \int_0^{L_1} m(x)\, dx, \quad c = \int_{L_1}^{L_2} x\, m(x)\, dx, \\ d &= \int_{L_1}^{L_2} m(x)\, dx. \end{aligned}$$

But (3.80) is the desired result, and so we are done. **QED**.

Now, the proof is a tedious application of algebra and elementary calculus. We can rearrange (3.77) to

$$\frac{\int_{-\infty}^{T} \lambda_2 \exp(\lambda_2 r)[1 - E_{\lambda_2}(r)]^{k+1}[E_{\lambda_2}(r)]^{N-k-1}\, dr}{\int_{-\infty}^{T} \lambda_2 \exp(\lambda_2 r)[1 - E_{\lambda_2}(r)]^{k}[E_{\lambda_2}(r)]^{N-k}\, dr} \geq \frac{\int_{-\infty}^{T} \lambda_1 \exp(\lambda_1 r)[1 - E_{\lambda_1}(r)]^{k+1}[E_{\lambda_1}(r)]^{N-k-1}\, dr}{\int_{-\infty}^{T} \lambda_1 \exp(\lambda_1 r)[1 - E_{\lambda_1}(r)]^{k}[E_{\lambda_1}(r)]^{N-k}\, dr}. \tag{3.81}$$

Now, note that for $j = 1, 2$,

$$\frac{\int_{-\infty}^{T} \lambda_j \exp(\lambda_j r)[1 - E_{\lambda_j}(r)]^{k+1}[E_{\lambda_j}(r)]^{N-k-1}\, dr}{\int_{-\infty}^{T} \lambda_j \exp(\lambda_j r)[1 - E_{\lambda_j}(r)]^{k}[E_{\lambda_j}(r)]^{N-k}\, dr} = \frac{\int_{-\infty}^{T} \lambda_j \exp(\lambda_j r)[1 - E_{\lambda_j}(r)]^{k}[E_{\lambda_j}(r)]^{N-k-1}\, dr}{\int_{-\infty}^{T} \lambda_j \exp(\lambda_j r)[1 - E_{\lambda_j}(r)]^{k}[E_{\lambda_j}(r)]^{N-k}\, dr} - 1, \tag{3.82}$$

and so (3.81) can be rearranged to

$$\frac{\int_{-\infty}^{T} \lambda_1 \exp(\lambda_1 r)[1 - E_{\lambda_1}(r)]^{k}[E_{\lambda_1}(r)]^{N-k}\, dr}{\int_{-\infty}^{T} \lambda_1 \exp(\lambda_1 r)[1 - E_{\lambda_1}(r)]^{k}[E_{\lambda_1}(r)]^{N-k-1}\, dr} \geq \frac{\int_{-\infty}^{T} \lambda_2 \exp(\lambda_2 r)[1 - E_{\lambda_2}(r)]^{k}[E_{\lambda_2}(r)]^{N-k}\, dr}{\int_{-\infty}^{T} \lambda_2 \exp(\lambda_2 r)[1 - E_{\lambda_2}(r)]^{k}[E_{\lambda_2}(r)]^{N-k-1}\, dr} \tag{3.83}$$

By the easily-checked identity $E_{\lambda_1}(x) = E_{\lambda_2}[(\lambda_1/\lambda_2)x]$, it follows that (3.83) can be rearranged to

$$\frac{\int_{-\infty}^{(\lambda_1/\lambda_2)T)} f(r) E_{\lambda_2}(r)\, dr}{\int_{-\infty}^{(\lambda_1/\lambda_2)T} f(r)\, dr} \geq \frac{\int_{-\infty}^{T} f(r)\, E_{\lambda_2}(r)\, dr}{\int_{-\infty}^{T} f(r)\, dr}, \tag{3.84}$$

where

$$f(r) = \lambda_2 \exp(\lambda_2(r))[1 - E_{\lambda_2}(r)]^{k}[1 - E_{\lambda_2}(r)]^{N-k-1}.$$

Finally, using the substitution $x = E_{\lambda_2}(r)$ (3.84) becomes

$$\frac{\int_0^{L_2} x\, m(x)\, dx}{\int_0^{L_2} m(x)\, dx} \geq \frac{\int_0^{L_1} x\, m(x)\, dx}{\int_0^{L_1} m(x)\, dx}, \tag{3.85}$$

where

$$L_1 = E_{\lambda_2}(T), \quad L_2 = E_{\lambda_2}(\frac{\lambda_1}{\lambda_2}T), \quad m(x) = \exp[2\lambda_2 E_{\lambda_2}^{-1}(x)][1 - x]^{k} x^{N-k-1}.$$

But the result of the Lemma guarantees that (3.85) is valid, thus proving the proposition. **QED**.

3.4.3 Verification of Second Condition

There is no apparent way to prove the validity of condition 2 (see Equation 3.54 or 3.72). We can, however, prove that (3.57) always has a unique solution. This is computationally useful, because it narrows down to a single threshold the candidate threshold solutions to Equation 3.54.

To prove that (3.57) has a unique solution, we first need to catalog some conditional probabilities. First, it is readily verified (by application of Bayes' law) that

$$f_{R|Y_i}(r|y_i) = \frac{1}{y_i}, \quad 0 \le r \le y_i. \tag{3.86}$$

This result can be used to establish the value for $\Pr[K_i = k-1|y_i, H_j, T]$. In general,

$$\Pr[K_i = k-1|y_i, H_j, T] = \binom{N-1}{k-1} \int_0^\infty f_{R|Y_i}(r|y)[1 - E_{\lambda_j}(T-r)]^{k-1}[E_{\lambda_j}(T-r)]^{N-k}\, dr \tag{3.87}$$

Using (3.86), Equation 3.87 can be simplified as follows:

1. For $1 \le k \le N-1$,

$$\Pr[K_i = k-1|y_i, H_j, T] = \binom{N-1}{k-1} \int_0^{\min(y,T)} \frac{1}{y}[\exp(-\lambda_j(T-r)]^{k-1}[1 - \exp(-\lambda_j(T-r)]^{N-k}\, dr \tag{3.88}$$

2. For $k = N$, $y \le T$,

$$\Pr[K_i = k-1|y_i, H_j, T] = \int_0^y \frac{1}{y}[\exp[-\lambda_j(T-r)]]^{N-1}\, dr.$$

3. For $k = N$, $y > T$,

$$\Pr[K = k-1|y_i, H_j, T] = \frac{y-T}{y} + \int_0^T \frac{1}{y}[\exp[-\lambda_j(T-r)]]^{N-1}\, dr. \tag{3.89}$$

Using the definition of $\alpha'(\cdot)$ (see Equation 3.55), together with (3.88)-(3.89), we see that

$$\begin{aligned} \alpha'(y_i, k^*, T)|_{T=y_i} &= \frac{\Pr[H_1]}{\Pr[H_2]} \frac{\int_0^y [1 - E_{\lambda_1}(y-r)]^{k^*-1}[E_{\lambda_1}(y-r)]^{N-k^*}\, dr}{\int_0^y [1 - E_{\lambda_2}(y-r)]^{k^*-1}[E_{\lambda_2}(y-r)]^{N-k^*}\, dr} \\ &= \frac{\Pr[H_1]}{\Pr[H_2]} \frac{\int_0^y [\exp(-\lambda_1 r)]^{k^*-1}[1 - \exp(-\lambda_1 r)]^{N-k^*}\, dr}{\int_0^y [\exp(-\lambda_2 r)]^{k^*-1}[1 - \exp(-\lambda_2 r)]^{N-k^*}\, dr}, \end{aligned} \tag{3.90}$$

for $1 \leq k^* \leq N$. Coupling (3.73) with (3.90), it is clear that (3.57) will have a unique solution if and only if

$$\left(\frac{\lambda_2}{\lambda_1}\right)^2 \exp[(\lambda_1 - \lambda_2)y] = \frac{\Pr[H_1]}{\Pr[H_2]} \frac{\int_0^y [\exp(-\lambda_1 r)]^{k^*-1}[1-\exp(-\lambda_1 r)]^{N-k^*}\, dr}{\int_0^y [\exp(-\lambda_2 r)]^{k^*-1}[1-\exp(-\lambda_2 r)]^{N-k^*}\, dr}, \tag{3.91}$$

has a unique solution.

Since $\lambda_1 > \lambda_2$, the likelihood ratio on the left-hand side of (3.91) monotonically increases (by inspection). As we presently show, the $\alpha'(\cdot)$ on the right-hand side of (3.91) monotonically decreases. Hence, there exists a single intersection of the two sides of (3.91), and that intersection point is the solution.

We can show that $\alpha'(Y_i = y_i, k^*, T = y_i)$ monotonically decreases by using two steps. First, we let

$$\begin{aligned} f(r) &= [\exp(-\lambda_1 r)]^{k-1}[1-\exp(-\lambda_1 r)]^{N-k}, \\ g(r) &= [\exp(-\lambda_2 r)]^{k-1}[1-\exp(-\lambda_2 r)]^{N-k}. \end{aligned}$$

For $r > 0$, one can show (by tediously computing the derivative) that the ratio $f(r)/g(r)$ monotonically decreases with r.

The second step is to note that for $y > r \geq 0$,

$$\begin{aligned} \frac{f(r)}{g(r)} \geq \frac{f(y)}{g(y)} &\Longrightarrow g(y)f(r) \geq f(y)g(r) \\ &\Longrightarrow g(y)\int_0^y f(r)\, dr \geq f(y)\int_0^y g(r)\, dr \\ &\Longrightarrow \frac{d}{dy}\frac{\int_0^y f(r)\, dr}{\int_0^y g(r)\, dr} \leq 0 \end{aligned}$$

Hence, we have established that $\alpha'(Y_i = y_i, k^*, T = y_i)$ is monotonically decreasing. Thus, (3.57) has a unique solution. **QED**.

In light of (3.88) it is clear that if $k^* < N$, then $\alpha'(\cdot)$ is constant (as a function of y_i) for $y_i \geq T$. Hence, since $L(y_i)$ monotonically increases, we conclude that any solution to (3.57) will also satisfy (3.54) in the region $y_i \geq T$ (when $k^* < N$). This brings us agonizingly close to analytically showing that (3.54) always has a solution. Unfortunately, there is no apparent way to show that solutions to (3.57) are also solutions to (3.54) in the region $y_i < T$. Hence, we resort to a numerical approach in the next section.

3.4.4 Numerical Results

In the last section, we showed that there is always a unique solution to (3.57). For the specific case of $N = 3$, $k^* = 2$, $\Pr(H_1) = \Pr(H_2) = 1/2$, Figure 3.5 numerically shows that the unique solution to (3.57) is also a solution to (3.54).

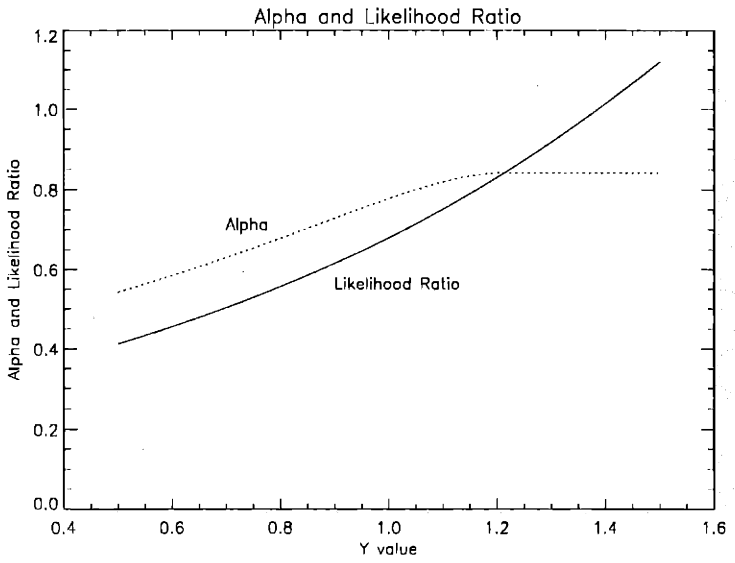

Figure 3.5: $\alpha_i'(y_i, k^*, T^*)$ and $L(y_i)$ functions of y_i.

Conclusion

We have again seen the pbp-optimality of a threshold strategy in a conditionally dependent problem. Unfortunately, no general principles have concomitantly emerged; we only have a qualitative feeling of encouragement that conditionally dependent problems are not always as nasty as the general theory says that they are capable of being.

3.5 Case Study 4: Multiple Hypothesis Gaussian Problem

Beyond stating the pbp-optimality conditions, we have not discussed the multiple hypothesis Bayesian detection problem. Little has been written in the literature on this problem, and virtually nothing is known about the structure of optimal strategies, even for specific cases.

In this section, we establish a slight amount of structure for an optimal strategy of a particular Gaussian detection problem. First, we formulate the problem, and then we prove a proposition concerning optimal strategies for that problem.

3.5.1 Problem Statement

Consider the following observation model, wherein the observation at each peripheral sensor is a signal immersed in Gaussian noise:

$$\begin{aligned} \Pr[S = s_j] &= \Pr(H_j), \qquad 1 \le j \le M \\ Y_i &= S + W_i, \qquad 1 \le i \le N, \\ W_i &\sim \mathrm{N}(0, \sigma^2), \end{aligned} \tag{3.92}$$

with all of the Gaussian random variables W_1, ..., W_N mutually independent, and with $s_1, ..., s_M$ an arbitrary collection of real scalars. What is the optimal strategy, for a given value of D? Note that conditional on the true hypothesis, the peripheral sensor observations are independent.

3.5.2 Structure of Optimal Decision Rules

It is clear that without loss of optimality, each of the peripheral sensors can use a decision rule of the form

$$\gamma_i(y_i) = \begin{cases} d_1, & y_i \leq T_{1i}, \\ d_2, & T_{1i} < y_i \leq T_{2i}, \\ & \vdots \\ d_{k_i+1}, & y_i > T_{k_i i}, \end{cases} \tag{3.93}$$

where

$$\begin{aligned} k_i &\geq 1, \\ d_l &\in \{1, ..., D\}, \quad d_l \neq d_{l+1}, \quad 1 \leq l \leq k_i, \\ T_{1i} &< T_{2i} < ... < T_{k_i i}. \end{aligned} \tag{3.94}$$

This is just a formal way of saying that with respect to the real-axis observation space, any decision rule can be expressed as a set of decision regions. For example, if $D = 2$, then any decision rule can be expressed as alternating regions of "send message 1" and "send message 2." Note that sensor i has k_i different thresholds, each acting as an alternation point from one message region to the next message region.

In general, there is no known bound on the number of regions needed for an optimal decision rule for the multiple hypothesis testing problem. However, for the Gaussian problem just described, there is an upper-bound on k_i $(\forall i)$. The following proposition establishes that bound.

Proposition 12 *For the Gaussian problem described in (3.92), there always exists a strategy of the form (3.93) where*

$$k_i \leq (M-1)\frac{D(D-1)}{2}, \quad 1 \leq i \leq N.$$

Proof: We will need the following lemma for the proof.

Lemma: Let $\alpha_1, \beta_1, ..., \alpha_N, \beta_N$ be an arbitrary collection of finite, real scalars, where $\alpha_i \neq 0$ for at least one value of i, $1 \leq i \leq N$. The equation

$$\sum_{i=1}^{N} \alpha_i \exp(\beta_i x) = 0$$

has no more than $N - 1$ finite real roots.

Proof: We establish the lemma by induction. For $N = 1$, there are clearly no finite real roots. Now, assume that the lemma is true for $N = k \geq 1$, and consider the case $N = k + 1$. Then it is easy to see that

$$\begin{aligned} R\left\{\sum_{n=1}^{k+1} \alpha_n \exp(\beta_n x)\right\} &= R\left\{\exp(\beta_{k+1}x)\left[\alpha_{k+1} + \sum_{n=1}^{k} \alpha_n \exp[(\beta_n - \beta_{k+1})x]\right]\right\} \\ &= R\left\{\alpha_{k+1} + \sum_{n=1}^{k} \alpha_n \exp[(\beta_n - \beta_{k+1})x]\right\} \\ &\leq R\left\{\sum_{n=1}^{k} (\beta_n - \beta_{k+1})\alpha_n \exp[(\beta_n - \beta_{k+1})x]\right\} + 1 \\ &\leq (k-2) + 1. \end{aligned}$$

Here, we have used the notation $R\{\cdot\}$ to denote the number of finite real roots of the enclosed expression. The equalities on the first and second lines are straightforward. The third line follows because the number of roots of a sum of real exponentials is upper-bounded by one more than the number of roots of the derivative of the sum of exponentials. The final line follows from the induction hypothesis. **QED.**

Returning now to the proposition, we first prove the result for the special case of $D = 2$. The generalization to arbitrary D, $D \geq 2$, will then readily follow.

For $D = 2$, the peripheral sensor pbp-optimality condition (2.14) can be written as

$$\gamma_i(y) = \arg\min_{d=1,2} \sum_{j=1}^{M} b_i(d, H_j) f_{Y_i|H_j}(y|H_j), \tag{3.95}$$

where

$$\begin{aligned} b_i(d, H_j) &= \Pr[H_j] a_i(d, H_j, y_i) \\ &= \Pr[H_j] a(i, H_j), \end{aligned}$$

where the second form follows since $a(\cdot)$ is independent of y_i.

A simple rearrangement of (3.95) yields the rule

$$\frac{1}{\sqrt{2\pi}\sigma}\left(\sum_{j=0}^{M-1} c_{ij} \exp[-(y - s_j)^2/2\sigma]\right) \underset{H_1}{\overset{H_2}{\gtrless}} 0,$$

or, equivalently, after multiplying both sides by $\exp(y^2/2\sigma^2)$,

$$\left(\sum_{j=0}^{M-1} \alpha_{ij} \exp(\beta_j y)\right) \underset{H_1}{\overset{H_2}{\gtrless}} 0, \tag{3.96}$$

where $c_{ij} = b_i(d = 1, H_{j+1}) - b_i(d = 2, H_{j+1})$, and

$$\begin{aligned} \alpha_{ij} &= c_{ij} \exp\left(\frac{-s_j^2}{2\sigma^2}\right), \\ \beta_j &= s_j/\sigma^2. \end{aligned}$$

The form of the decision rule in (3.96) makes it clear how to find the thresholds, T_{ij} for the decision rule (3.93). In particular, each real root (with respect to y_i) of the function on the left-hand side of (3.96) marks the location of a threshold. Thus, any upper bound that we can find for the number of real roots of that function is also an upper bound on the number of thresholds in an optimal rule. But from the lemma, we immediately obtain the upper bound of $M - 1$, thus establishing the proposition for the special case of $D = 2$.

Now, we generalize the result to arbitrary D. To motivate the generalization, consider the case of $D = 3$. It is straightforward to see that the number of decision region transitions cannot be more than the number of intersections (as in (3.96)) between decisions 1 and 2 plus between 1 and 3, plus between 2 and 3. But, from the $D = 2$ analysis, the maximum number of intersections for each of these is $M - 1$.

In general, we must consider $\binom{D}{2}$ pairs of intersections, which yields the upper-bound in the Proposition. **QED**.

This result is entirely new. The generalization to $D > 2$ was suggested by [T91].

3.5.3 Comment

Note that our proof of this bound is critically dependent on having the same noise variance, σ^2, for all of the peripheral sensor observations.

Also, it is unknown whether the bound in the proposition is tight; intuition would seem to suggest that it is not. The problem is that when $D = 2$, we might expect that $k_i = 1$, while the upper-bound (for $D = 2$) is $M - 1$. A simple test would be to try finding a set of pbp-optimal regions (by utilizing, say, the nonlinear Gauss-Seidel algorithm), and to count how many regions result. We leave this issue to future work.

Chapter 4

Numerical Methods I

4.1 Nonlinear Gauss-Seidel

The nonlinear Gauss-Seidel algorithm is a general purpose technique for minimizing multivariable functions. The principal is very simple. In the context of our problem, we begin with some strategy, γ^0, that we expect might perform adequately. Next, we generate a sequence of strategies with nonincreasing (and hopefully decreasing) cost, by successively optimizing each of the decision rules. We choose the rule for sensor i in the $(k+1)$th strategy of the sequence via

$$\gamma_i^{k+1} = \arg\min_{\gamma_i} J(\gamma_0^{k+1}, ..., \gamma_{i-1}^{k+1}, \gamma_i, \gamma_{i+1}^{k}, ..., \gamma_N^{k}), \tag{4.1}$$

where $J(\gamma_0, ..., \gamma_N)$ is strategy probability of error, and where all rules but $\gamma_i(\cdot)$ are held fixed. We continue this procedure until the cost decrease (from one iteration to the next) is less than some user-supplied constant, EPS.

The FORTRAN-like code fragment in Figure 4.1 summarizes the algorithm. In the sections that follow, we do operations counts for the various parts of the algorithm. The details of these counts are quite tedious; without loss of continuity, the uninterested reader can skip directly to Section 4.1.7.

4.1.1 Initialization

Finding a reasonable initial strategy is not necessarily easy, and is more an art than a science. For now, we assume that its challenges have been surmounted; we will revisit the issue in the next chapter, in the context of some specific examples.

4.1.2 Fusion

The fusion rule is stored as a table. There are D^N different combinations of messages that the fusion center can receive, and each combination needs its own explicit entry in the table. To fill in the table, we use the pbp-optimality condition (2.11), which

```
INITIALIZE γ_1, ..., γ_N
k = 0                                  !count of number of iterations
BUILD_FUSION_TABLE(γ_0^0; γ_1^0, ..., γ_N^0)
error_min = J(γ^0)
DO
    DO i = 1, N                        !Optimize w.r.t. each peripheral sensor
        γ_i^{k+1} = ARG MIN_{γ_i} (γ_0^{k+1}, ..., γ_{i-1}^{k+1}, γ_i, γ_{i+1}^k, ..., γ_N^k)
        BUILD_FUSION_TABLE(γ_0^{k+1}; γ_1^{k+1}, ..., γ_i^{k+1}, γ_{i+1}^k, ..., γ_N^k)
    END DO
    k = k + 1
    error = error_min
    error_min = J(γ^k)                 !error of current best strategy
WHILE((error - error_min) > eps)
```

Figure 4.1: Code fragment for heart of nonlinear Gauss-Seidel algorithm.

requires that we first compute

$$\Pr[U_1 = u_1, ..., U_N = u_N | H_j], \quad 1 \le u_1, ..., u_N \le D, \quad 1 \le j \le M. \tag{4.2}$$

Depending on whether $U_1, ..., U_N$ are conditionally independent or dependent, we use different methods to compute these joint PMFs.

Conditionally independent case

In the conditionally independent case, the following identity is used to simplify the computation of the joint PMFs,

$$\Pr[U_1 = u_1, \ ..., \ U_N = u_N | H_j] = \prod_{i=1}^{N} \Pr(U_i = u_i | H_j). \tag{4.3}$$

There are NMD distinct marginal PMF terms of the form $\Pr[U_i = j | H_k], 1 \le i \le N, \ 1 \le j \le D, \ 1 \le k \le M$. We compute them all in advance, and then we combine them appropriately to generate the MD^N joint PMF terms of the form (4.2). The cost of doing the numerical integration to compute a single marginal PMF is denoted by $\mathrm{O}(I)$. Then, the total cost of constructing the fusion table is $\mathrm{O}(NMDI + MD^N)$ arithmetic operations.

Note that no matter how large N is, only one dimensional numerical integration is required to generate the joint PMF terms, and hence to generate the fusion table. This is in marked contrast to the conditionally dependent case, as we will presently see.

Conditionally dependent case

In the conditionally dependent case, we can no longer factor a joint PMF into a product of marginal PMFs. Thus, we are forced to numerically evaluate MD^N distinct N-dimensional integrals, each requiring $O(I^N)$ effort. In all, then, constructing the fusion table requires $O[M(DI)^N]$ arithmetic operations. If I is large, then this becomes quite formidable.

4.1.3 Parametrization of Peripheral Sensor Rules

We first describe the methods that we use to parametrize the decision rules. Then, we discuss the computational complexity of actually finding the rule parameters.

For the binary hypotheses, conditionally independent problem, each sensor uses a threshold rule. We parametrize each rule by a two-column array; in one column are $D-1$ likelihood ratio thresholds, and in the other column are D messages. For example, the decision rule

$$\gamma_i(y) = \begin{cases} 2 & \\ 1 & \text{for } 0.3 \le L(y) < 1.1 \\ 3 & \end{cases} \quad \begin{matrix} L(y) < 0.3 \\ \\ L(y) \ge 1.1 \end{matrix} \tag{4.4}$$

would be stored in the array as

$$\begin{matrix} 0.3 & 2 \\ 1.1 & 1 \\ \text{blank} & 3. \end{matrix} \tag{4.5}$$

In light of Proposition 4, the second column of the array in (4.5) may seem superfluous. However, Proposition 4 does NOT preclude non-monotone threshold strategies from being optimal. Since there is nothing in the Gauss-Seidel algorithm that constrains strategies to be monotone, and non-monotone strategies can be optimal, we do indeed need to keep track of the information in the second column of the array in (4.5).

For all multiple hypotheses problems and for all conditionally dependent problems, we again parametrize by a two-column array. Now, however, there are two differences from the case discussed above. An example will clarify.

For our example, we consider the decision rule

$$\gamma_i(y) = \begin{cases} 1 & \\ 2 & \text{for } -0.5 \le y < 2.3 \\ 1 & \end{cases} \quad \begin{matrix} y < -0.5 \\ \\ y \ge 2.3 \end{matrix} \; . \tag{4.6}$$

We would store this rule in the array

$$\begin{matrix} 2 & \text{blank} \\ -0.5 & 1 \\ 2.3 & 2 \\ \text{blank} & 1. \end{matrix} \tag{4.7}$$

The 2 at the top of the left column denotes the number of thresholds in the rule. This counter is needed because we do not know in advance how many thresholds the rule will use; this is the first difference from the binary hypothesis, conditionally independent case. The second difference is that the thresholds are with respect to the y-values themselves (rather than with respect to, say, some likelihood ratio). This is an arbitrary choice for the binary hypothesis, conditionally dependent problem, but is a necessity for all multiple hypothesis problems.

In the sections that follow, we describe the computational complexity of finding the thresholds for the rules.

4.1.4 Once a-Coefficients are Known

The pbp-optimality condition for the peripheral sensor rules is given in (2.14). Note that part of that condition are the a-coefficients, which are defined in (2.15). For now, we assume that the values of these a-coefficients are known and we describe how we then proceed in, first, the special case of a binary hypothesis, conditionally independent problem and, second, in all other cases. Later, we will return to fill in the missing details concerning the a-coefficients.

Binary hypotheses

Recall from the derivation of Proposition 4 that for the binary hypothesis problem (with conditionally independent observations), we can reformulate (2.14) as in (2.20).

Thus, we have D lines corresponding to the D different messages, and for each message we must find the portion of the L-axis for which the corresponding line in the minimum. Given a set of lines, $y_d = c_{1d} + c_{2d}x, \quad 1 \leq d \leq D$, the following is a systematic procedure for finding where line d has the lowest ordinate (i.e., is minimum). We carry out the procedure D times, once for each message.

For $d = 1$, we must solve the $(D-1)$ simultaneous equations

$$r_{2j}x \quad \leq \quad r_{1j}, \quad 2 \leq j \leq D, \tag{4.8}$$

where

$$r_{1j} \quad = \quad b(1,2) - b(j,2) \quad r_{1j} = b(j,1) - b(1,1). \tag{4.9}$$

Clearly, these should be very straightforward to solve. The code fragment in Figure 4.2 collapses this set of inequalities into a single inequality,

$$\text{left\_edge} \leq x < \text{right\_edge}. \tag{4.10}$$

There are probably more elegant ways to solve for the decision regions, though this method is both conceptually simple and adequately fast for the values of D that are of interest. With computational cost $O(D^2)$, we can fill in the two-dimensional array (described above) for the parametrization of a single sensor's rule.

```
left_edge = 0
right_edge = ∞
Do j = 2, D
    IF (r_2j .eq. 0) THEN
        IF (r_1j < 0) THEN
            ERROR
            RETURN
        ENDIF
    ELSE IF (r_2j < 0) THEN
        left_edge = MAX(left_edge, r_1j/r_2j).
    ELSE
        right_edge = MIN(right_edge,r_1j/r_2j).
    ENDIF
END DO
```

Figure 4.2: Code fragment to find endpoints of decision region

All Other Cases

For the given sensor, we define C sample points on the sensor's real-axis observation space. We let $c_k = c_1 + (k-1)\Delta$, $1 \leq k \leq C$, where c_k is the k-th sample point. We select C, c_1, and Δ so that the resulting points are satisfactorily representative of the whole observation space. The exact meaning of this will become clear as the discussion progresses.

We evaluate $d_k = \gamma_i(y)|_{y=c_k}$, $1 \leq k \leq C$, using the pbp-optimality condition as it is expressed in (2.14). Thus, for each k, we must find the minimum amongst D summations of M terms each. Since, for now, we are assuming that we already know the a-coefficients, it follows that we can straightforwardly find $d_1, ..., d_C$ with a total of $O(MDC)$ arithmetic operations.

For concreteness, we explain the next step through an example. Suppose that

$$f_{Y_i|H_j}(y|H_j) = \frac{\exp[-(y-s_j)^2/2\sigma^2]}{\sqrt{2\pi}\sigma}, \quad j = 1, 2, \tag{4.11}$$

that $M = 2$, $D = 3$, and that $C = 7$. Suppose, further, that we have computed $(d_1, ..., d_7) = (1, 1, 2, 2, 3, 3, 3)$. This set of values of d_k suggest that there are two thresholds, $c_2 < T_1 < c_3$ and $c_4 < T_2 < c_5$ such that the pbp-optimal decision rule is

$$\gamma_i(y) = \begin{cases} 1 & y < T_1 \\ 2 \quad \text{for} & T_1 \leq y < T_2 \\ 3 & y \geq T_2 \end{cases}. \tag{4.12}$$

If we have sampled the y-axis finely enough (we will return to this issue momentarily), then the rule in (4.12) will indeed have the correct form. Then, we can find

T_1 more accurately by noting that it is the value of y for which the function $g_{1,2}(y)$ crosses zero, where

$$g_{d_1,d_2}(y) = \sum_{j=1}^{M} f_{Y_i|H_j}(y|H_j)\Pr(H_j)[a_i(d = d_1, H_j) - a_i(d = d_2, H_j)]. \qquad (4.13)$$

Hence, we build the function $g_{12}(y)$ and look for a root of it in the interval $c_2 \leq y \leq c_3$; the y-coordinate of that root is T_1. In a similar manner, T_2 is the root of $g_{23}(y)$ in the interval $c_4 \leq y \leq c_5$. Since finding a bracketed root of a one-dimensional function is routine, this procedure turns out to work quite well. With computational cost $O(MDC)$ (ignoring, for now, the a-coefficients), we can fill in the 2-D array (as described in (4.7)) for the parametrization of a single sensor's rule.

We have mentioned that sampling is an important issue. In the context of our example, a finer sampling of the y-axis might reveal that we should use *two* thresholds between c_2 and c_3. Similarly, a broader range of sampling might reveal the need for a threshold at, say, $c_1 - 1.66$. The point is that we must have values for C, c_1 and Δ that allow us to be fairly certain that we do not miss a transition from one decision region to another.

Numerical experimentation indicates that common-sense and trial-and-error work quite well for finding appropriate sampling parameters. For example, in our Gaussian problem, above, most of the observations will fall in the region

$$\frac{s_1 + s_2}{2} - 3\sigma \leq y \leq \frac{s_1 + s_2}{2} + 3\sigma; \qquad (4.14)$$

thus common sense suggests trying $c_1 = [(s_1 + s_2)/2] - 3\sigma$, and $C \approx 6\sigma/\Delta$. This choice will likely handle the sampling *range* issue. Selecting a value for Δ to handle the sampling *fineness* issue is more tricky. Trial and error seems to be helpful. For instance, if in the above Gaussian example we had computed $(d_1, ..., d_7) = (1, 2, 1, 3, 2, 1, 2)$, then we would suspect that we missed some transitions between decision regions, which would dictate trying a smaller Δ (and so a larger C). This issue cannot be discussed intelligently in the abstract. We will return to it in the context of specific problems in the next chapter.

4.1.5 Computing a-Coefficients

Now, we discuss computation of the a-coefficients, which are defined in (2.15). Note that each coefficient is a sum of $O(D^{N-1})$ probabilities, where each summand is the value of a $(N-1)$-dimensional joint PMF. For example, for sensor 1, we are summing probabilities of the form $\Pr[U_2 = u_2, ..., U_N = u_N|H_k]$ for various values of u_2, ..., u_N and k.

The issues involved in computing these probabilities are the same as the issues involved in computing the probabilities that are needed to fill in the fusion table; in particular, the probabilities are fairly easy to compute in the conditionally independent case and are quite difficult to compute in the conditionally dependent case. See Section 4.1.2 for details.

There is another issue that separates the conditionally independent and dependent cases. In the conditionally independent case the a-coefficients are independent of y. Thus, when they are used to update a peripheral sensor's decision rule (see previous section), they need only be computed once at the beginning of the rule update. Then, that same set of coefficients can be used to evaluate the pbp-optimal $\gamma_i(\cdot)$ for any desired value of y. This can be done with a total of $\mathrm{O}(NMDI + MD^N)$ arithmetic operations per sensor rule update.

On the other hand, in the conditionally dependent case, the a-coefficients are not independent of y. Hence, they must all be recomputed for every value of y for which we want to know the pbp-optimal $\gamma_i(y)$. This adds severe computational overhead. In particular, this can be done with $\mathrm{O}(CM(ID)^N))$ arithmetic operations per sensor rule update.

4.1.6 Computing Probability of Error

Computing the probability of error for a given strategy requires the same order of computational effort as does computing the fusion rule for a given strategy.

4.1.7 Summary of Computational Complexity

The following summarizes the number of arithmetic operations needed for one iteration of the Gauss-Seidel algorithm. Here, we define a single iteration to be the collection of operations between successive increments to the counter k in the code fragment of Figure 4.1. Also, as stated before, $\mathrm{O}(I)$ is the number of arithmetic operations needed to numerically evaluate a single one-dimensional integral, while C is the number of cells into which we divide each sensors' real-axis observation space.

In the binary hypothesis, conditionally independent case,

$$\begin{aligned} \frac{\text{arithmetic operations}}{\text{iteration}} &= N[\mathrm{O}(D^2) + \mathrm{O}(NMDI + MD^N)] + \\ &\qquad N\mathrm{O}(NMDI + MD^N) \\ &= \mathrm{O}(N^2MDI + ND^N). \end{aligned} \tag{4.15}$$

In the multiple hypothesis, conditionally independent case,

$$\begin{aligned} \frac{\text{arithmetic operations}}{\text{iteration}} &= N[\mathrm{O}(MDC) + \mathrm{O}(NMDI + MD^N)] + \\ &\qquad N\mathrm{O}(NMDI + MD^N) \\ &= \mathrm{O}[NMDC + N^2MDI + NMD^N]. \end{aligned} \tag{4.16}$$

In all conditionally dependent problems,

$$\begin{aligned} \frac{\text{arithmetic operations}}{\text{iteration}} &= N[\mathrm{O}(MDC)\mathrm{O}[CM(ID)^N]] + N\mathrm{O}[M(DI)^N] \\ &= \mathrm{O}[ND(MC)^2(ID)^N + NM(ID)^N]. \end{aligned} \tag{4.17}$$

4.1.8 Practical Experience

I tested this algorithm on the Gaussian problems that are described in the first two case studies of the previous chapter. For the conditionally dependent problem, the algorithm is not very useful for $N > 2$; the required numerical integration is too burdensome. For the $N = 2$ case, on the other hand, the algorithm will converge (for $D = 2, M = 2$) in a matter of minutes (on a VAX workstation). Very low values of C can be used; I had success with $C \approx 5$.

4.1.9 Convergence

At each step of the algorithm, the strategy cost is monotonically nonincreasing and is bounded below (zero cost is a lower bound). These two properties guarantee that the cost will converge. The difficulty is that the cost to which the algorithm converges might only correspond to the cost of a pbp-optimal strategy rather than to the cost of a globally optimal strategy.

In practive, however, the algorithm has been found to work quite well. For conditionally independent Gaussian problems (like the one analyzed in Case Study 1 in the previous chapter), the Gauss-Seidel has been found to work extremely well. In experiments that I have done on a VAX workstation, with $M = 2$, $D = 2$ or 3, and with $2 \leq N \leq 7$, the routine does indeed converge to a pbp-optimal strategy. The convergence occurs within roughly five iterations and, in terms of real time, is virtually instantaneous. Furthermore, for the Gaussian problem, it seems that most pbp-optimal strategies are also globally optimal; hence the algorithm seems to be finding globally optimal strategies virtually instantaneously.

4.1.10 Applicability to Neyman-Pearson Problem

The Gauss-Seidel algorithm, as we have presented it, is not directly applicable to the Neyman-Pearson problem. Our algorithm has been based on the Bayesian pbp-optimality conditions, which do not accommodate prior specification of a false alarm rate. The tantalizing part is that from Proposition 7 we know that whatever optimum our Gauss-Seidel algorithm does yield will also be an optimum of a Neyman-Pearson problem for some false alarm rate, but not likely the false alarm rate that we want. However, under the critical assumption that the set of points $Q = \left\{(J^F(\gamma), J^D(\gamma)) | \gamma \in \Gamma\right\}$ is convex, we can make progress.

Suppose that we want to achieve a false alarm rate of α. Our convexity assumption, coupled with Proposition 8, means that if we scan the set of Bayesian optimal strategies corresponding to different values of $\Pr(H_2)$, then we are guaranteed to find a strategy with the desired false alarm probability α. As $\Pr(H_2)$ monotonically increases from zero to one, the false alarm rate of the Bayesian optimum will also monotonically increase from zero to one.

To efficiently scan the Bayesian optima, we treat the Neyman-Pearson problem as a root-finding exercise. In particular, any root of the nonlinear equation

$$\alpha - J^F(\gamma'(P_2)) = 0 \tag{4.18}$$

corresponds to an optimal strategy for the Neyman pearson problem. In this expression, $P_2 = \Pr(H_2)$ is the independent variable, and γ' is the optimal Bayesian strategy for the given value of P_2. If we are fortunate, our detection problem will not have very many pbp-optima that are not also global optima. Then, in numerically solving (4.18), we can harness the Gauss-Seidel algorithm to evaluate $\gamma'(P_2)$ efficiently.

4.1.11 Historical Notes

The nonlinear Gauss-Seidel algorithm was described in [TPK89], though it was probably used by other researchers prior to 1989. In [TPK89], a Gaussian detection problem is solved with the algorithm, thus demonstrating that from a practical perspective, the algorithm can have very satisfactory performance. The algorithm is also briefly discussed in [T89].

4.2 Chernoff Exponent Minimizer

One difficulty with the Gauss-Seidel algorithm is its great combinatorial complexity when there are a large number of peripheral sensors. One way to reduce computational complexity is to exclusively search over strategies that use identical sensor rules. Unfortunately, there is no apparent way to streamline the Gauss-Seidel algorithm so that its search is so restricted. Thus, we consider a heuristically-based alternative search algorithm for the binary hypothesis testing problem. The *Chernoff exponent minimizer* provides a natural way to search for good (though suboptimal) identical decision strategies. To motivate the method, we first pause to explain the theoretical utility of the Chernoff exponent. Then, we derive the algorithm.

4.2.1 Background on Chernoff Exponent

Suppose that we want to quantify the benefit (in terms of decreasing probability of error) of increasing the number of peripheral sensors in a decentralized detection system. For any given strategy, with a given number of peripheral sensors, the exact probability of error expression is (when the hypotheses are equiprobable)

$$J_{error}(\gamma) = \frac{1}{2} \sum_{u_1,\ldots,u_N} \min_{j=1,2} \{\Pr[u_1,\ldots,u_N|H_j]\}. \tag{4.19}$$

This expression provides virtually no insight into how $J_{error}(\gamma)$ decreases as more sensors are added to the system. The problem is that there is no decoupling of the error reduction contribution of each sensor. This is where the Chernoff exponent plays a role.

Consider the simple inequality (seemingly plucked out of thin air)

$$\min(x,y) < x^s y^{1-s}, \quad x,y > 0, \quad 0 < s < 1. \tag{4.20}$$

We apply (4.20) to (4.19) to yield

$$J_{error}(\gamma) \leq \frac{1}{2}\exp(\mu(s)), \tag{4.21}$$

where

$$\mu(s) = \log\left[\sum_{u_1,\ldots,u_N} \Pr^{1-s}[u_1,\ldots,u_N|H_1]\Pr^{s}[u_1,\ldots,u_N|H_2]\right]. \tag{4.22}$$

The inequality in (4.21) holds in general, but is most useful when the peripheral sensor observations are independent conditioned on the true hypothesis.

When the conditional independence property holds, then by routine algebra, (4.22) can be rearranged to

$$\exp(\mu(s)) = \sum_{u_1,\ldots,u_N} \Pr[u_1,\ldots,u_N|H_1] \cdot \prod_{k=1}^{N} \exp(sL(u_k)), \tag{4.23}$$

where $L(\cdot)$ is the log likelihood ratio,

$$L(u_k) = \log \frac{\Pr[\gamma_k(Y_k) = u_k|H_2]}{\Pr[\gamma_k(Y_k) = u_k|H_1]}. \tag{4.24}$$

We assume that the summation in (4.23) is only over the N-tuples (u_1, ..., u_N) satisfying

$$\Pr[u_1,\ \ldots,\ u_N|H_1] \neq 0.$$

Equation 4.23) expresses $\exp(\mu(s))$ as a conditional expectation of a product of conditionally independent random variables. Since in general

$$\mathrm{E}\left[\prod_{i=1}^{n} X_i\right] = \prod_{i=1}^{n} \mathrm{E}(X_i), \tag{4.25}$$

when $X_1,\ldots,X_n$ are mutually independent (and the expectation exists), we can further rearrange (4.23) to

$$\exp(\mu(s)) = \prod_{i=1}^{N} \exp[\mu_i(\gamma_i, s)), \tag{4.26}$$

or

$$\mu(s) = \sum_{i=1}^{N} \mu_i(\gamma_i, s), \tag{4.27}$$

where

$$\mu_i(\gamma_i, s) = \log\left[\sum_{u_i=1}^{D} \Pr^{1-s}[u_i|H_1]\Pr^{s}[u_i|H_2]\right]. \tag{4.28}$$

Our notation explicitly brings out the dependence of $\mu_i(\cdot)$ on the decision rule used at peripheral sensor i.

We summarize these findings in the following proposition.

Proposition 13 *For any set of peripheral sensor decision rules, there exists a fusion rule such that the overall strategy, γ, satisfies*

$$[1 - J^D(\gamma)] + J^F(\gamma) \leq \exp[\mu(s^*)], \tag{4.29}$$

where $J^D(\gamma)$ and $J^F(\gamma)$ are defined in Section 2.8, and where s^* minimizes $\mu(s)$ over $0 < s < 1$. In (4.29) $\mu(s)$ is in general defined as in (4.22) but for the special case of conditionally independent peripheral sensor observations, $\mu(s)$ decouples as in (4.27) and (4.28).

Some additional points will make the bound in (4.29) more useful. First, in general $\mu(s)$ is guaranteed to be non-positive. One can readily show (though we will not here) that the only way $\mu(s)$ can be identically equal to zero is for the joint distribution of $U_1, ..., U_N$ to be identical conditioned on either hypothesis; in all other cases, $\mu(s)$ will be strictly negative. This is important, because it suggests that in the conditionally independent case, the probability of error will drop exponentially fast (with N) for any reasonable set of decision rules.

Our second point concerns the asymptotic tightness of (4.29). Suppose that the conditional independence property holds, and that the marginal distributions of the peripheral sensor observations are identical, given either hypothesis. If all of the peripheral sensors use the decision rule $\gamma_1(\cdot)$ and the fusion center uses the MAP rule, then under some technical assumptions (likely to be satisfied in cases of interest), the overall strategy will satisfy

$$\lim_{N \to \infty} \frac{\log J_{error}(\gamma)}{N} = \mu_1(\gamma_1, s^*). \tag{4.30}$$

The proof of this result is fairly involved; see [SGB67] and [T88] for details.

4.2.2 Why the Chernoff Exponent?

We can now see the possible merit of basing a strategy search algorithm on the minimization of the Chernoff exponent. The exponent provides a measure of dissimilarity between the joint peripheral sensor message distributions under the two hypotheses. Naturally, we want the dissimilarity to be as great as possible, so that the fusion center's decision problem is as easy as possible. We also have reassurance that at least asymptotically (as $N \to \infty$), the exponent provides a very good measure of distributional distance.

The second, more important reason for considering the Chernoff exponent is that in the conditionally independent case, minimization of the exponent decouples into a separate, independent optimization problem for each sensor rule. This decoupling occurs in a natural way that was seemingly unachievable with the Gauss-Seidel algorithm.

4.2.3 Algorithm Derivation

There are two versions of the algorithm: one for the conditionally independent case and one for the conditionally dependent case. We derive the conditionally

independent version in detail; then, because the conditionally dependent version is so similar, we do not derive it, but instead only summarize the algorithm steps.

Minimizing with respect to s

The algorithm that we will describe treats s as a constant; with respect to any fixed value of s, $0 < s < 1$, the algorithm find the best decision rule, $\gamma_i(\cdot)$. One practical way to minimize $\mu_i(\gamma_i, s)$ with respect to s as well, is to repeatedly rerun the algorithm, each time with a different value of s; by so "sweeping" the values of s, the $\gamma_i(\cdot)$ that truly minimizes the Chernoff exponent can be found.

Conditionally independent case

For each peripheral sensor, we wish to minimize $\mu_i(\gamma_i, s)$. Incidentally, note that if the marginal distributions of $Y_1, ..., Y_N$ are all identical, then all of the sensor rules for the Chernoff exponent minimizing strategy will be identical. The procedure is the same for minimizing each of the $\mu_i(\gamma_i, s)$ terms, so we focus only on $\mu_1(\cdot)$.

The first step is to initialize $\gamma_1(\cdot)$ to some rule that we expect will effect reasonably different distributions for $\Pr(U_1|H_1)$ and $\Pr(U_1|H_2)$. As with the Gauss-Seidel algorithm, there is no known optimal method for carrying out this initialization.

Next, using the initial decision rule, we compute

$$L(U_1 = u_1) = \frac{\Pr[U_1 = u_1|H_2]}{\Pr[U_1 = u_1|H_1]} \tag{4.31}$$

for $1 \leq u_1 \leq D$. This likelihood ratio is used in the next step, which is also the most critical step.

To improve on the initial $\gamma_1(\cdot)$, we note that

$$\begin{aligned}
\exp(\mu_1(\gamma_1, s)) &= \sum_{d=1}^{D} \Pr^{1-s}[d|H_1]\Pr^{s}[d|H_2] \\
&= \sum_{d=1}^{D}(1-s)\Pr[d|H_1]L^{s}(d) + s\Pr[d|H_2]L^{s-1}(d) \\
&= \int dy_1 \sum_{d=1}^{D}(1-s)f_{Y_1|H_1}(y_1|H_1)L^{s}(\gamma_1(y_1)) + \\
&\qquad s f_{Y_1|H_2}(y_1|H_2)L^{s-1}(\gamma_1(y_1))
\end{aligned} \tag{4.32}$$

Now, we update $\gamma_1(\cdot)$ via

$$\gamma_1(y_1) = \underset{d=1,...,D}{\arg\min} \left\{(1-s)f_{Y_1|H_1}(y_1|H_1)L^{s}(d) + s f_{Y_1|H_2}(y_1|H_2)L^{s-1}(d)\right\}, \tag{4.33}$$

where we use the old values of $L(\cdot)$, as computed in the first step. For each value of y, this rule minimizes the integrand of (4.32) with respect to the fixed $L(\cdot)$. The rule's structure is quite similar to the pbp-optimality condition's structure (2.14).

It is clear that this updated $\gamma_1(\cdot)$ will not increase (and will hopefully decrease) the value of (4.32) when used with the old value of $L(\cdot)$ is used. What is not clear is whether any gains we have made will be lost when $L(\cdot)$ is updated to its new value as effected by the new $\gamma_1(\cdot)$.

Fortunately, updating $L(\cdot)$ to its correct new value further lowers the value of (4.32). To see this, consider the problem from the perspective of finding a functional $g(u)$ that minimizes

$$h(g(\cdot), s) = \sum_{d=1}^{D} (1-s)\Pr(d|H_1)g^s(d) + s\Pr(d|H_2)g^{s-1}(d).$$

Taking derivatives,

$$\frac{\partial h(\gamma_1, s)}{\partial [g(u_1)]} = s(1-s)\Pr[u_1|H_1]g^{s-1}(u_1) - s(1-s)\Pr[u_1|H_2]g^{s-2}(u_1),$$

and so

$$\frac{\partial h(\gamma_1, s)}{\partial [g(u_1)]} \Big|_{g(u_1)=L(u_1)} = 0.$$

Also, at the stationary point (with respect to $g(u_1)$),

$$\begin{aligned} \frac{\partial^2 h(\gamma_1, s)}{\partial [g(u_1)]^2} \Big|_{g(u_1)=L(u_1)} &= \Pr[u_1|H_1]L^{s-2}(u_1)s(1-s) \\ &> 0. \end{aligned}$$

Hence, the minimizing function is $g(u) = L(u)$, and so we update $L(\cdot)$ to its correct new value.

Following the update of $L(\cdot)$, we have a new, improved decision rule. Then, the whole procedure is repeated. We summarize the overall algorithm in the FORTRAN-like code fragment in Figure 4.3. The code is hopefully self-explanatory.

Comments

A few comments are in order concerning the above derivation. Most importantly, note that (4.33) can be equivalently expressed as

$$\gamma_1(\ell(y_1)) = \underset{d=1,\ldots,D}{\arg\min} \; [b_{1d} + b_{2d}\ell(y)], \tag{4.34}$$

where the constants b_{1d}, b_{2d} are given by

$$b_{1d} = (1-s)L^s(d), \quad b_{2d} = sL^{s-1}(d), \quad 1 \le d \le D.$$

Also,

$$\ell(y) = \frac{f_{Y|H_2}(y|H_2)}{f_{Y|H_1}(y|H_1)},$$

INITIALIZE $\gamma_1(\cdot)$
COMPUTE $L(\cdot)$
mu_min = CHERNOFF_EXPONENT(γ_1, L)
$k = 0$
DO

$\gamma_1^{k+1} =$ ARG MIN$_{\gamma_1}$ CHERNOFF_EXPONENT (γ_1^k, L)

UPDATE $L(\cdot)$.
$k = k + 1$
mu = mu_min
mu_min = CHERNOFF_EXPONENT(γ_1^k, L)
WHILE (mu - mu_min > eps)

Figure 4.3: Code fragment for conditionally independent Chernoff exponent minimizer

which is distinctly different from $L(\cdot)$.

Note that (4.34) has exactly the same form as (2.20). The conclusion, then, is that a threshold rule can always be used to minimize the Chernoff exponent (in the conditionally independent case). Though our derivation (patterned after [FG87], [LLG90]) is appropriate for developing the algorithm, the derivation belies the simple underlying principle that makes threshold rules optimal.

In particular, one can readily show that the Chernoff exponent is a convex function of the probabilities $\Pr[\gamma_i(Y_i) = d|H_j]$, $1 \leq d \leq D$, $j = 1, 2$. More precisely, let Q be the $2D$-vector

$$Q = [Q_1^T \;\; Q_2^T]^T,$$

where

$$Q_j = [\Pr[\gamma_1(Y_1) = 1|H_j] \; ... \; \Pr[\gamma_1(Y_1) = D|H_j]]^T, \quad j = 1, 2.$$

Then, $\mu_i(Q; s)$ is concave in Q. Now, it is shown in [T89] that in general, there exists a deterministic threshold strategy that minimizes $f(Q)$ for any $f(\cdot)$ that is continuous and concave in Q. We will not pursue this point further; we only mention it in passing.

Conditionally Dependent Case

For the conditionally dependent case, the optimization of the peripheral sensor rules no longer decouples. However, it is possible to develop an iterative algorithm in which the optimization of the rules is coupled. The algorithm is a rather natural extension of our conditionally independent case algorithm. For this extension, the goal is to minimize (4.22). For that reason, we only summarize its steps.

1. Initialize decision rules $\gamma_1(\cdot), ..., \gamma_N(\cdot)$.

2. Compute $L(U_1 = u_1, ..., U_N = u_N)$ for $1 \leq u_1, ..., u_N \leq D$.

3. Fix decision rules $\gamma_2(\cdot), ..., \gamma_N(\cdot)$, and also fix $L(\cdot)$. Now, update $\gamma_1(\cdot)$ via

$$\gamma_1(y) =$$

$$\arg\min_{d=1,...,D} \sum_{u_2,...,u_N} (1-s) f_{Y_1|H_1}(y_1|H_1) \Pr[u_2, ..., u_N|y_1, H_1] L^s(d, u_2, ..., u_N) +$$
$$s f_{Y_1|H_2}(y_1|H_2) \Pr[u_2, ..., u_N|y_1, H_2] L^{s-1}(d, u_2, ..., u_N) \quad (4.35)$$

4. Update $L(\cdot)$.

5. Repeat steps 2 through 4 $(N-1)$ more times, each time optimizing the next decision rule (i.e., $\gamma_2(\cdot)$, then $\gamma_3(\cdot), ...$) while holding all of the other rules fixed.

6. At this point, we have completed one pass of optimizing the whole strategy. Now, we can start all over, again beginning by optimizing $\gamma_1(\cdot)$.

4.2.4 Historical Notes

Our derivation of the algorithm is patterned after [FG87] and [LLG90]. In both of these papers, a key point was missed. That is, as shown in [T89], for the conditionally independent problem, a strategy that minimizes the Chernoff exponent can always be found within the class of threshold strategies. In [LLG90], the authors do numerical experiments with a Gaussian problem that is very similar to the Gaussian problem that analytically consider in Chapter 3. In [LLG90], the authors did not compare their numerically-derived "optimal" strategy with the true optimal, or with the "optimal" from a Gauss-Seidel algorithm.

4.2.5 Practical Experience

I implemented the conditionally dependent version of the Chernoff exponent minimizer to find strategies for the Gaussian problem described in the second case study of the previous chapter. I did NOT minimize with respect to s, but rather fixed s at the value one-half. The reason for fixing s was that the algorithm would have otherwise been prohibitively slow.

For this particular (Gaussian) problem, the Chernoff exponent minimizer did not compare well with the nonlinear Gauss-Seidel algorithm. The difficulty is that for the conditionally dependent case, the Chernoff minimizer has virtually identical computational complexity to the Gauss-Seidel algorithm. The Gauss-Seidel directly optimizes the cost criterion that we are actually interested in (i.e., the probability of error), unlike the Chernoff exponent minimizing algorithm. In fact, I found for the Gaussian problem, that as the Chernoff exponent decreased, the probability of error would increase. This ill behavior might disappear with larger values of N or D, but those larger values are too computationally complex to actually run.

In conclusion, I did not have very much success with the Chernoff exponent minimizer.

4.3 Product of Marginals

As we have seen, finding an optimal strategy for a conditionally dependent problem can be computationally expensive. We consider here a heuristic for reducing this computation.

The idea is simple. We formulate a nearby problem (in a sense to be shown) that is conditionally independent. Since conditionally independent problems are much less computationally burdensome than their conditionally dependent counterparts, we can hopefully solve our nearby problem easily. Then, we try using the resulting strategy in our original, conditionally dependent problem. If we are fortunate, the performance will be adequate.

We choose the density for our nearby, conditionally independent problem in the following eminently reasonable way as follows,

$$f^{\text{nearby}}_{Y_1,\ldots,Y_N|H_j}(y_1,\ldots,y_N|H_j) = \prod_{i=1}^{N} f_{Y_i|H_j}(y_i|H_j). \tag{4.36}$$

In words, we use the product of the marginal densities from our original problem as the joint density in our nearby problem.

The following example shows that, in general, there is no limit to how poorly this heuristic will perform.

- **Example**

  Suppose that the true, conditionally dependent joint-density is

$$f_{Y_1,Y_2|H_1}(y_1,y_2|H_1) = \begin{cases} \dfrac{\exp[-(y_1^2+y_2^2)/2]}{\pi}, & y_1<0, \text{ and } y_2\geq 0, \text{ or} \\ & y_1\geq 0 \text{ and } y_2<0, \end{cases} \tag{4.37}$$

  and

$$f_{Y_1,Y_2|H_2}(y_1,y_2|H_2) = \begin{cases} \dfrac{\exp[-(y_1^2+y_2^2)/2]}{\pi}, & y_1\geq 0, \text{ and } y_2\geq 0, \text{ or} \\ & y_1<0 \text{ and } y_2<0. \end{cases}, \tag{4.38}$$

  Note that under either hypothesis, the marginal densities are

$$f_{Y_i|H_j}(y_i,H_j) = \frac{\exp(-y^2/2)}{\sqrt{2\pi}}, \quad i=1,2, \quad j=1,2. \tag{4.39}$$

  Thus,

$$f^{\text{nearby}}_{Y_1,Y_2|H_j}(y_1,y_2|H_j) = \frac{\exp[-(y_1^2+y_2^2)/2]}{2\pi}, \quad j=1,2, \tag{4.40}$$

and so the nearby problem has the same joint-distribution, conditioned on either hypothesis. Thus, an optimal strategy for the nearby problem is for the fusion center to simply always decide in favor of H_1. This will have a probability of error of 0.5 in the original problem.

On the other hand, the following strategy (based on the actual PDF) has a probability of zero in the original problem:

$$\gamma_i(y_i) = \begin{cases} 0, & y_i < 0, \\ 1, & y_i \geq 0, \end{cases} \quad i = 1, 2. \tag{4.41}$$

The MAP rule is used at the fusion center.

Chapter 5

Numerical Methods II

5.1 Random Array Problem

The Gaussian problem that we have considered is rather benign in the sense that all reasonable strategies seem to have roughly the same performance. There has been only a negligible difference between the optimal solution and the suboptimal alternatives. In retrospect, we can see why this is so.

1. Excluding degenerate cases (see example in Chapter 2), there are virtually no pbp-optimal strategies that are not coincidentally globally optimal. Thus, the Gauss-Seidel does not get stuck far from the optimal.

2. The optimal strategy has been experimentally found to always be a threshold strategy using identical rules at the peripheral sensors.

3. The problem is insensitive enough to suboptimality that an optimal fusion rule (which is always possible to use) can compensate for most of the deficiencies of suboptimal peripheral sensor rules.

One serious difficulty that we had with the Gaussian problem was that numerical integration was a greater stumbling block than were any of the combinatorial issues. Since we are really not interested in numerical integration issues, we would like to find a problem that is challenging combinatorially while requiring trivial numerical integration.

In this section, we examine a problem that fits our desired description. First, we formulate the problem, and describe some of the implementation issues associated with the numerical experimentation. Then we summarize the experimentation results.

5.1.1 Problem Formulation

This problem is a special case of the NP-complete problem that was described in Section 2.6 of Chapter 2. We are given a square $S \times S$ incidence matrix its entries are all zeros and ones), A_1, and its complement, $A_2 = E - A_1$, where E is a matrix

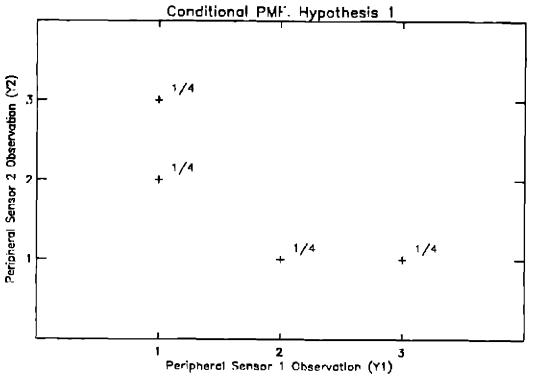

Figure 5.1: Example probability mass function.

of all ones (i.e. $E_{ij} = 1, 1 \leq i, j \leq S$). From the matrices, we construct the joint PMF

$$f_{Y_1,Y_2|H_k}(i,j|H_k) = \frac{(A_k)_{ij}}{z_k}, \quad 1 \leq i,j \leq S, \quad k = 1,2, \tag{5.1}$$

and z_k is the normalizing constant

$$z_k = \sum_{i=1}^{S}\sum_{j=1}^{S}(A_k)_{ij}, \quad k = 1,2. \tag{5.2}$$

To make the conditional marginal distributions identical (i.e., $f_{Y_1|H_k}(y|H_k) = f_{Y_2|H_k}(y|H_k), \ k = 1,2$), we constrain A_1 to be symmetric (which then automatically makes A_2 symmetric). Lastly, we constrain

$$1 \leq \sum_{i=1}^{S}(A_1)_{ij} \leq S - 1 \tag{5.3}$$

so that $f_{Y|H_k}(y|H_k) > 0, \ \forall y, \ k = 1,2$.

We assume that $\Pr(H_1) = \Pr(H_2) = 1/2$. Also, each peripheral sensor has a code alphabet of size D. We want to find a strategy that minimizes the probability of error.

Figures 5.1 and 5.2 show the conditional PMFs that correspond to the $S = 3$ matrix

$$A_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{5.4}$$

The plots make clear the triviality of the centralized version of this problem. That is, knowledge of both y_1 and y_2 uniquely specifies the true hypothesis, and so if a sensor had access to both of these values, then it can always achieve perfect detection performance. On the other hand, the decentralized version is quite nontrivial.

In a sense, this problem is rather whimsical, because it seems so devoid of any real-world application. However, we are not treating it as a link to a genuine

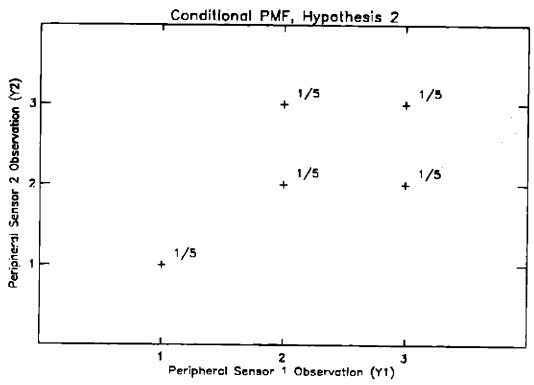

Figure 5.2: Example probability mass function.

detection system, but rather as a worst-case scenario for our detection algorithms. Intuitively, we might expect that unlike the Gaussian problem, this one will have many pbp-optima (many, perhaps, far from optimal). Also, there is no apparent reason to believe that a threshold strategy will be optimal, or that an identical peripheral sensor rule strategy will be optimal.

5.1.2 Incidence Matrix Generation

We describe the random matrix generator that we used to generate $S \times S$ incidence matrices that satisfy the constraints listed in the last section. As we will see, the generated matrices come from a uniform distribution over all valid (as constrained in the last section) $S \times S$ matrices.

In the first step of the generation process, we ignore the constraint in (5.3). Without this constraint, we are left with the problem of selecting uniformly from the class of $S \times S$ symmetric incidence matrices. But this is easy. We generate $S(S+1)/2$ independent, identically distributed (iid) Bernoulli random variables, $X_1, ..., X_{S(S+1)/2}$ with $\Pr(X_i = 0) = \Pr(X_i = 1) = 1/2, \ \forall i$. These values are then arranged into a lower-triangular $S \times S$ matrix. For example, in the 3×3 case, we would generate 6 Bernoulli random variable values, and we would arrange them as

$$\begin{bmatrix} x_1 & & \\ x_2 & x_4 & \\ x_3 & x_5 & x_6 \end{bmatrix}. \tag{5.5}$$

Actually, the configuration in (5.5) completely specifies the symmetric matrix; we reflect values across the diagonly to yield the complete matrix

$$A_1 = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_2 & x_4 & x_5 \\ x_3 & x_5 & x_6 \end{bmatrix}. \tag{5.6}$$

It is easy to see that this procedure selects uniformly from the set of $2^{S(S+1)/2}$ symmetric incidence matrices.

The second step of the generation process is to check whether the constraint in (5.3) is satisfied. If it is, then we generate $A_2 = E - A_1$, and we are done. If (5.3) is not satisfied, then we discard our current A_1, and we begin the whole process from scratch. There are perhaps efficient methods that bypass this rejection step, but the algorithm just described is conceptually simple and in practice seems to work virtually instantaneously (for the matrix sizes that we are interested in).

5.1.3 Exhaustive Search

Because the observation space at each peripheral sensor is discrete, it is possible to do an exhaustive search over all strategies. In this section, we show the results of a simple numerical experiment in which we do exhaustive searches. These searches will be for the case of $D = 2$.

We use the random-array generator that was described in the last section. Each new $S \times S$ array yields a new optimal-strategy search problem. We compare the performance of the optimal strategy from each of the following four classes:

1. General strategies (this is where the global optimum will always be found),
2. General strategies with the same rule used at both peripheral sensors,
3. General threshold strategies (this is NOT guaranteed to be optimal, since the problem is not conditionally independent),
4. Threshold strategies with the same threshold at both peripheral sensors.

Table 5.1.3 summarizes the findings for this experiment. The table requires some explanation. For each value of S ($S = 5, 7, 8$), we created three different random arrays, and for each array, we searched for the optimum strategies. That is why, in the table, there are three rows of error values of each value of S. In the *General Strategies / All* column, the error of the globally optimal strategies are given. In the columns corresponding to the other three classes of strategies, the deterioration of the error–relative to the error of the optimal strategy–is given. The deterioration is defined by

$$\text{Deterioration} \equiv \frac{\text{Error of suboptimal strategy}}{\text{Error of optimal strategy}}. \tag{5.7}$$

Thus, a strategy with a deterioration of unity is optimal, while a deterioration greater than unity is suboptimal. The values of S used in this experiment are quite modest. This is because for $S > 8$, the run-time for an exhaustive search becomes excessive.

Of course, the best general strategy is always globally optimal. Also the best threshold strategy always performs at least as well as the best threshold strategy with identical rules. Beyond these two statements, though, there are only trends, not certain relationships between the performance of different techniques. *Usually*, but not always, the globally optimal strategy uses identical rules at the peripheral

	General Strategies		*Threshold Strategies*	
	All	Identical	All	Identical
Number to consider ($S = 5$)	225	15	16	4
	0.1987	1.000	1.677	1.677
Deterioration	0.1667	1.428	1.952	1.952
	0.1603	1.000	1.959	1.959
Number to consider ($S = 7$)	3,969	63	36	6
	0.2241	1.000	1.022	1.022
Deterioration	0.2284	1.064	1.000	1.064
	0.2724	1.059	1.408	1.448
Number to consider ($S = 8$)	16,129	127	49	7
	0.2210	1.000	1.413	1.413
Deterioration	0.2282	1.000	1.546	1.546
	0.1875	1.000	1.444	1.444

sensors. Also usually, but not always, the identical-rule, general strategy performs better than the general threshold strategy. It is unknown whether these qualitative trends in the performance of different strategies could somehow be captured by an analytical relationship. This very issue has frusturated researchers for at least a decade.

5.1.4 Nonlinear Gauss-Seidel

In this section, we consider the sensitivity of the Gauss-Seidel algorithm to different initializing strategies. The discussion will be rather informal. This informality is appropriate, because the only purpose of this experiment is to obtain a *qualitative* feel for the performance of the Gauss-Seidel algorithm.

We used the random-array generator to construct three different instances of the hypothesis testing problem; the constructed instances have dimensions $S = 10$, 20, and 30, respectively.

For each instance, the Gauss-Seidel algorithm was run three times. Each time it was run, it was initialized with a different strategy. These initializing strategies were chosen randomly.

Note that, in general, there does not seem to be a great deal of sensitivity to the initializing strategy. Usually, within three to four iterations, the algorithm reaches a pbp-optimum cost level.

Unfortunately, the dimesion of these instances is too large to find the globally optimal strategy. Hence, it is only a conjecture that the strategy performance shown in the figures is close to optimal.

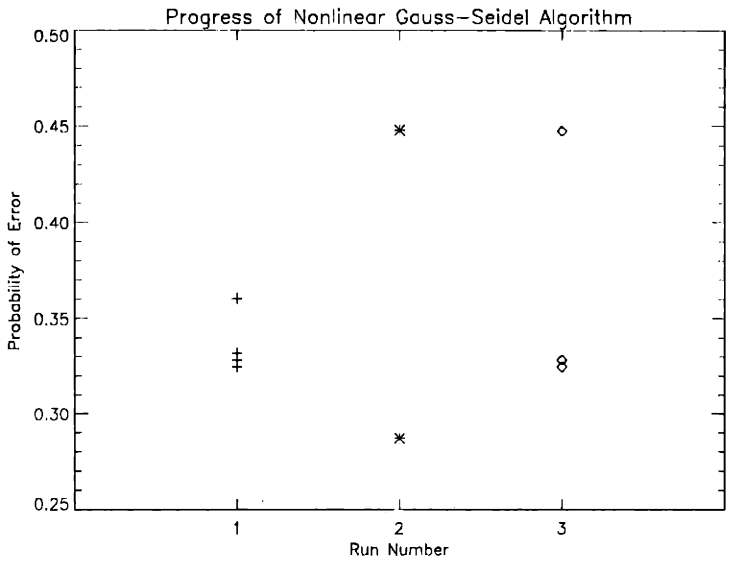

Figure 5.3: Progress of Nonlinear Gauss-Seidel Algorithm, from various initializing strategies for $S = 10$.

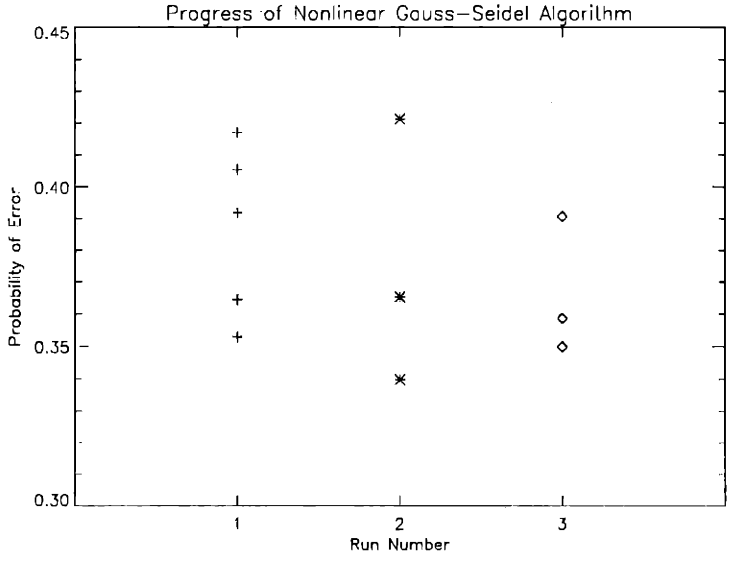

Figure 5.4: Progress of Nonlinear Gauss-Seidel Algorithm, from various initializing strategies for $S = 20$.

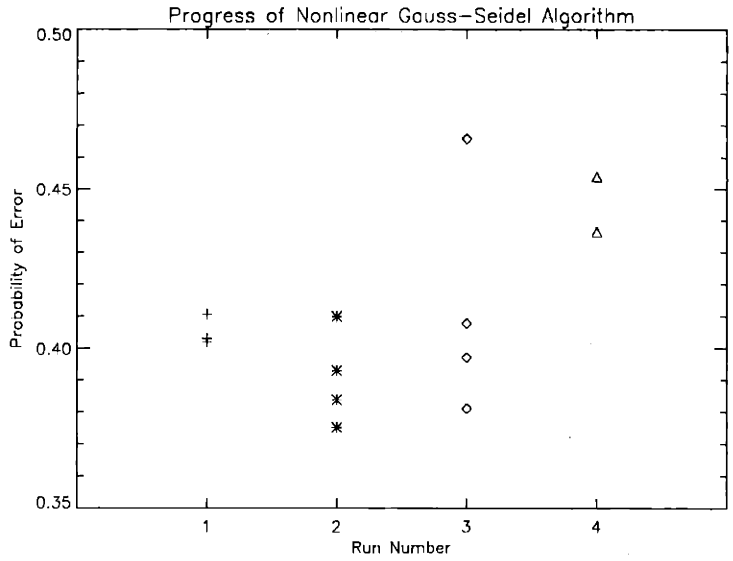

Figure 5.5: Progress of Nonlinear Gauss-Seidel Algorithm, from various initializing strategies for $S = 30$.

Chapter 6

Parameter Estimation

6.1 Introduction

In this chapter, we consider a decentralized parameter estimation problem. First, is a formal statement of the problem. As we will see, the formulation is very similar to that of the hypothesis testing problem (see Chapter 2), with two critical differences. The first difference is that the underlying hypothesis has a continuous rather than discrete (e.g., H_1, H_2) distribution. The second difference is that the cost criterion is now the mean-square error (MSE).

Next, are two lower bounds on the MSE of any decentralized strategy. The first is related to the a posteriori mean, $E(A|Y_1 = y_1, ..., Y_N = y_N)$, where A is the random parameter. The second is a version of the Cramer-Rao bound for biased estimates of random parameters.

Next is a derivation of the person-by-person optimality conditions. From a computational standpoint, it will turn out that these conditions are not very useful for finding a good strategy; the conditions require too much computation (especially numerical integration). For this reason, we consider alternatives to using the pbp-optimality conditions.

In the context of a particular decentralized parameter estimation problem, we consider three suboptimal alternatives to the pbp-optimality conditions. The three alternatives are based on different "educated" guesses about how a low-MSE strategy might be structured; after guessing a structure, we try to find the best strategy within the structured class that we have guessed.

6.2 Formulation

Here is the formal problem statement.

6.2.1 Problem Statement

We have a scalar random parameter, A, that has probability density $f_A(a)$, $L_1 \leq a \leq L_2$, with L_1, L_2 possibly infinite. Detecting the parameter are N peripheral

sensors, with sensor i $(1 \leq i \leq N)$ receiving scalar, real-values observation y_i. We assume that conditional on A, the realizations of the peripheral sensor observations obey the known joint PDF $f_{Y_1,...,Y_N|A}(y_1,...,y_N|a)$, with

$$f_{Y_1,...,Y_N|A}(y_1,...,y_N|a) = f_{Y_1|A}(y_1|a) \cdot ... \cdot f_{Y_N|A}(y_N|a), \tag{6.1}$$

meaning that conditional on the true parameter value, the peripheral sensor observations are independent.

Each peripheral sensor, upon receiving its observation, evaluates a message $u_i = \gamma_i(y_i) \in \{1,...,D\}$. All N messages (one from each peripheral sensor) are sent to the fusion center, where the final decision, $\gamma_0(u_1,...,u_N)$, is made. The goal is to find a deterministic strategy, γ, that minimizes the mean-square error,

$$J(\gamma) = E\{[A - \gamma_0(\gamma_1(Y_1),...,\gamma_N(Y_N))]^2\}. \tag{6.2}$$

The expectation on the right-hand side of (6.2) is over A, and $Y_1,...,Y_N$.

6.2.2 Comment on the Formulation

As with all parameter estimation problems, there is some arbitrariness in the choice of what error criterion to use. One reason that we have chosen to use the MSE is that it is well-established criterion in the literature on centralized parameter estimation [VT68]. A related reason is that the criterion lends some analytic tractability to the problem; we will find that we can analytically establish some bounds on the MSE of decentralized strategies.

It is, however, important to bear in mind that other error criteria are also possible. One alternative is the expected absolute value of the error,

$$E\{|A - \gamma_0(\gamma_1(Y_1,...,\gamma_N(Y_N))|\}.$$

We will leave consideration of the alternatives to future work.

Also note that we have restricted attention to the conditionally independent case. This is consistent with our choice to avoid multiple hypothesis, conditionally dependent hypothesis testing problems. Simply put, we do not consider the conditionally dependent parameter estimation problem because it is more difficult than the conditionally independent problem, which is itself difficult enough.

6.3 Lower Bounds for Mean-Square Error

In this section, we establish two lower bounds on the MSE of decentralized parameter estimation strategies.

6.3.1 Relation to Aposteriori Mean

We prove the following proposition.

Proposition 14 *The following inequality holds for any N-peripheral sensor strategy, γ:*

$$J(\gamma) \geq E\{[E(A|Y_1, ..., Y_N) - A]^2\}, \tag{6.3}$$

where the left-hand side is defined in (6.2).

Proof: Note that we can rewrite (6.2) as

$$\begin{aligned} J(\gamma) = E\{[(A - E(A|Y_1, ..., Y_N)) + \\ (E(A|Y_1, ..., Y_N) - \gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_1)))]^2\}, \end{aligned} \tag{6.4}$$

where we have simply added and subtracted $E(A|Y_1, ..., Y_N)$. But the right-hand side of this can be expanded into three terms. First is

$$E\{[A - E(A|Y_1, ..., Y_N)]^2\}, \tag{6.5}$$

which is the mean-square error of the (centralized) aposteriori mean estimator. Second is

$$2E\{[A - E(A|Y_1, ..., Y_N)][E(A|Y_1, ..., Y_N) - \gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_N))]\}. \tag{6.6}$$

Since the orthogonality principle says that

$$E\{[A - E(A|Y_1, ..., Y_N)][g(Y_1, ..., Y_N)]\} = 0 \tag{6.7}$$

for any function $g(Y_1, ..., Y_N)$, it follows that (6.6) is identically zero. The third term in the expansion of (6.4) is

$$E\{[E(A|Y_1, ..., Y_N) - \gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_N))]^2\}, \tag{6.8}$$

which is the mean-square error between the centralized aposeteriori mean estimator and the decentralized strategy estimator. Since this last term must be greater than or equal to zero, the inequality (6.3) is established. **QED**.

This proof has been based on [LR90].

6.3.2 Comment on Aposteriori Mean Bound

This last result has an interesting interpretation. Combining Equations 6.5 through 6.8, we obtain

$$\begin{aligned} J(\gamma) \;=\; & E\{[A - E(A|Y_1, ..., Y_N)]^2\} \;+ \\ & E\{[E(A|Y_1, ..., Y_N) - \gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_N))]^2\}. \end{aligned} \tag{6.9}$$

Since the strategy γ is independent of the first term on the right-hand side, the best the strategy can do is to minimize the second term on the right-hand side. But minimizing the second term is the same as minimizing the mean-square difference between the centralized aposteriori mean estimator and the decentralized strategy estimator. Thus, the optimal decentralized strategy "mimics" the centralized aposteriori mean estimator as best it can.

6.3.3 Cramer-Rao Bound

The bound in Proposition 14 is often difficult to compute. In this section, we consider (a version of) the Cramer-Rao bound, which can sometimes be easier to compute than the above bound.

Proposition 15 *The mean-square error of any estimator $\hat{A}(\bar{Y})$ satisfies the inequality*

$$E\{[\hat{A}(\bar{Y}) - A]^2\} \geq \frac{[1 - \beta(\hat{A})]^2}{E\left\{\left[\frac{\partial}{\partial A} \ln f_{\bar{Y},A}(\bar{Y}, A)\right]^2\right\}}, \tag{6.10}$$

where the expectations on both sides are over both A and $\bar{Y}$. In this expression, $\bar{Y} = [Y_1\ Y_2\ ...\ Y_N]^T$; also,

$$\beta(\hat{A}) = f_A(a = L_2)B(\hat{A}; a = L_2) - f_A(a = L_1)B(\hat{A}; a = L_1), \tag{6.11}$$

with

$$B(\hat{A}; a) = E[\hat{A}(\bar{Y})|A = a] - a. \tag{6.12}$$

Two technical conditions that must be satisfied for the bound to hold are the inequalities

$$\int_{L_1}^{L_2} \left|\frac{\partial f_{\bar{Y},A}(\bar{y}, a)}{\partial a}\right| da < \infty, \quad \int_{-\infty}^{\infty} \left|\frac{\partial f_{\bar{Y},A}(\bar{y}, a)}{\partial a}\right| d\bar{y} < \infty. \tag{6.13}$$

Proof: First, we multiply both sides of (6.12) by $f_A(a)$, and differentiate with respect to a:

$$\frac{d}{da}[f_A(a)B(\hat{A}; a)] = -\int_{-\infty}^{\infty} f_{\bar{Y},A}(\bar{y}, a)\, d\bar{y} + \int_{-\infty}^{\infty} \frac{\partial f_{\bar{Y},A}(\bar{y}, a)}{\partial a}[\hat{A}(\bar{Y}) - a]\, d\bar{y}. \tag{6.14}$$

Now, integration with respect to a yields

$$f_A(a)B(\hat{A}; a)|_{L_1}^{L_2} = -1 + \int_{L_1}^{L_2}\int_{-\infty}^{\infty} \frac{\partial f_{\bar{Y},A}(\bar{y}, a)}{\partial a}[\hat{A}(\bar{y} - a]\, d\bar{y}\, da. \tag{6.15}$$

Note that the left-hand side of this last expression is $\beta(\hat{A})$. For the right-hand side,

$$\frac{\partial f_{\bar{Y},A}(\bar{y}, a)}{\partial a} = \frac{\partial \ln f_{\bar{Y},a}(\bar{y}, a)}{\partial a} f_{\bar{Y},A}(\bar{y}, a), \tag{6.16}$$

and hence, by the Cauchy-Schwarz inequality, the integral on the right-hand side of (6.15) is bounded by

$$\left[\int_{L_1}^{L_2}\int_{-\infty}^{\infty} \frac{\partial f_{\bar{Y},A}(\bar{y}, a)}{\partial a}[\hat{A}(\bar{y}) - a]\, d\bar{y}\, da\right]^2 \leq$$

$$E\{[\hat{A}(\bar{Y}) - A]^2\}E\{[\frac{\partial}{\partial A} \ln f_{\bar{Y},A}(\bar{Y}, A)]^2\}. \tag{6.17}$$

Finally, applying this last inequality to (6.15) yields (6.10), thus establishing the proposition. **QED**

This proof has been based on some minor modifications of the Cramer-Rao bound discussion in [VT68].

6.3.4 Comment on Cramer-Rao Bound

With regard to this last bound, note that as the estimator $\hat{A}(\cdot)$ changes, the bound (6.10) changes, through its dependence on $\beta(\hat{A})$. This has the unfortunate consequence that if $\exists\ \hat{A}\ s.t.\ \beta(\hat{A}) = 1$, then (6.10) becomes the useless (though correct) inequality

$$E\{[\hat{A}(\bar{Y}) - A]^2\} \geq 0. \tag{6.18}$$

Thus, to make (6.10) useful, it is necessary to somehow constrain $\beta(\hat{A})$. Then, the bound will be valid for all estimators satisfying the constraint. Fortunately, constraining $\beta(\hat{A})$ is, in its own right, a reasonable thing to do. This is because constraining $\beta(\hat{A})$ is equivalent to constraining the bias of the estimator $\hat{A}(\cdot)$.

We defer discussing particular constraints on $\beta(\hat{A})$ until we introduce the specific parameter estimation problem that will be the focus of the second-half of this chapter.

6.3.5 Comment on Both Bounds

One shortcoming of the bounds in Propositions 14 and 15 is that neither one has any D-dependence. To more carefully describe the consequences of this lack of D-dependence, let $\Gamma(D)$ be the set of all deterministic strategies that have message alphabet-size D at the peripheral sensors. Then, clearly, $\Gamma(D_1) \subset \Gamma(D_2)$, for $D_2 > D_1$, and thus, $J(\gamma^1) \geq J(\gamma^2)$, where γ^j is an optimal strategy for $\gamma^j \in \Gamma(D_j)$, $j = 1, 2$. Thus, unless $J(\gamma^D) = J(\gamma^{D'})$ for some small D and all $D' \geq D$, it follows that our bounds *cannot* be tight for the values of D that are of interest to us.

In the specific parameter estimation problem that we consider later in this chapter, the Cramer-Rao bound will turn out to be $J(\gamma) \geq k/N$, where k is some constant, and N is (as usual) the number of peripheral sensors. For the same problem, we will find a strategy that satisfies $J(\gamma) = k'/N$ for some $k' > k$. Thus, the bound will not be met with equality (as we have discussed that it likely cannot be), but the order of decay, $O(\frac{1}{N})$, is the same for both the bound and the actual strategy. Thus, at least heuristically, there is some reason to believe that the actual strategy is performing quite satisfactorily. It is in this comforting, heuristic sense that the Cramer-Rao bound will be useful in our work.

6.4 Person-by-Person Optimality Conditions

The following is a development of necessary conditions for an optimal strategy. First, we discuss the fusion rule, and then the peripheral sensor rules.

6.4.1 Fusion Rule

For the fusion rule, we have the following proposition.

Proposition 16 *With respect to fixed peripheral sensor decision rules* $\gamma_1(\cdot), ..., \gamma_N(\cdot)$, *the optimal fusion rule is*

$$\gamma_0(u_1, ..., u_N) = E[A|u_1, ..., u_N], \tag{6.19}$$

which is the aposteriori mean.

Proof: We first expand (6.2) to the convenient form

$$J(\gamma) = \sum_{u_1,...,u_N} \Pr[U_1 = u_1, ..., U_N = u_N] \int_{L_1}^{L_2} [a - \gamma_0(u_1, ..., u_N)]^2 \cdot f_{A|U_1,...,U_N}(a|u_1, ..., u_N)\, da. \tag{6.20}$$

Note that because the peripheral sensor rules are fixed, it follows that

$$\Pr[u_1, \ ..., \ u_N] \text{ and } f_{A|U_1, \ ..., \ U_N}(a|u_1, \ ..., \ u_N)$$

are fixed. Thus, of the terms in (6.20), the fusion rule can only influence the value of $\gamma_0(u_1, \ ..., \ u_N)$. For each summand (i.e., each combination $(u_1, \ ..., \ u_N)$) the optimal fusion value $\gamma_0(u_1, \ ..., \ u_N)$ is the value, r, that minimizes the integral

$$\int_{L_1}^{L_2} [a - r]^2 f_{A|U_1, \ ..., \ U_N}(a|u_1, \ ..., \ u_N)\, da.$$

But it is well-known that the mean of the aposteriori density is the minimizer; that is, $r = \gamma_0(u_1, \ ..., \ u_N) = E(A|u_1, \ ..., \ u_N)$, thus proving the proposition. **QED.**

This result is not at all surprising. Just as with hypothesis testing, we can view the set of peripheral sensor messages as a vector of observations that the fusion center receives. Thus, from the perspective of the fusion center, the problem is a classical centralized parameter estimation problem with associated density $f_{A|U_1,...,U_N}(a|u_1, ..., u_N)$. Since the minimizer of the mean-square error (in the centralized problem) is the aposteriori mean, the fusion rule (6.19) must be optimal.

6.4.2 Peripheral Sensor Rules

The following proposition gives necessary conditions for optimal peripheral sensor decision rules.

Proposition 17 *With respect to fixed sensor decision rules*

$$\gamma_0(\cdot), \ ..., \ \gamma_{i-1}(\cdot), \ \gamma_{i+1}(\cdot), \ ..., \ \gamma_N(\cdot),$$

the optimal peripheral sensor rule $\gamma_i(\cdot)$ *is*

$$\gamma_i(y_i) = \underset{d=1,...,D}{arg\ min} \int_{L_1}^{L_2} f_{Y_i|A}(y_1|a) b_i(d, a)\, da, \tag{6.21}$$

where

$$b_i(d, a) = E\{[\gamma_0(\gamma_1(Y_1), ..., \gamma_{i-1}(Y_{i-1}), d, \gamma_{i+1}(Y_{i+1}), ..., \gamma_N(Y_N)) - A]^2 | A \tag{6.22}$$

We do not prove this result, since it requires only a trivial modification of the proof of Proposition 3.

6.4.3 Need for Alternatives to Pbp-Optimality Conditions

As one can see from (6.21), and (6.22) the peripheral sensor pbp-optimality condition is not very helpful for providing insight into the structure of the optimal strategy. Furthermore, consider the computational complexity of using the condition in, say, the Gauss-Seidel algorithm. Computing (6.21) requires integrating an iterated summation of probabilities which themselves require integration to compute. This is very messy, and not very insightful; for these reasons, we do not exploit the peripheral sensor pbp-optimality condition to search for optimal strategies. On the other hand, the fusion center condition is straightforward, and we will exploit it when possible.

6.4.4 Breakpoint Quantizer Strategy

We define a breakpoint quantizer strategy.

> Suppose that there are thresholds t_1, ..., t_{D-1} satisfying $-\infty < t_1 \leq t_2 \leq \ ... \ \leq t_{D-1} < \infty$, and there are intervals $I_1 = [-\infty, t_1]$, $I_2 = [t_1, t_2]$, ..., $I_D = [t_{D-1}, \infty]$. Then, decision rule $\gamma_i(\cdot)$ is called a *monotone breakpoint quantizer rule* if
>
> $$\gamma_i(y_i) = d \text{ only if } y_i \in I_d. \tag{6.23}$$
>
> More generally, decision rule $\gamma_i(\cdot)$ is called *breakpoint quantizer rule* if there exists a permutation mapping $\sigma : \{1, ..., D\} \rightarrow \{1, ..., D\}$ such that $\sigma \circ \gamma_i$ is a monotone breakpoint quantizer rule.
>
> Finally, a strategy is a *breakpoint quantizer strategy* if all of the peripheral sensors use breakpoint quantizer rules.

Note that this definition is closely related to the concept of a threshold strategy. The difference is that a threshold strategy is defined with respect to a likelihood ratio; since there is no likelihood ratio in the parameter estimation problem, the concept of the breakpoint quantizer is needed.

We will try breakpoint quantizer strategies for the estimation problem that is introduced in the following section.

6.5 Gaussian-Corrupted Parameter Estimation Problem

For the rest of this chapter, we concentrate on the following parameter estimation problem.

> We have a scalar random parameter, A, that has uniform distribution,
>
> $$f_A(a) = \frac{1}{L}, \quad 0 \leq a \leq L.$$

Peripheral sensor i receives scalar observation, y_i,

$$Y_i = A + W_i, \quad W_i \sim \mathrm{N}(0, \sigma), \quad 1 \leq i \leq N,$$

with $A, W_1, ..., W_N$ all mutually independent.

For a given value of D ($D \geq 2$), we want to find a strategy that minimizes the mean-square error (6.2).

We are also interested in the tradeoff between number of peripheral sensors and number of bits transmitted per peripheral sensor. For example, if the fusion center cannot receive more than 12 bits, then is it better to have 12 sensors transmitting 1 bit each, or 6 sensors transmitting 2 bits each, or 4 sensors transmitting...?

In the sections that follow, we experiment with various suboptimal strategies. Until section 6.10.3, we exclusively consider the case of $D = 2$. In section 6.10.3, we consider larger values of D in an experiment concerning the tradeoff between N (number of peripheral sensors) and D (message alphabet size per sensor). First, however, we tailor the above Cramer-Rao bound to this problem.

6.6 Reconsideration of Cramer-Rao Bound

In the context of this particular parameter estimation problem, we reconsider the Cramer-Rao bound, as developed in Section 6.3.3. In particular, recall that for the bound (6.10) to be useful, we must somehow constrain $\beta(\hat{A})$ (see Equation 6.11).

Suppose we limit attention to estimators satisfying

$$B(\hat{A}; a) \leq fL, \quad a = 0 \text{ and } a = L, \tag{6.24}$$

where $B(\cdot)$ was defined in (6.12), and where f is any fixed value in the range $0 \leq f < 1/2$. This means, for example, that if $f = 1/4$, A has a valid range of $0 \leq A \leq 10$, and the true value is $A = 5$, then our estimator must produce an estimate with an expected value somewhere in the interval $2.5 \leq E[\hat{A}(\bar{Y})|A = 5] \leq 7.5$. For estimators satisfying the constraint (6.24), it follows from (6.11) that

$$\begin{aligned} |\beta(\hat{A})| &\leq f_A(A = L)|B(\hat{A}; a = L)| + f_A(A = 0)|B(\hat{A}; a = 0)| \\ &\leq 2f \end{aligned} \tag{6.25}$$

and so (6.10) becomes

$$\begin{aligned} E\{[\gamma_0(\gamma_1(Y_1), ..., \gamma_N(Y_N)) - A]^2\} &\geq (1 - 2f)^2\{E[\frac{\partial}{\partial A} \ln f_{Y,A}(Y, A)]^2\}^{-1} \\ &\geq \frac{(1 - 2f)^2\sigma^2}{N}, \end{aligned} \tag{6.26}$$

which is considerably more useful than (6.18). The first line follows immediately from the identification $\hat{A}(\bar{Y}) = \gamma_0(\gamma_1(Y_1), \ ..., \ \gamma_N(Y_N))$. The second line follows from straightforward evaluation of the expectation.

Thus, we see that by introducing a reasonable bias constraint, it is possible to make the Cramer-Rao bound meaningful.

6.7 Design 1: Indicator Functions to Simulate Expectation

In this section, we consider a strategy that has particularly simple structure. Let the peripheral sensors use breakpoint quantizer rules,

$$\gamma_i(y_i) = \begin{cases} 0, & y_i < T_i, \\ 1 & \text{otherwise} \end{cases}, \tag{6.27}$$

where

$$\begin{aligned} T_i &= T_0 + \Delta(i-1), \\ T_0 &= \frac{L}{2} - \Delta\left(\left\lceil \frac{N}{2} \right\rceil - 1\right). \end{aligned}$$

This set of thresholds is evenly spaced about $L/2$, in increments of Δ.

For the fusion center, we abandon the computationally expensive a posteriori mean (6.19), and instead we use the simple summation rule

$$\gamma_0(u_1, ..., u_N) = \Delta \left[\sum_{i \in T^+} u_i \right] - \Delta \left[\sum_{i \in T^-} (1 - u_i) \right], \tag{6.28}$$

where T^+ is the set of i's for which $T_i \geq 0$, and T^- is the set of i's for which $T_i < 0$.

Note that fixed N, there is only a single degree of freedom in the design of this strategy. The degree of freedom is in choosing the optimal spacing (i.e., the value of Δ) between the breakpoints. We show how to find the optimal Δ in section 6.7.2. To make explicit the dependence of $J(\gamma)$ on N and Δ, we express the strategy MSE error by $J(\gamma; N, \Delta)$.

This strategy, which was introduced in [GBL85], is beautifully simple. Its succes is related to the identity

$$E(X) = \left[\int_0^\infty \Pr[X > x]\, dx \right] - \left[\int_{-\infty}^0 \Pr[X \leq x]\, dx \right]. \tag{6.29}$$

This identity can be readily verified via integration by parts and holds for any random variable, X, that has a mean.

The following proposition exploits (6.29) to yield the asymptotic performance of this strategy.

6.7.1 Asymptotic Performance

Proposition 18 *For* $\Delta = 1/\sqrt{N}$,

$$\lim_{N \to \infty} E[\gamma_0(U_1, ..., U_N) | A = a] = a, \tag{6.30}$$

meaning that the strategy is asymptotically unbiased.

Proof: To prove that the strategy is asymptotically unbiased,

$$\begin{aligned}
&\lim_{N\to\infty} E[\gamma_0(U_1,...,U_N)|a] \\
&= \frac{1}{\sqrt{N}}\left[\sum_{i\in T^+} E[u_i]\right] - \frac{1}{\sqrt{N}}\left[\sum_{i\in T^-} 1 - E[u_i]\right] \\
&= \frac{1}{\sqrt{N}}\left[\sum_{i\in T^+} \Pr[y_i \geq T_i|a]\right] - \frac{1}{\sqrt{N}}\left[\sum_{i\in T^-} \Pr[y_i < T_i|a]\right] \\
&\underset{N\to\infty}{\longrightarrow} \left[\int_0^\infty \Phi\left(\frac{a-x}{\sigma}\right)\,dx\right] - \left[\int_{-\infty}^0 \Phi\left(\frac{x-a}{\sigma}\right)\,dx\right] \\
&= a.
\end{aligned}$$

where we have used the relation

$$E[U_i|a] = \begin{cases} \Pr[y_i < T_i|a] & T_i \in T^-, \\ \Pr[y_i \geq T_i|a] & T_i \in T^+ \end{cases} . \tag{6.31}$$

and where we have used the identity (6.29). **QED**

From (6.26), this is the optimal order of error decay, and so asymptotically, this strategy is quite good. The results in Propostion 18 motivate our interest in testing the strategy for finite N.

6.7.2 Performance for Finite Number of Sensors

Figure 6.1 shows how the mean-square error depends on Δ when $L = 10, \sigma = 1, N = 5$. The figure suggests that for the finite N case, it is worthwhile to optimize $J(N,\Delta)$ with respect to Δ.

To do this optimization, we note that

$$J(N,\Delta) = \int_0^L E[(\gamma_0(U_1,...,U_N) - a)^2|a]\,da, \tag{6.32}$$

where

$$E[(\gamma_0(U_1,...,U_N) - a)^2|a] = E[(\gamma_0(U_1,...,U_N))^2|a] - 2aE[\gamma_0(U_1,...,U_N)|a] + a^2,$$

with

$$\begin{aligned}
&\frac{E[(\gamma_0(U_1,...,U_N))^2|a]}{\Delta^2} \\
&= \sum_i E(U_i|a) + \sum_{i\in T^+} E(U_i|a) \sum_{\substack{j\in T^+, \\ j\neq i}} E(U_j|a) \\
&+ \sum_{i\in T^-} E(U_i|a) \sum_{\substack{j\in T^- \\ j\neq i}} E(U_j|a) - 2\sum_{i\in T^+} E(U_i|a) \sum_{j\in T^-} E(U_j|a).
\end{aligned} \tag{6.33}$$

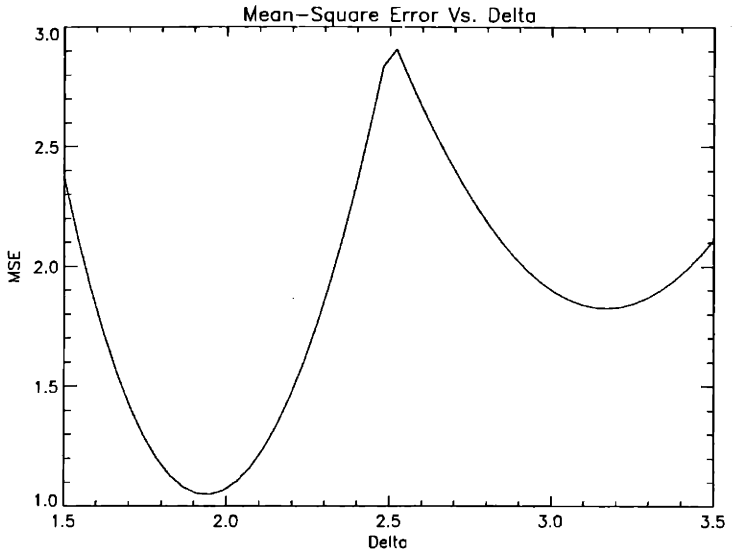

Figure 6.1: Design 1 dependence of mean-square estimation error on Δ when $L = 10, \sigma = 1, N = 5$. Here, $\Delta_{opt} = 1.94$. The discontinuity of the derivative at $\Delta \approx 2.5$ is due to one of the peripheral sensor thresholds moving from T^+ to T^-.

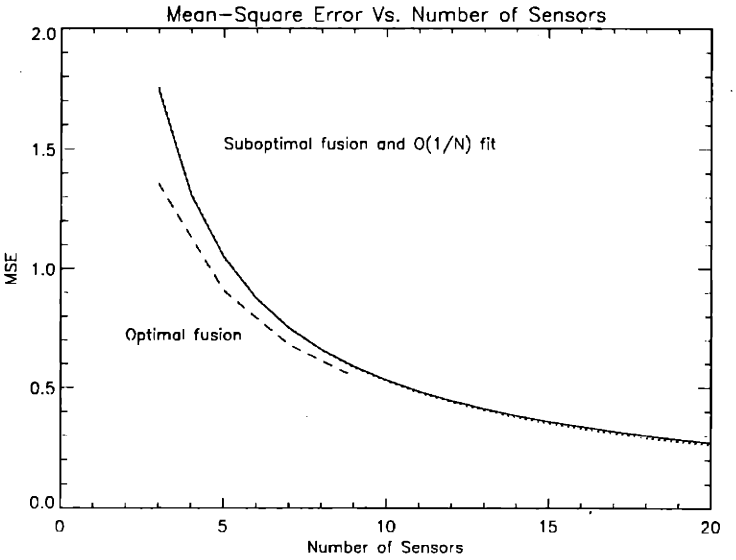

Figure 6.2: Design 1 mean-square estimation error as a function of number of sensors. The overlay dotted curve is $5.26/N$, so the decay rate is certainly $O(\frac{1}{N})$. The dashed curve is the MSE when the optimal MAP rule (6.19) is used in place of the suboptimal rule in (6.28).

The summands in (6.33) are computed via (6.31), and the integration (6.32) is not time-consuming. Hence, finding the optimal Δ is computationally feasible.

Figure 6.2 is a plot of $J(N, \Delta_{opt})$ as a function of N. Note that the error decay is almost perfectly $O(\frac{1}{N})$. Thus, even for finite N, the error decay rate seems quite reasonable.

One might wonder how much has been lost by using the suboptimal fusion rule (6.28) in lieu of the pbp-optimal fusion rule (6.19). The dashed line in Figure 6.2 shows that there is very little difference in the MSE between the two fusion rules; since (6.28) is considerably easier to compute than (6.19), the slight loss in MSE optimality is worthwhile.

In Section 6.8.3, we will compare this strategy design to the other designs to be introduced

6.8 Design 2: Breakpoint Quantizer Strategy

In this section, we again consider breakpoint quantizer strategies, but we no longer constrain the thresholds to be evenly spaced as they were in the last section. Once we give up the even spacing, we can no longer use the fusion rule (6.28). Hence, we use the pbp-optimal fusion rule (6.19).

Because $D = 2$, we can compactly summarize the decision rules used at the peripheral sensors by the N-vector,

$$\bar{T} = [T_1 \; T_2 \; ... \; T_N]^T, \tag{6.34}$$

where T_i is the breakpoint in the breakpoint quantizer rule at peripheral sensor i.

To make explicit the dependence of $J(\gamma)$ on N and on $\bar{T}$, we write $J(\gamma; \bar{T}, N)$. Note that $J(\gamma; N, \bar{T})$ is completely determined by N and $\bar{T}$:

$$J(\bar{T}, N) = \frac{1}{L} \int_0^L da \sum_{u_1,...,u_N} [\gamma_0(u_1, ..., u_N) - a]^2 \prod_i \Pr[U_i = u_i|a]. \tag{6.35}$$

In (6.35),

$$\gamma_0(u_1, ..., u_N) = \int_0^L a \; f_{A|U_1,...,U_N}(a|u_1, ..., u_N) \; da, \tag{6.36}$$

where by Bayes' rule,

$$f_{A|U_1,...,U_N}(a|u_1, ..., u_N) = \frac{\prod_{i=1}^N \Pr[U_i = u_i|a]}{\int_0^L \prod_{i=1}^N \Pr[U_i = u_i|a] \; da},$$

with

$$\Pr[U_i = u_i|a] = \begin{cases} \Phi\left(\frac{T_i - a}{\sigma}\right) & u_i = 1 \\ \Phi\left(\frac{a - T_i}{\sigma}\right) & u_i = 2 \end{cases}.$$

For fixed N, we can minimize $J(N, \bar{T})$ over the N-dimensional $\bar{T}$-space, and thereby find the optimal breakpoint quantizer strategy.

Unfortunately, the N-dimensional minimization of $J(N, \bar{T})$ is not at all trivial. One difficulty is that the function is not convex, and thus, there is no guarantee that a local minimum is also the global minimum. This same difficulty is familiar from the hypothesis testing problem, where there is no guarantee that a pbp-optimal strategy is also globally optimal. This difficulty, in fact, is common to minimization of all non-convex functions. A perhaps more serious difficulty is that every evaluation of $J(\gamma, \bar{T})$ is extremely (computationally) expensive.

To ease the computational burden, we reduce the dimension of the minimization problem. We do this dimension reduction by imposing symmetry on $\bar{T}$. We consider two different extremes of symmetry. First, we reduce the dimension of the problem to one, by requiring that all sensors use the same threshold. Then, we consider a less extreme dimension reduction, to a dimension of $\lfloor \frac{N}{2} \rfloor$, by requiring that the thresholds be symmetric about $L/2$. Naturally, this dimension reduction will adversely affect the performance of the resulting strategy, but the sacrifice in performance will be offset by the reduction in computational complexity.

Here is an outline of what follows. First, we describe the minimization technique that we use to search over $\bar{T}$-space for a strategy that minimizes $J(N, \bar{T})$. Then, we numerically experiment with the technique.

6.8.1 Powell's Multidimensional Direction Set Method

There are standard numerical techniques for finding local minima of multivariable functions. The most efficient techniques use the gradient of the function. Unfortunately, the gradient of $J(N, \bar{T})$ is analytically intractable. Thus, we have two options:

1. Numerically estimate the gradient, and use this estimate in one of the standard minimization techniques that requires the gradient, or

2. Use a completely different technique that does not need gradient information at all.

Brent [B73] has suggested that the first of these options is aesthetically unpleasing. His argument is that when the gradient information is not analytically available, it follows that using technique 1 is akin to "crowbaring" the numerical minimization technique into a form that is usable on the given function. We agree with Brent's contention, and so we follow the second of the above options. In the following section, we describe Powell's direction set method [PFTV88], which is an improved version of Brent's direction set method [B73].

Powell's method is described quite lucidly in [PFTV88]. The following is a brief summary of that discussion.

The key to the method is to break the minimization problem down into a sequence of one-dimensional minimizations along intelligently-chosen directions in the N-dimensional $\bar{T}$-space. Each of these one-dimensional minimizations is called a *line minimization*. Formally, each line minimization proceeds as follows:

We have scalar-valued function $g(\bar{x})$ of N-dimensional variable $\bar{x}$. Given a starting point, $\bar{P}$, and direction, $\bar{\theta}$, both in N-space, we find the value of the scalar λ that minimizes $g(\bar{P} + \lambda\bar{\theta})$. Once this value is found, the line minimization is complete.

It is quite straightforward to carry out each of the one-dimensional minimizations, and so we direct the reader to [PFTV88] for a discussion of that issue. The more challenging issue is to find good directions, $\hat{\theta}$, for the line-minimizations.

A naive choice is to use the principal coordinate directions $\bar{e}_1$, ..., $\bar{e}_N$ for the line minimizations, where $\bar{e}_1 = [1\ 0\ ...\ 0]^T$, etc. This is equivalent to a sort of modified Gauss-Seidel algorithm. To see the similarity, suppose that we wished to update the rule for the first peripheral sensor.

A Gauss-Seidel approach holds all other peripheral sensor rules fixed, and also holds the fusion rule fixed. With respect to all of these, it finds the pbp-optimal rule for the first sensor. Note that there is no guarantee that the pbp-optimal rule will be a breakpoint quantizer rule, and, in fact, quite likely the rule will NOT be a breakpoint quantizer rule.

The line-minimization approach again holds all of the other peripheral sensor rules fixed. However, the fusion rule is now simultaneously updated along with the first peripheral sensor's rule. Also, the peripheral sensor rule is not updated to the pbp-optimal rule, but rather, to the optimal breakpoint quantizer rule. Thus, there is a tradeoff here. On the one hand, simultaneously updating the fusion rule and the peripheral sensor rule is more computationally expensive than merely updating the peripheral sensor rule. This favors the Gauss-Seidel approach. On the other hand, constraining the rule to always be a breakpoint quantizer keeps the structure of the strategy very simple. This favors the line-minimization technique.

Minimizing along the principal coordinate directions actually works quite well for some functions, but can be disastrously slow for others. The reasons for this are discussed in [PFTV88].

Powell suggested a fairly straightforward algorithm for choosing the directions of the line minimizations. In general, his method works better than does the method that uses the principal coordinate directions.

Here is the basic idea:

1. Let

$$\bar{Q}_i = \bar{e}_i, \quad 1 \leq i \leq N$$

be the set of initializing directions for the line minimizations. In this expression, the vectors $\bar{e}_i$ are the directions of the principal axes.

2. Start at some initializing point, $\bar{P}_0$. As with initializations of all other algorithms in this thesis, there is no optimal algorithm for selecting the initializing point.

3. For $i = 1$, ..., N, do a line minimization, starting at $\bar{P}_{i-1}$, and moving in the direction $\bar{Q}_i$. Let $\bar{P}_i = \bar{P}_{i-1} + \lambda\bar{Q}_i$, where λ is the minimizing scalar in the line minimization. Keep track of $(\Delta g)_i$, where $(\Delta g)_i = g(\bar{P}_{i-1}) - g(\bar{P}_i)$.

4. Replace the $\bar{Q}_i$ corresponding to the maximum $(\Delta g)_i$ with $\bar{P}_N - \bar{P}_0$; keep all of the other line minimization directions the same. Now, let $\bar{P}_0 = \bar{p}_N$ (thereby creating an an improved initializing point), and return to step 1.

For an explanation of why this works, refer to [PFTV88] or to [A70].

To express the computational complexity of searching for the minimizer of $J(N, \bar{T})$, we let a unit of computation be the calculation of a single expectation (6.36). Then, computing $J(N, \bar{T})$ requires $O(2^N)$ effort for each value of $\bar{T}$. The minimization of $J(N, \bar{T})$ requires $O(N^2)$ line minimizations, and each line minimization requires $O(1)$ function evaluations. Hence, the overall complexity is $O(N^2 2^N)$ function evaluations.

This complexity is overwhelming for values of $N > 5$. For this heuristic to be practical, we need further computational complexity reductions.

6.8.2 Reduction of Complexity: Single Breakpoint

We can most drastically reduce the computational complexity by requiring that all of the peripheral sensors use the same threshold (breakpoint). Symmetry then simplifies the fusion rule to

$$\gamma_0(u_1, ..., u_N) = \gamma_0(k), \quad k = \sum_{i=1}^{N} u_i,$$

where

$$\gamma_0(k) = \int_0^L a \, f_{A|K}(a|k) \, da,$$

with

$$f_{A|K}(a|k) = \frac{\Pr[K = k|a]}{\int_0^L \Pr[K = k|a] \, da}, \tag{6.37}$$

and

$$\Pr[K = k|a] = \binom{N}{k} \left[\Phi\left(\frac{a - T}{\sigma}\right)\right]^k \left[1 - \Phi\left(\frac{a - T}{\sigma}\right)\right]^{N-k}. \tag{6.38}$$

Mean-Square Error

The mean-square error of this strategy is completely determined by N and the scalar T:

$$J(N, T) = \sum_{k=0}^{N} \frac{1}{L} \int_0^L da \, [\gamma_0(k) - a]^2 \binom{N}{k} \left[\Phi\left(\frac{a - T}{\sigma}\right)\right]^k \left[1 - \Phi\left(\frac{a - T}{\sigma}\right)\right]^{N-k} \tag{6.39}$$

With respect to T, we can routinely do the one-dimensional optimization of $J(N, T)$. However, without doing any calculation at all, we might intuitively expect that the optimal T will be $L/2$. This hunch is partially confirmed by the relation

$$\frac{\partial}{\partial T} J(N, T)|_{T=L/2} = 0. \tag{6.40}$$

To show (6.40), first note that (6.37) and (6.38) imply that

$$T = L/2 \implies f_{A|k}(a|k) = f_{A|k}(L-a|N-k) \implies \gamma_0(k) = L - \gamma_0(N-k) \quad (6.41)$$

Now, for notational convenience, we let

$$J'_k(N,T) = \frac{\partial}{\partial T}\int_0^L da\,[\gamma_0(k)-a]^2\left[\Phi\left(\frac{a-T}{\sigma}\right)\right]^k\left[1-\Phi\left(\frac{a-T}{\sigma}\right)\right]^{N-k}$$

so that

$$\frac{\partial}{\partial T}J(N,T) = \frac{1}{L}\sum_{k=0}^{N}\binom{N}{k}J'_k(N,T). \quad (6.42)$$

We can now use (6.41) to show that

$$T = L/2 \implies J'_k(N,T) = -J'_{N-k}(N,T). \quad (6.43)$$

To prove this for the case of $k = 0$, note that

$$\begin{aligned} J'_0(N,T) &= \int_0^L da[\gamma_0(0)-a]^2 N\Phi^{N-1}(L/2-a)\frac{e^{-(L/2-a)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \\ &= \int_0^L da[\gamma_0(N)-a]^2 N\Phi^{N-1}(a-L/2)\frac{e^{-(L/2-a)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \\ &= -J'_N(N,T). \end{aligned}$$

The verification for other k pairs is similar, but more tedious; we omit the details.

Finally, substituting (6.43) into (6.42) yields the result that

$$\frac{\partial}{\partial T}J(N,T) = 0 \quad (6.44)$$

A review of the above analysis will show that it heavily depends on symmetry. There is no apparent way to analytically find the derivative of $J(N,T)$ (with respect to T) at any threshold other than $T = L/2$. Thus, it is unknown (in general) whether $J(N,T)$ could have a derivative of zero at other thresholds. Hence, we have NOT shown that $T = L/2$ is the optimal threshold; we have only shown that it is a strong candidate for being the optimal threshold.

In Figure 6.3, we show that, indeed, $T = L/2$ can be the optimal threshold.

Asymptotic Considerations

It would be nice if we could somehow derive an analytic asymptotic form for $J(N, T_{opt})$ as $N \to \infty$. Unfortunately, there is no apparent way to do this; however, we do have the following weaker result:

Proposition 19 *For the breakpoint quantizer strategy that uses a single breakpoint (threshold),*

$$\lim_{N\to\infty} J(N, T_{opt}) = 0. \quad (6.45)$$

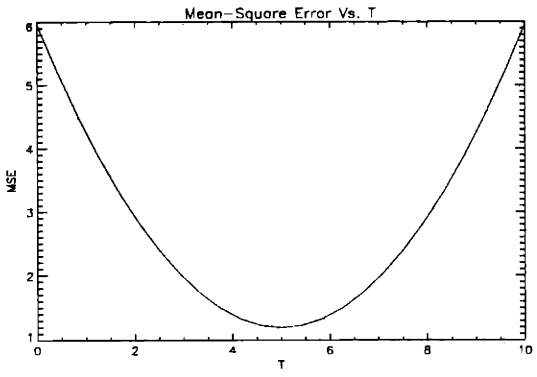

Figure 6.3: Design 2 (single threshold case) dependence of mean-square estimation error on T when $L = 10, \sigma = 1, N = 5$. Clearly $T_{opt} = L/2$.

Proof: We first recall that (6.37) yields a lower MSE than any other fusion rule. Now consider the alternative fusion rule

$$\gamma_0'(k) = T_{opt} + \sigma\Phi^{-1}\left(\frac{k}{N}\right). \tag{6.46}$$

This fusion rule is motivated by the Strong Law of Large Numbers, which says that

$$\begin{aligned} &\lim_{N\to\infty} \quad \frac{k}{N} = \Phi\left(\frac{a - T_{opt}}{\sigma}\right) \quad \text{w.p. } 1 \quad \Longrightarrow \\ &\lim_{N\to\infty} \quad T_{opt} + \sigma\Phi^{-1}\left(\frac{k}{N}\right) = a \quad \text{w.p. } 1. \end{aligned} \tag{6.47}$$

Hence, the MSE of this alternative fusion rule, $J'(N, T_{opt})$, goes to zero; since $J(N, T_{opt}) \leq J'(N, T_{opt})$, the result follows. **QED**.

Results

Figure 6.4 is a plot of $J(N, T = L/2)$ as a function of N. Note that according to the analytic curve that was fit to this MSE, the limiting MSE is 0.3. Since Proposition 19 says that the MSE converges to zero, we conclude that the analytic fit can only have a limited range of validity. Nevertheless, for the range of N plotted, the fit is quite good, and it shows that the single-breakpoint strategy does NOT have $O(\frac{1}{N})$ MSE decay (for the range of N plotted).

6.8.3 Reduction of Complexity: Symmetric Breakpoint Strategy

A less extreme reduction in complexity comes from constraining the thresholds to be symmetric about $L/2$. For example, suppose $L = 10$, and $T_1 = 1, T_2 = 4$; if $N = 5$, then we only have 2 degrees of freedom in selecting the thresholds, and $T_3 = 5, T_4 = 6, T_5 = 9$. This is symmetry constraint has the aesthetically

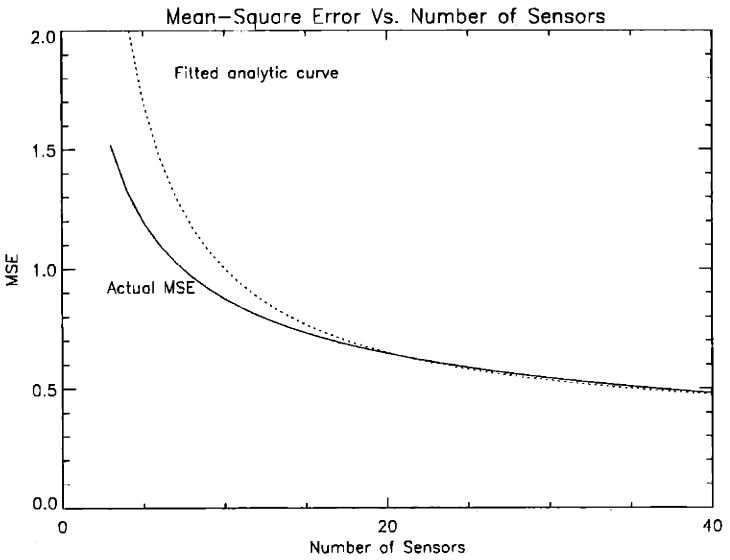

Figure 6.4: Design 2 (single threshold case) mean-square estimation error as a function of number of sensors. The overlay dotted curve is $0.30 + \frac{7.00}{N}$.

Number of Sensors	Indicator (Section 6.7)	Single Threshold (Section 6.8.2)	Multiple Thresholds (Section 6.8.3)
3	1.74	1.52	1.17
5	1.05	1.20	0.78
7	0.75	1.03	0.59
9	0.59	0.92	0.48

Figure 6.5: Mean-square error of optimal strategy from each of three different design techniques. This is for $L = 10, \sigma = 1$.

pleasing consequence that the estimator is unbiased when the true parameter value is $A = L/2$. With the constraint, we need to search over a $\lfloor N/2 \rfloor$ dimensional space for the minimum of $J(N, \bar{T})$. This search still has complexity $O(N^2 2^N)$, but there is a constant multiplicative reduction to roughly $2^{N/2}$ the computational effort required for N arbitrarily placed thresholds.

Table 6.5 provides an example of how well a symmetric breakpoint strategy can perform.

As one would expect, the multiple breakpoint strategy has the best performance. What the table does not show is the great computational burden required to search for the optimal breakpoint strategy. To find the optimal $N = 7$ strategy ($T_1 = T_2 = 2.5$, $T_3 = 3.7$, $T_4 = 5$, $T_5 = 6.3$, $T_6 = T_7 = 7.5$) took 1.5 hours of CPU time on a VAX workstation. To find the optimal $N = 9$ strategy ($T_1 = T_2 = T_3 = 2.45$, $T_4 = 4.4$, $T_5 = 5$, $T_6 = 5.6$, $T_7 = T_8 = T_9 = 7.55$) took 17 hours of CPU time. Clearly, there is a strong law of diminishing returns at work here.

6.9 Design 3: Quantizer

Now we consider designing a strategy in which each of the N peripheral sensors tries to determine a different bit of an N-bit quantized version of A. In this way, we break the parameter estimation problem into a collection of binary hypothesis testing problems (*is the bit high or low?*). The fusion center then uses the N bits to reconstruct the quantized version of A.

To coordinate the strategy, we define

$$z_k = (2k+1)\frac{L}{2^{N+1}}, \quad 0 \le k \le 2^N - 1.$$

Note that these z values are evenly spaced over the valid parameter range. Peripheral sensor i is given the hypothesis testing problem

$$\begin{aligned} H_1 : &\quad \left\lfloor \tfrac{k}{2^{i-1}} \right\rfloor \quad \text{is even,} \\ H_2 : &\quad \left\lfloor \tfrac{k}{2^{i-1}} \right\rfloor \quad \text{is odd,} \end{aligned} \tag{6.48}$$

where K is the random variable defined by

$$K \;=\; \arg\min_i \; |A - z_i|, .$$

Thus, peripheral sensor 1 is trying to determine the value of the lowest bit, peripheral sensor N is trying to determine the value of the highest bit, etc.

Minimizing the probability of error in the bit decisions of the peripheral sensors is a difficult task. The difficulty is that the two hypotheses are not conditionally independent (and for that matter, they are not even approximately conditionally independent). A reasonable decision rule to try is

$$\gamma_i(y_i) \;=\; \begin{cases} 0 & \text{if } \left\lfloor \tfrac{k'(y_i)}{2^{i-1}} \right\rfloor \text{ even} \\ 1 & \text{otherwise} \end{cases}, \tag{6.49}$$

where

$$k'(y_i) \;=\; \arg\min_k \; |y_i - z_k|, .$$

One can readily show that in the absence of noise (i.e., $Y_i = a, \forall i$) the strategy using (6.49) and the fusion rule

$$\gamma_0(u_1, ..., u_N) = z_{u_0} \quad \text{with} \quad u_0 = \sum_{i=1}^{N} (u_i)\, 2^{i-1}.$$

has a mean-square error of

$$\mathrm{E}[(A - \gamma_0(U_1, ..., U_N))^2] = \frac{1}{12}(2^{-N}L)^2.$$

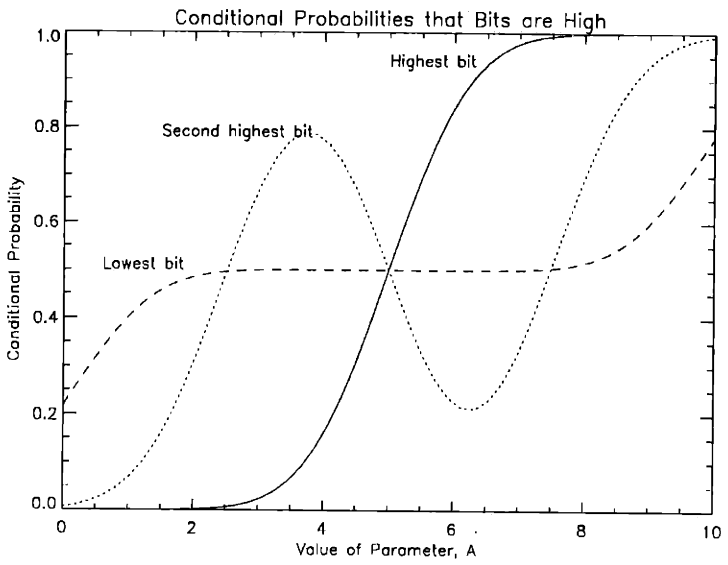

Figure 6.6: $\Pr[U_i = 1|a]$ for $N = 5$, when the rule in (6.49) is used. The solid line is the highest bit ($i = 5$), the dashed line is the second highest bit ($i = 4$), and the dotted line is the lowest order bit ($i = 1$).

Since this error decreases exponentially with N, one could speculate that even with noise in the observations Y_i, the error might still be tolerable.

Figure 6.6 conveys the unfortunate problem with the conjecture. Note that for the lowest order bit, $\Pr[U_1 = 1|a] \approx 1/2$, independent of A. Thus, with the decision rule (6.49), peripheral sensor 1 achieves $P_{\text{errror}} \approx 1/2$ on its hypothesis testing problem. As N increases, this same difficulty will be faced by all the three (or so) peripheral sensors that are working on the highest three bits. Thus, increasing N does not buy additional decreases in the MSE.

This discussion would seem to indicate impossibility of the hypothesis testing problem faced by sensors solving for the lower-order bits. One workaround would be to coordinate many peripheral sensors to work on a single low-order bit. For example, a group of sensors could use the methods of Section 6.7 to find a, and then to classify the bit. But using this method to classify the bit means that that the hypothesis testing problem corresponding to that bit is as difficult as solving for a itself. In other words, we have gained nothing by breaking the parameter estimation problem into a series of binary hypothesis testing problems, because some of these constituent problems are as difficult as the original parameter estimation problem. In conclusion, we abandon this quantization heuristic as not very useful.

6.10 Multiple Bits Per Sensor

6.10.1 Generalization of Design 1: Indicator Functions Revisited

We now consider a multiple-bit/sensor generalization of the strategy considered in Section 6.7. This generalization is, we believe, new. Again, the peripheral sensors use breakpoint quantizer rules, but now, since each sensor can transmit $k, (k \geq 1)$ bits we can have $2^k - 1$ thresholds per sensor. The i-th threshold of sensor j, T_{ij} is defined by

$$\begin{aligned} T_{ij} &= T_{11} + [(i-1) + (2^k - 1)(j-1)]\Delta, \\ T_{11} &= \frac{L}{2} - \frac{(2^k - 1)N - 1}{2}\Delta. \end{aligned} \tag{6.50}$$

Thus, as in the single bit/sensor case, the thresholds are evenly spaced about $L/2$ in increments of Δ.

We have two types of peripheral sensor decision rules. If for sensor j, $T_{ij} < 0, \forall i$ or $T_{ij} \geq 0, \forall i$ (that is, if the thresholds for sensor j do not straddle zero), then

$$\gamma_j(y_j) = \begin{cases} 0, & y_j < T_{1j}, \\ i, & T_{ij} \leq y_j \leq T_{(i+1)j}, \quad 1 \leq i \leq 2^k - 2, \\ 2^k - 1, & y_j \geq T_{(2^k-1)j}, \end{cases} . \tag{6.51}$$

If, on the other hand, the thresholds for sensor j straddle zero, then we use the decision rule

$$\gamma_j(y_j) = \begin{cases} -m+1, & y_j < T_{1j}, \\ -m+i+1, & T_{ij} \leq y_j < T_{(i+1)j}, \quad 1 \leq i \leq 2^k - 2, \\ 2^k - m, & y_j \geq T_{(2^k-1)j}, \end{cases} , \tag{6.52}$$

where

$$m = \arg\min_i T_{ij} \geq 0. \tag{6.53}$$

Clearly, at most one sensor will use this second type of rule, while all of the other sensors will use the first type of rule.

For the fusion center, we use the rule

$$\gamma_0(u_1, ..., u_N) = \Delta \left[\sum_{i \in T^+} u_i \right] - \Delta \left[\sum_{i \in T^-} (2^k - 1 - u_i) \right] + \Delta u_{\text{straddle}}, \tag{6.54}$$

where T^+ is the set of peripheral sensors for which $T_{ij} \geq 0, \forall i$ and T^- is the set of peripheral sensors for which $T_{ij} < 0, \forall i$. Also, u_{straddle} correpsonds to the peripheral sensor whose thresholds straddle zero; if there is no such straddler, then $u_{\text{straddle}} = 0$.

6.10.2 Asymptotic Bias

Proposition 20 *For* $\Delta = 1/\sqrt{(2^k - 1)N}$, *we have*

$$\lim_{N\to\infty} E[\gamma_0(u_1, ..., u_N)|a] = a. \tag{6.55}$$

meaning that the strategy is asymptotically unbiased.

Proof: The proof is a simple modification of the proof of unbiasedness in Proposition 18. For simplicity, we assume that no sensor has thresholds that straddle zero.

$$\begin{aligned}
&\lim_{N\to\infty} E[\gamma_0(u_1, ..., u_N)|a] \\
&= \Delta\left[\sum_{i\in T^+} E(u_i)\right] - \Delta\left[\sum_{i\in T^-} 2^k - 1 - E(u_i)\right] \\
&= \Delta\left[\sum_{i\in T^+}\sum_{j=1}^{2^k-1} E(I_{ji})\right] - \Delta\left[\sum_{i\in T^-}\sum_{j=1}^{2^k-1} 1 - E(I_{ji})\right] \\
&= \Delta\left[\sum_{i\in T^+}\sum_{j=1}^{2^k-1} \Pr(y_i \geq T_{ji}|a)\right] - \Delta\left[\sum_{i\in T^-}\sum_{j=1}^{2^k-1} \Pr(y_i < T_{ji}|a)\right] \\
&\underset{N\to\infty}{\to} \left[\int_0^\infty \Phi\left(\frac{a-x}{\sigma}\right)\, dx\right] - \left[\int_{-\infty}^0 \Phi\left(\frac{x-a}{\sigma}\right)\, dx\right] \\
&= a,
\end{aligned} \tag{6.56}$$

where I_{ji} is an indicator function: $I_{ji} = 1$ iff $y_i \geq T_{ji}$.

This result motivates our interest in testing the algorithm for the finite N case.

6.10.3 Tradeoff Between Number of Sensors and Number of Bits Per Sensor

In Table 6.7, we show the performance tradeoff between N (number of peripheral sensors) and D (message alphabet size per sensor), for the multiple-breakpoint technique introduced in the last section. In particular, when the product DN is held constant, what (positive integer) values for N and D yield the lowest mean-square error?

The table indicates that for this particular problem, it is better to use as many sensors as possible (and thus, to use $D = 2$ at each sensor). This is an intuitively satisfying result. The reason that it is satisfying relates to the conditional independence of the sensor observations. Roughly speaking, by "spreading" the sensor observations over as many sensors as possible, the Gaussian corruption tends to even out more than if, say, only two sensors receive observations.

Number of Sensors	Number of Bits	Mean-Square Error
12	1	0.445
6	2	0.589
4	3	0.684
3	4	0.746
20	1	0.272
10	2	0.416
5	4	0.609

Figure 6.7: Tradeoff between number of sensors and number of bits per sensor. This is for $L = 10, \sigma = 1$.

Bibliography

[AS64] Abromowitz, M. and Stegun, I., *Handbook of Mathematical Functions*, National Bureau of Standards, 1964.

[BO78] Bender, C., and Orszag, S., *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, 1978.

[CV86] Chair, Z., and Varshney, P.K., "Optimum Data Fusion in Multiple Sensor Detection Systems", *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, 1986, pp. 98-101.

[FG87] Flynn, T., Gray, R., "Encoding of Correlated Observations", *IEEE Transactions on Information Theory*, Vol. IT-33, No.6, November, 1987.

[GBL85] Gray, R.M., Boyd, S., Lookabaugh, T., "Low rate distributed quantization of noisy observations", Proceedings of Allterton Conference, Monticello, IL, 1985, pp.354-358.

[LS82] Lauer, G.S., Sandell, N.R. Jr., "Distributed Detection with Waveform Observations: Correlated Observation Processes", *Proceedings of the 1982 American Control Conference*, vol. 2, pp.812-819, 1982.

[LLG90] Longo, M., Lookabaugh, T., Gray, R., "Quantization for Decentralized Hypothesis Testing under Communication Constraints", *IEEE Transactions on Information Theory*, Vol.36, No.2, March, 1990.

[PA90] Papastavrou, J.D. and Athans, M., "On Optimal Distributed Decision Architectures in a Hypothesis Testing Environment", *preprint* submitted to *IEEE Transactions on Automatic Control*, January, 1990.

[PFTV88] Press, W., Flannery, B., Teukolsky, S., Vetterling, W., *Numerical Recipes in C*, Cambridge University Press, 1988.

[S86] Sadjadi, F., "Hypothesis Testing in a Distributed Environment", *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, 1986, pp. 134-137.

[SGB67] Shannon, C.E., Gallager, R.G., Berlekamp, E.R., "Lower bounds to error probability for coding on discrete memoryless channels, I", *Information and Control*, 10, 1967, pp.65-103.

[SR83] Sarma, V.V.S., Rao, K.A.G., "Decentralized Detection and Estimation in Distributed Sensor Systems", in *Proceedings of the IEEE 1983 Cybernetics and Society Conference, 1*, pp. 438-441.

[T88] Tsitsiklis, J.N., "Decentralized Detection by a Large Number of Sensors", *Mathematics of Control, Signals, and Systems*, 1, 1988, pp.167-182.

[T89] Tsitsiklis, J.N., "Decentralized Detection", preprint, 1989.

[T89a] Tsitsiklis, J.N., "Extremal Properties of Likelihood Ratio Quantizers", LIDS-P-1923, November, 1989.

[T91] Tsitsiklis, J.N., Private Communication.

[TA85] Tsitsiklis, J.N., Athans, M., "On the Complexity of Decentralized Decision Making and Detection Problems", *IEEE Transactions on Automatic Control*, AC-30, 5, 1985, pp. 440-446.

[TPK89] Tang, Z.B., Pattipati, K.R., and Kleinman, D.L., "A Distributed M-ary Hypothesis Testing Problem with Correlated Observations", preprint, 1989.

[TS81] Tenney, R.R., and Sandell, N.R. Jr., "Detection with Distributed Sensors", *IEEE Transactions on Aerospace and Electronic Systems*, AES-17, 1981, pp. 501-510.

[VT68] Van Trees, H.L., *Detection Estimation and Modulation Theory*, Vol.I, J.Wiley, New York, 1968.