The Informational Complexity of Learning from Examples

by

Partha Niyogi

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1995

© Massachusetts Institute of Technology 1995

Signature of Author ....
Department of Electrical Engineering and Computer Science
January 30, 1995

Certified by ....
Tomaso Poggio
Professor of Brain and Cognitive Science
Thesis Supervisor

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

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

APR 13 1995 Eng.

The Informational Complexity of Learning from Examples

by

Partha Niyogi

Submitted to the Department of Electrical Engineering and Computer Science
on January 30, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis examines the problem of learning unknown target functions from examples. In particular, we focus on the informational complexity of learning these classes, i.e., the number of examples needed in order to identify the target with high accuracy and great confidence. There are a number of factors affecting the informational complexity, and we attempt to tease them apart in different settings, some of which are cognitively relevant.

1) We consider a wide class of pattern classification and regression schemes known as regularization networks. We investigate the number of parameters and the number of examples that we need in order to achieve a certain generalization error with prescribed cofidence. We show that the generalization error is due in part to the representational inadequacy (finite number of parameters) and informational inadequacy (finite number of examples), and bound each of these two contributions. In doing so, we characterize a) the inherent tension between these two forms of error: attempting to reduce one, increases the other b) the class of problems effectively solved by regularization networks c) how to choose an appropriately sized network for such a class of problems.

2) Rather than drawing its examples randomly (passively), suppose a learner were allowed to choose its own examples. Does this option allow us to reduce the number of examples? We derive a sequential version of optimal recovery allowing the active learner to adaptively choose points of maximum information. We compare this against the passive case, and classical optimal recovery, indicating superior performance.

3) We investigate the problem of language learning within the principles and parameters framework. We show how certain memoryless algorithms operating on finite parameter spaces can be effectively modeled as a Markov chain. This allows us to characterize the learnability, and sample complexity of such linguistic spaces.

4) We consider a population of learners attempting to learn a target language using some learning algorithm. We derive a dynamical system model (from the grammatical theory and learning paradigm) characterizing the evolving linguistic composition of the population over many generations. We examine the computational and linguistic consequences of this derivation, and show that it allows us to formally pose an evolutionary criterion for the adequacy of linguistic theories.

Thesis Supervisor: Tomaso Poggio
Title: Professor of Brain and Cognitive Science

Acknowledgments

My debts are many. My advisor, Tommy Poggio, supported me throughout, provided complete freedom to work on problems of my choosing, and reassured me of the usefulness of theoretical analyses. Bob Berwick introduced me to the fascinating world of modern linguistic theory, helped me to make contacts with the linguistics department, and provided advice, encouragement, and friendship. Chapters 4 and 5 are joint work with him. Federico Girosi taught me all I know about approximation theory, was willing to have endless discussions on a variety of subjects, collaborated directly on Chapter 2, and provided useful input on Chapter 3. Many thanks to Vladimir Vapnik and Ron Rivest for agreeing to be members of my committee. Vapnik had several long discussions with me explaining the structure of learning theory, read some of my papers very closely, and profoundly influenced my way of thinking through his celebrated book. Finally, Ron took time out of his busy schedule to read the thesis, had discussions with me at various points, taught me formal aspects of machine learning in his class, and was wonderfully supportive at all times we met.

Several people at MIT and outside have touched my life in many ways. Members of CBCL, especially Kah Kay Sung who worked with me on some problems; the SLS group, my first home, and first friends; various professors at MIT; the "Alliance", including Gaurav, Ranga, Abha, Waheed who spent numerous hours in coffee shops around Cambridge discussing the inequitable nature of the world, and organizing the struggle to change it; Somakda for friendship; Bulbulidi for allowing me to introspect; and Parvati for friendship, affection, and love.

Finally, my family—my parents, and brother Siddhartha; my extended family in Bombay, Delhi, and Calcutta who provided encouragement every step of the way; for innumerable telephone calls, letters, good wishes; for worrying, for caring, and for supporting; this thesis could never have been done without them.

Thesis Committee

Tomaso Poggio (Supervisor; Professor of Brain and Cognitive Science)
Robert C. Berwick (Professor of Computer Science)
Ronald L. Rivest (Professor of Computer Science)
Vladimir Vapnik (Member, Technical Staff; Bell Labs)
Federico Girosi (Principal Research Scientist)

Contents

List of Figures

Chapter 1

Introduction

Abstract

We introduce the framework in which learning from examples is to be studied. We develop a precise notion of informational complexity and discuss the factors upon which this depends. Finally, we provide an outline of the four problems discussed in this thesis, our major contributions, and their implications.

Learning is the centerpiece of human intelligence. Consequently any attempt to understand intelligence in the human being or to replicate it in a machine (as the field of artificial intelligence is committed to doing) must of necessity explain this remarkable ability. Indeed a significant amount of effort and initiative has gone into this enterprise and a collective wisdom has emerged regarding the paradigms in which this study is to be conducted.

Needless to say, learning can mean a variety of things. The ability to learn a language, to recognize objects, to manipulate them and navigate through them, to learn to play chess or to learn the theorems of geometry all touch upon different sectors of this multifaceted activity. They require different skills, operate on different spaces and use different procedures. This has naturally led to a spate of learning paradigms; but most share one thing in common, i.e., learning as opposed to "preprogrammed" or memorized behavior involves the updating of hypotheses on the basis of some kind of experience: an adaptation if you will to the environment on the basis of stimuli from it. The connection to complex adaptive systems springs to mind and later in this thesis we will make this connection more explicit in a specific context.

How then does one begin to study such a multifaceted problem? In order to meaningfully define the scope of our investigations, let us begin by considering a formulation by Osherson et al (1986). They believe (as do we) that learning typically involves

1. A learner

2. A thing to be learned.

3. An environment in which the thing to be learned is presented to the learner.

4. The hypotheses that occur to the learner about the thing to be learned on the basis of the environment.

Language acquisition by children is a classic example which fits well into this framework. "Children are the learners; a natural language is the thing to be learned; the corpus of sentences available to the child is the relevant environment; grammars serve as hypotheses." (from *Systems that Learn*; Osherson et al 1986). In contrast, consider an example from machine learning; the task of object recognition by the computer. Here the computer (or the corresponding algorithm) is the learner, the identity of objects (like chairs or tables, for example) are the things to be learned, examples of these objects in the form of images are the relevant environment, and the hypotheses might be decision boundaries which can be computed by a neural network.

In this thesis we will concern ourselves with learning input-output mappings from examples of these mappings; in other words, learning target functions which are assumed to belong to some class of functions. The view of the brain as an information processor (see Marr, 1982) suggests that in solving certain problems (like object recognition, for example) the brain develops a series of internal representations starting with the sensory (external) input; in other words, it computes a function. In some cases, this function is hardwired (like detecting the orientations of edges in an image, for example), in others the function is learned like learning to recognize individual faces.[1] As another example of an input-output function the brain has to compute, consider the problem of speech recognition. The listener is provided with an acoustic signal which corresponds to some underlying sentence, i.e., a sequence of phonetic (or something quite like it) categories. Clearly the listener is able to uncover the transformation from this acoustic space to the lexical space. Note also that this transformation appears to be different for different languages, i.e., different languages have different inventories of phonetic symbols. Further, they carve up the acoustic space in different ways; this accounts for why the same acoustic stimuli might be perceived differently as belonging to different phonetic categories by a native speaker

[1] Functions mapping images of faces to the identity of the person possessing them may of course themselves be composed of more primitive functions, like edge detectors, which are hardwired. There is a considerable body of literature devoted to identifying the hardwired and learned components of this entire process from a neurobiological perspective. The purpose of this example was merely to observe that the brain appears to learn functions of various kinds; consequently studying the complexity of learning functions is of some value.

of Bengali and a native speaker of English. Since children are not genetically predisposed to learn Bengali as opposed to English (or vice versa) one might conclude that the precise nature of this transformation is learned.

Not all the functions we consider in this thesis can be psychologically well-motivated; while some chapters of this thesis deal with languages and grammars which are linguistically well motivated, Chapter 2, which concentrates in large part on Sobolev spaces, can hardly seem to be interesting psychologically. However, the central strand running through this thesis is the informational complexity of learning from examples. In other words, if information is provided to the learner about the target function in some fashion, how much information is needed for the learner to learn the target well? In the task of learning from examples, (examples, as we shall see later are really often nothing more than $(x, y = f(x))$ pairs where $(x, y) \in X \times Y$ and $f : X \longrightarrow Y$) how many examples does the learner need to see? This same question is asked of strikingly different classes of functions: Sobolev spaces and context free languages. Certain broad patterns emerge. Clearly the number of examples depend upon the algorithm used by the learner to choose its hypotheses, the complexity of the class from which these hypotheses are chosen, the amount and type of noise and so on. We will try in this thesis to tease apart the relative contributions of each in specific settings in order to uncover fundamental constraints and relationships between oracle and learner; constraints which have to be obeyed by nature and human in the process of living.[2]

This then is our point of view. Let us now discuss some of the relevant issues in turn, briefly evaluate their importance in a learning paradigm, and the conceptual role they have to play in this thesis.

1.1 The Components of a Learning Paradigm

1.1.1 Concepts, Hypotheses, and Learners

Concept Classes

We need to define the "things" to be learned. In order to do this, we typically assume the existence of identifiable entities (concepts) which are to be learned and which belong perhaps to some set or class of entities (the concept class). Notationally, we can refer to the concept class by $\mathcal{C}$ which is a set of concepts $c \in \mathcal{C}$. These concepts

[2] Even if we are totally unconcerned with human learning and are interested only in designing machines or algorithms which can learn functions from examples, a hotly pursued subject in machine learning, the issue of number of examples is obviously of considerable importance

need to be described somehow and various representation schemes can be used. For example, researchers have investigated concept classes which can be expressed as predicates in some logical system (Michalski, Carbonell, and Mitchell; 1986). For our purposes we concentrate on classes of functions, i.e., our concept classes are collections of functions from X to Y where X and Y are sets. We will define the specific nature of these functions over the course of this thesis.

Information Sources

Information is presented to the learner about a target concept $c \in \mathcal{C}$ in some fashion. There is a huge space of possibilities ranging from a "divine" oracle simply enlightening the learner with the true target concept in one fell sweep to adversarial oracles which provide information in a miserly, deliberately malicious fashion. We have already restricted our inquiry to studying the acquisition of function classes. A natural and well studied form of information transmission is to allow the learner access to an oracle which provides (x, y) pairs or "labelled examples" perhaps tinged with noise. In a variant of the face recognition problem (Brunelli and Poggio, 1992; where one is required to identify the gender of some unknown person), for example, labelled examples might simply be (image,gender) pairs. On the basis of these examples then, the learner attempts to infer the target function.

We consider several variants to this theme. For example, in Chapter 2, we allow the learner access to (x, y) pairs drawn according to a fixed unknown arbitrary probability distribution on some space $X \times Y$. This represents a passive learner who is at the mercy of the unknown probability distribution, which could, in principle provide unrepresentative data with high probability. In Chapter 3 we explore the possibility of reconstructing functions by allowing the learner to choose his or her own examples, i.e., an active collector rather than a passive recipient of examples. This is studied in the context of trying to learn functional mappings of various sorts. Mathematically, there are connections to adaptive approximation, a somewhat poorly studied problem. Active learning (as we choose to call it) is inspired by various strategies of selective attention that the human brain develops to solve some cognitive tasks. In Chapters 4 and 5 which concentrate on learning the class of natural languages, the examples are sentences spoken by speakers of the target language. We assume again that these sentences are spoken according to a probability distribution on all the possible sentences; there are two further twists: 1) no negative examples occur and 2) typically a bound on the length of the sentences is observed. In all these cases, the underlying question of interest is: given the scheme of presenting examples to the learner, how many examples does the learner need to see to learn well? This question will be sharpened as we progress.

The Learner and Its Hypotheses

The learner operates with a set of hypotheses about reality. As information is presented to it, it updates its hypothesis, or chooses[3] among a set of alternate hypotheses on the basis of the experience (evidence, data depending upon your paradigm of thinking). Clearly then, the learner is mapping its data onto a "best" hypothesis which it chooses in some sense from a set of hypotheses (which we can now call the hypothesis class, $\mathcal{H}$). This broad principle has found instantiations in many differing forms in diverse disciplines.

Consider an example chosen from the world of finance. A stockbroker might wish to invest a certain amount of money on stock. Given the variation of share values over the past few years (a time series) and given his or her knowledge or understanding of the way the market and its players operate, he or she might choose to invest in a particular company. As the market and the share prices unfold, he (or she) might vary the investments (buying and selling stock) or updating the hypotheses. Cumulative experience then might "teach" him/her (or in other words, he/she might "learn") to play this game well.

Or consider another mini-example from speech recognition (specifically phonetic recognition) mapping data to hypotheses. Among other things, the human learner has to discriminate between the sounds /**s**/ and /**sh**/. He or she learns to to do this by being exposed to examples (instances) of each phoneme. Over the course of time, after exposure to several examples, the learner develops a perceptual decision boundary to separate /**s**/ sounds from /**sh**/ sounds in the acoustic domain. Such a decision boundary is clearly learned; it marginally differs from person to person as evidenced by differing responses humans might have when asked to classify a particular sound into one of the two categories. This decision boundary, h, can be considered to be the learner's hypothesis of the **s/sh** distinction (which he or she might in principle pick from a class of possible decision boundaries $\mathcal{H}$ on the basis of the data).

As a matter of fact, the scientific enterprise itself consists of the development of hypotheses about underlying reality. These hypotheses are developed by observing patterns in the physical world and represented as *models, schema* or *theories* which describe these patterns concisely.

[3] In artificial intelligence, this task of "searching" the hypothesis space has been given a lot of attention resulting in a profusion of searching heuristics and characterizations of the computational difficulty of this problem. In this thesis, we ignore this issue for the most part.

If indeed the learner is performing the task of mapping data to hypotheses, it becomes of interest to study the space of algorithms which can perform this task. Needless to say, the operating assumption is that the human learner is also following some algorithm; insights from biology or psychology might help the computer scientist to narrow the space of algorithms and a biologically plausible computational theory (Marr, 1982) might emerge. For our purposes then the learner is an algorithm (or a partial recursive function) from data sets to hypothesis classes.

There is a further important connection between concepts and hypotheses which should be highlighted here. In our scheme of things, concepts are assumed to be the underlying reality; hypotheses are models of this reality. Clearly for successful learning (we discuss learnability in the next section) to occur, the elements of $\mathcal{H}$ should be able to approximate the elements of $\mathcal{C}$, in other words, $\mathcal{H}$ should have sufficient power or complexity to express $\mathcal{C}$. For learnability in the limit (Gold, 1967) or PAC-style (Probably Approximately Correct; Valiant, 1984) models for learnability, this notion can be made more precise. For example, if $\mathcal{C}$ is some class of real valued functions, $\mathcal{H}$ should probably be dense in $\mathcal{C}$.

1.1.2 Generalization, Learnability, Successful learning

In addition to the four points noted earlier, another crucial component of learning is a criterion for success. Formally speaking, one needs to define a metric on the space of hypotheses in order to measure the distance between differing hypotheses, as also between the target concept and the learner's hypothesis. It is only when such a metric is imposed, that one can meaningfully decide whether a learner has "learned" the target concept. There are a number of related notions which might be worthwhile to introduce here.

First, there is the issue of *generalization*. It can be argued, that a key component of learning is not just the development of hypotheses on the basis of finite experience (as experience must be), but the use of those hypotheses to generalize to unseen experience. Clearly successful generalization necessitates the closeness (in some sense) of the learner's hypothesis and the target concept, for it is only then that unseen data (consistent with the target concept) can be successfully modeled by the learner's hypothesis. Thus successful learning would involve successful generalization; this thesis deals with the informational complexity of successful generalization. The *learnability* of concepts implies the existence of algorithms (learners) which can develop hypotheses which would eventually converge to the target. This convergence "in the limit" is analogous to the notion of consistency in statistical estimators and was introduced to the learning community by Gold (1967) and remains popular to this day as a criterion for language learning.

In our case, when learning function classes, $\mathcal{H}$ and $\mathcal{C}$ contain functions from some space X to some space Y, examples are (x, y) pairs consistent with some target function $c \in \mathcal{C}$. Let the learner's hypothesis after m such examples be $h_m \in \mathcal{H}$. According to some pre-decided criterion, we can put a distance metric d on the space of functions to measure the distance between concept and hypothesis (this is our generalization error) $d(h_m, c)$. Learnability in the limit would require $d(h_m, c)$ to go to zero as the number of examples, m, goes to infinity. The sense in which this convergence occurs might depend upon several other assumptions; one might require this convergence to hold for every learning sequence, i.e., for every sequence of examples, or one might want this to be satisfied for almost every sequence in which case one needs to assume some kind of measure on the space according to which one might get convergence in measure (probability).

Convergence in the limit measures only the asymptotic behavior of learning algorithms; they do not characterize behavior with finite data sets. In order to correct for this it is required to characterize the rates of the above-mentioned convergence; roughly speaking how many examples does the learner need to collect so that the generalization error will be small. Again depending upon individual assumptions, there are several ways to formally pose this question. The most popular approach has been to provide a probabilistic formulation; Valiant (1984) does this in his PAC model which has come to play an increasingly important role in computational learning theory. In PAC learning, one typically assumes that examples are drawn according to some unknown probability distribution on $X \times Y$ and presented to the learner. If there exists an algorithm $\mathcal{A}$ which computes hypotheses from data such that for every $\epsilon > 0$ and $0 \leq \delta \leq 1$, $\mathcal{A}$ collects $m(\epsilon, \delta)$ examples and outputs a hypothesis h_m satisfying $d(h_m, c) \leq \epsilon$ with probability greater than $1 - \delta$, then the algorithm is said to PAC-learn the concept c. If the algorithm can PAC-learn every concept in $\mathcal{C}$ then the concept class is said to be PAC-learnable. Looking closely, it can be realized that PAC learnability is essentially the same as weak convergence in probability of hypotheses (estimators) to their target functions with polynomial rates of convergence. In any case, PAC like formulations play a powerful role in characterizing the informational complexity of learning; we have a great intellectual debt to this body of literature and its influence in this thesis cannot be overemphasized.

Remark Sometimes, an obsession with proving the convergence of learning algorithms might be counterproductive. A very good example of that considered in this thesis is the problem of language learning and language change. We need to be able to explain how children learn the language of their social environment on the basis of example sentences. In particular, researchers have postulated algorithms by means of which they can do this; considerable effort has gone into showing that these algorithms successfully converge to the target. However, this does not explain the simultaneously confounding fact that languages change with time. If generation after generation, children successfully converge to the language of their parental generation, then languages would never change. The challenge lies in constructing learning paradigms which can explain both. In our thesis, we demonstrate this by moving into a model for language change by starting out with a model for language learning. The language change model is a dynamical system characterizing the historical evolution of linguistic systems; a formalization of ideas in Lightfoot (1991) and Gell-Mann (1989).

1.1.3 Informational Complexity

We have discussed how the learner chooses hypotheses from $\mathcal{H}$ on the basis of data and how one needs to measure the relative "goodness" of each hypothesis to set a precise criterion for learning. We have also introduced the spirit of the Gold and Valiant formulations of learning and their relationship to the issues of the number of examples and successful generalization. We pause now to comment on some other aspects of this relationship.

First, note that for a particular concept $c \in \mathcal{C}$, given a distance metric d, there exists a best hypothesis in $\mathcal{H}$ given by

$$h_\infty = \arg \min_{h \in \mathcal{H}} d(c, h)$$

Clearly, if $\mathcal{H}$ has sufficient expressive power, then $d(h_\infty, c)$ will be small (precise learnability would actually require it to be 0). If $\mathcal{H}$ is a small class, then $d(c, h_\infty)$ might be large for some $c \in \mathcal{C}$ and even in the case of infinite data, poor generalization will result. This is thus a function of the complexity of the model class $\mathcal{H}$ and how well matched it is to $\mathcal{C}$, a matter discussed earlier as well.

Having established that h_∞ is the best hypothesis the learner can possibly postulate; it is consequently of interest to be able to characterize the convergence of the learner's hypothesis h_m to this best hypothesis as the number of data, m, goes to infinity. The number of examples the learner needs to see before it can choose with high confidence a hypothesis close enough to the best will be our notion of informational complexity. A crucial observation we would like to make is that the number of examples depends (among other things, and we will discuss this soon) upon the size of the class $\mathcal{H}$. To intuitively appreciate this, consider the pathological case of $\mathcal{H}$ consisting of just one hypothesis. In that case, $h_m \in \mathcal{H}$ is always equal to $h_\infty \in \mathcal{H}$

and the learner needs to see no data at all. Of course, the expressive power of such a class $\mathcal{H}$ would be extremely limited. If on the other hand, the class $\mathcal{H}$ is very complex and for a finite data set has a large number of competing hypotheses which fit the data but extend in very different ways to the complete space, then considerably more data would be needed to disambiguate between these hypotheses. For certain probabilistic models (where function learning is essentially equivalent to statistical regression) Vapnik and Chervonenkis studied this problem closely and developed the notion of VC-dimension: a combinatorial measure of the complexity of the class $\mathcal{H}$ which is related to its sample complexity (see also Blumer et al (1986) for applications to computational learning theory).

Thus broadly speaking, the more constrained the hypothesis class $\mathcal{H}$, the smaller is the sample complexity (i.e. the easier it is to choose from finite experience the best hypothesis) but then again, the poorer is the expressive power and consequently even h_∞ might be far away from the reality c. On the other hand, increasing the expressive power of $\mathcal{H}$ might decrease $d(h_\infty, c)$ but increase the sample complexity. There is thus an inherent tension between the complexity of $\mathcal{H}$ and the number of examples; finding the class $\mathcal{H}$ of the right complexity is the challenge of science. Part of the understanding of biological phenomena involves deciding where on the tightrope between extremely complex and extremely simple models the true phenomena lie. In this respect, informational complexity is a powerful tool to help discriminate between models of different complexities to describe natural phenomena.

One sterling example where this information-complexity approach has startlingly revised the kinds of models used can be found in the Chomskyan revolution in linguistics. Humans develop a mature knowledge of language which is both rich and subtle on the basis of example sentences spoken to them by parents and guardians during childhood. On observing the child language acquisition process, it is remarkable how few examples they need to be able to generalize in very sophisticated ways. Further it is observed that children generalize in roughly the same way; too striking a coincidence to be attributed purely to chance. Languages are infinite sets of sentences; yet on the basis of exposure to finite linguistic experience (sentences) children generalize to the infinite set. If it were the case that children operated with completely unconstrained hypotheses about languages, i.e., if they were willing to consider all possible infinite extensions to the finite data set they had, then they would never be able to generalize correctly or generalize in the same manner. They received far too few examples for that. This "poverty of stimulus" in the child language acquisition process motivated Chomsky to suggest that children operate with hypotheses about language which are constrained in some fashion. In other words, we are genetically

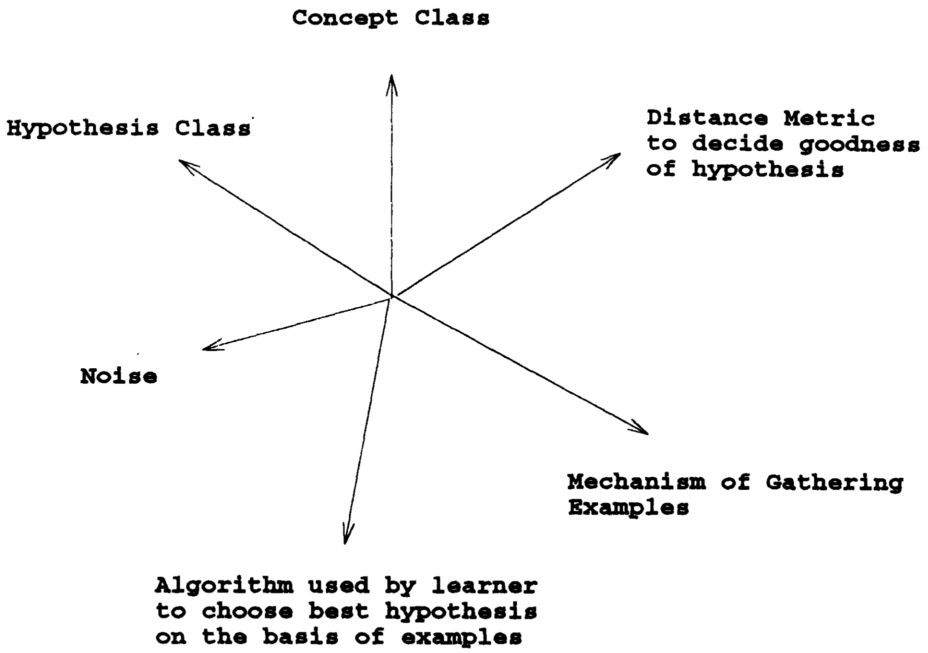
Concept Class
Hypothesis Class
Distance Metric to decide goodness of hypothesis
Noise
Mechanism of Gathering Examples
Algorithm used by learner to choose best hypothesis on the basis of examples

Figure 1-1: The space of possibilities. The various factors which affect the informational complexity of learning from examples.

predisposed as human beings to choose certain generalizations and not others; we operate with a set of restricted hypotheses. The goal of linguistics then shifted to finding the class $\mathcal{H}$ with the right complexity; something which had large enough expressive power to capture the natural languages, and low enough to be learned by children. In this thesis we spend some time on models for learning languages.

Thus we see that an investigation of the informational complexity of learning has implications for model building; something which is at the core of the scientific enterprise. Particularly when studying cognitive behavior, it might potentially allow us to choose the right complexity, i.e., how much processing is already built into the brain (the analog of Hubel and Wiesel's orientation-specific neurons or Chomsky's universal grammar) and how much is acquired by exposure to the environment. At this point, it would be worthwhile to point out that the complexity of $\mathcal{H}$ is only one of the factors influencing the informational complexity. Recall that we have already sharpened our notion of informational complexity to mean the number of examples needed by the learner so that $d(h_m, h_\infty)$ is small. There are several factors which could in principle affect it and Figure 1.1 shows them as decomposed along several different dimensions in the space of possibilities.

Clearly, informational complexity might depend upon upon the manner in which examples are obtained. If one were learning to discriminate between the sounds /s/ and /**sh**/, for example, one could potentially learn more effectively if one were presented with examples drawn from near the decision boundary, i.e., examples of sounds which were likely to be confused. Such a presentation might conceivably help the learner acquire a sharper idea of the distinction between the two sounds rather than if it were simply presented with canonical examples of each phoneme. Of course, it might well be the case that our intuition is false in this case, but we will never know unless the issue is formally addressed. In similar fashion, the presence and nature of the noise corrupting the examples could affect sample complexity. In the case of **s/sh** classification, a lot of noise in high frequency bands of the signal could affect our perception of frication and might delay learning; on the other hand noise which only affects volume of the signal might have less effect. The algorithm used to compute a best hypothesis h_m from the data might affect both learnability and sample complexity. A muddle-headed poorly motivated algorithm might choose hypotheses at random or it might choose hypotheses according to some criterion which has nothing to do with the metric d by which success is to be measured. In such cases, it is possible that h_m might not converge to h_∞ at all, or it might take a very long time. Finally the metric d according to which success is to be measured is clearly a factor.

These different factors interact with each other; our central goal in this thesis is to explore this possibility-space at many different points. We will return to this space and our points of exploration later. It is our hope that after seeing the interaction between the different dimensions and their relation to informational complexity, our intuitions about the analysis of learning paradigms will be sharpened.

1.2 Parametric Hypothesis Spaces

We have already introduced the notion of hypotheses and hypothesis classes from which these hypotheses are chosen. We have also remarked that the number of examples needed to choose a "best" hypothesis (or at any rate, one close enough to the best according to our distance metric) depends inherently upon the complexity of these classes. Another related question of some interest is: how do we represent these hypotheses? One approach pervasive in science is to capture the degree of variability amongst the hypotheses in a parametric fashion. The greater the flexibility of the parameterization, the greater the allowed variability and the less is the inbuilt constraints, i.e., the larger the domain and consequently the larger the search space. One can consider several other examples from the sciences where parametric models

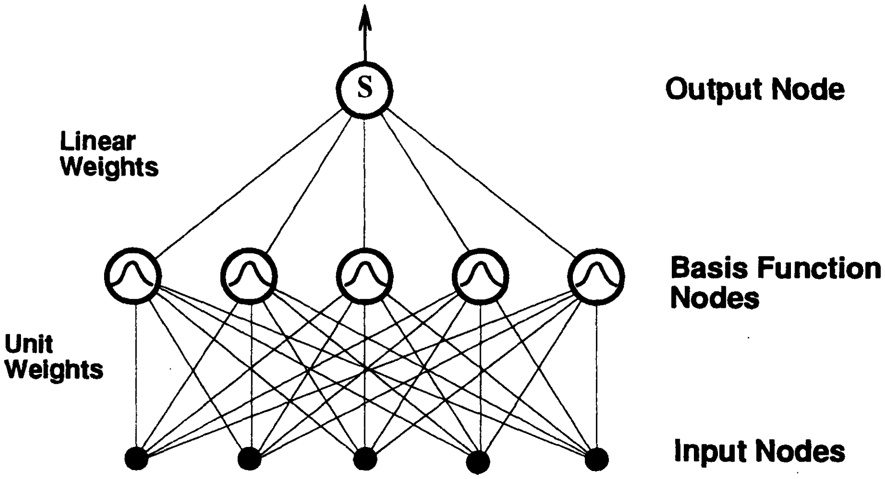

Figure 1-2: The structure of a Hyper Basis Function Network (same as regularization network).

have been developed for some task or other.

In our thesis, we spend a considerable amount of time and energy on two parametric models which are remarkably different in their structural properties and analyze issues of informational complexity in each. It is worthwhile perhaps to say a few words about each.

Neural Networks

Feed-forward "neural networks" (Lippman, 1987) are becoming increasingly popular in science and engineering as a modelling technique. We consider a class of feed-forward networks known as Gaussian regularization networks (Poggio and Girosi, 1990). Essentially, such a network performs a mapping from $\Re^k$ to $\Re$ given by the following expression

$$y = \sum_{i=1}^{n} c_i G(\frac{|\mathbf{x} - \mathbf{t_i}|}{\sigma_i})$$

Fig. 1-2 shows a diagrammatic (it is particularly popular in the neural net communities to show the diagrams or architecture and we see no need to break with tradition here) representation of the network. The c_i's are real-valued, G is a Gaussian function (activation function), the $\mathbf{t_i}$'s are the centers, and the σ_i's are the spreads of the Gaussian functions.

Clearly then, one can consider H_n to be the class of all functions which can be represented in the form above. This class would consist of functions parameterized by

$3n$ parameters; corresponding to the free variables c_i, t_i, and σ_i. One can make several alterations to the architecture; changing for example the number of layers, changing the activation functions, putting constraints on the weights and so on thereby arriving at different kinds of parameterized families, e.g., the multilayer perceptrons with sigmoidal units, hierarchical mixture of experts (Jacobs et al, 1991) etc. Such feed forward networks have been used for tasks as diverse as discriminating between virgin and non-virgin olive oil, speech recognition, predicting the stock market, robotic control and so forth. Given the prevalence of such neural networks, we have chosen in this thesis to investigate issues pertaining to informational complexity of networks.

Natural Languages

Natural languages can be described by their grammars which are essentially functional mappings from strings to the set $\{0, 1\}$. According to conventional notation, there is an alphabet set Σ which is a finite set of symbols. In the case of a particular natural language, like English, for example, this set is the vocabulary: a finite set of words. These symbols or words are the basic building blocks of sentences which are just strings of words. $\Sigma*$ denotes the set of all finite sentences and a language L is a subset of $\Sigma*$, i.e., some collection of sentences which belong to the language. For example, in English,*I eat bananas* is a sentence (an element of $\Sigma*$), being as it is a string of the three words (elements of Σ), *I*, *eat*, and *bananas*. Further, this sentence belongs to the set of valid English sentences. On the other other hand, the sentence *I bananas eat*, though a member of $\Sigma*$ is not a member of the set of valid English sentences.

The grammar $G_L$associated with the language L then is a functional description of the mapping from $\Sigma*$ to $\{0, 1\}$, all sentences belonging to $\Sigma*$ which belong to L are mapped onto 1 by G_L, the rest are assigned to 0. According to current theories of linguistics which we will consider in this thesis, it is profitable for analysis to let the set Σ consist of syntactic categories like verbs, adverbs, prepositions, nouns, and so on. A sentence could now be considered to be a string of such syntactic categories; each category then maps onto words of the vocabulary. Thus the string of syntactic categories **Noun Verb Noun** maps onto *I eat bananas*; the string **Noun Noun Verb** maps onto *I bananas eat.* A grammar is a systematic system of rules and principles which pick out some strings of syntactic categories as valid, others as not. Most of linguistic theory concentrates on generative grammars; grammars which are able to build the valid sentences out of the syntactic components according to certain rules. Phrase structure grammars build sentences out of phrases; and phrases out of other phrases or syntactic categories.

Over the last decade, a parametric theory of grammars (Chomsky, 1981) has begun to evolve. According to this, a grammar $G(p_1, \ldots, p_n)$ is parameterized by a finite (in this case, n) number of parameters p_1 through p_n. If these parameters are set to one set of values, one would obtain the grammar of a specific language, say, German. Setting them to another set of values would define the grammar of another language, say English. To get a feel for what parameters are like, consider an example from X-bar theory; a subcomponent of grammars. According to X-bar theory, the structure of an XP or X-phrase (where X could stand for adjective, noun, verb, etc.) is given by the following context-free production rules which are parameterized by two parameters p_1 and p_2.

$$XP \longrightarrow \text{Spec } X'(p_1 = 0) \text{ or } X' \text{ Spec } (p_1 = 1)$$

$$X' \longrightarrow \text{Comp } X'(p_2 = 0) \text{ or } X' \text{ Comp } (p_2 = 1)$$

$$X' \longrightarrow \text{Comp } X(p_2 = 0) \text{ or } X \text{ Comp } (p_2 = 1)$$

$$\text{Comp } \longrightarrow YP$$

For example, English is a comp-final language ($p_2 = 1$) while Bengali is a comp-first language($p_2 = 0$). Notice how all the phrases (irrespective of whether it is a noun phrase, verb phrase etc.) in English have their complement in the end, while Bengali is the exact reverse. This is one example of a parameterized difference between the two languages.

Also shown in figures 1-4, and 1-5, we have the tree diagrams corresponding to the sentence "with one hand" in English and Bengali. English is spec-first and comp-final (i.e., $p_1 = 0$ and $p_2 = 1$); Bengali on the other hand is spec-first and comp-first ($p_1 = 0$ and $p_2 = 0$).

1.3 The Thesis: Technical Contents and Major Contributions

So far we have discussed in very general terms, the various components of a learning paradigm and their relationship to each other. We have stated our intention of analyzing the informational complexity of learning from examples; we have thus defined for ourselves the possibility space of Figure 1.1 that needs to be explored. In this thesis, we look at a few specific points in this space; in doing so, the issues involved in informational complexity can be precisely formalized and sharper results obtained. Chapters 2 and 3 of this thesis are completely self contained. Chapters 4 and 5 should

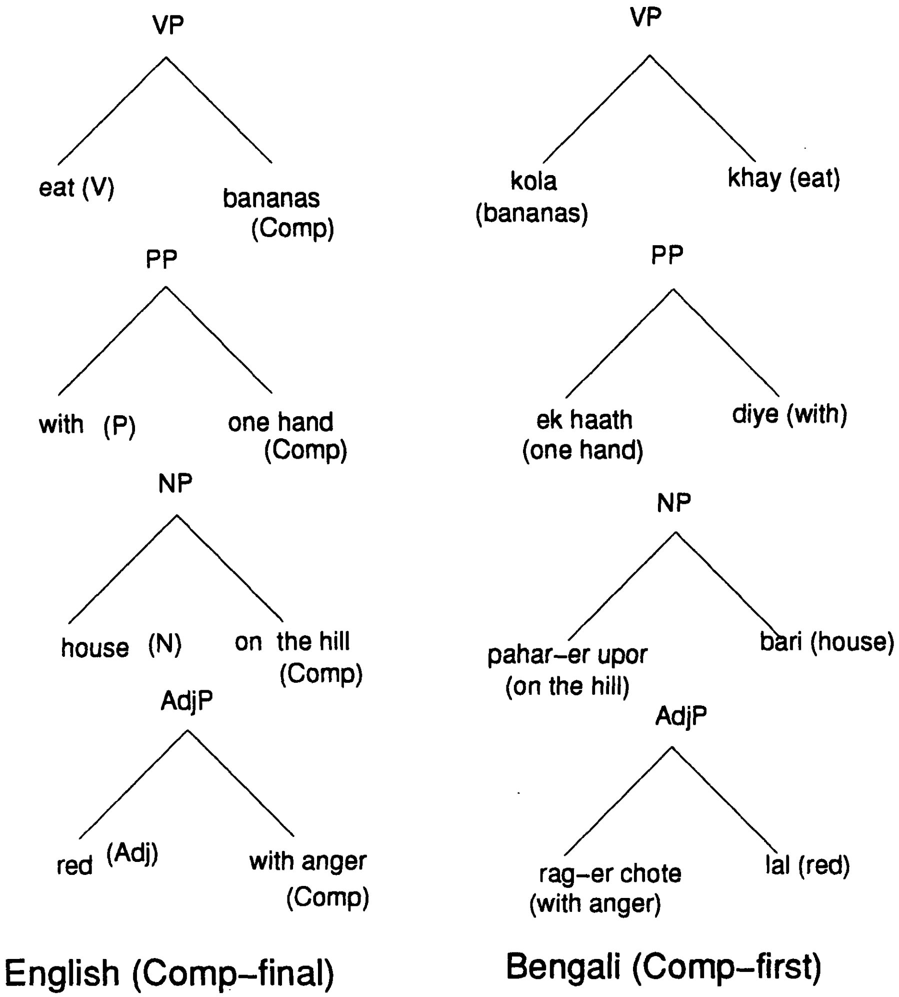

Figure 1-3: Parametric difference in phrase structure between English and Bengali on the basis of the parameter p_2.

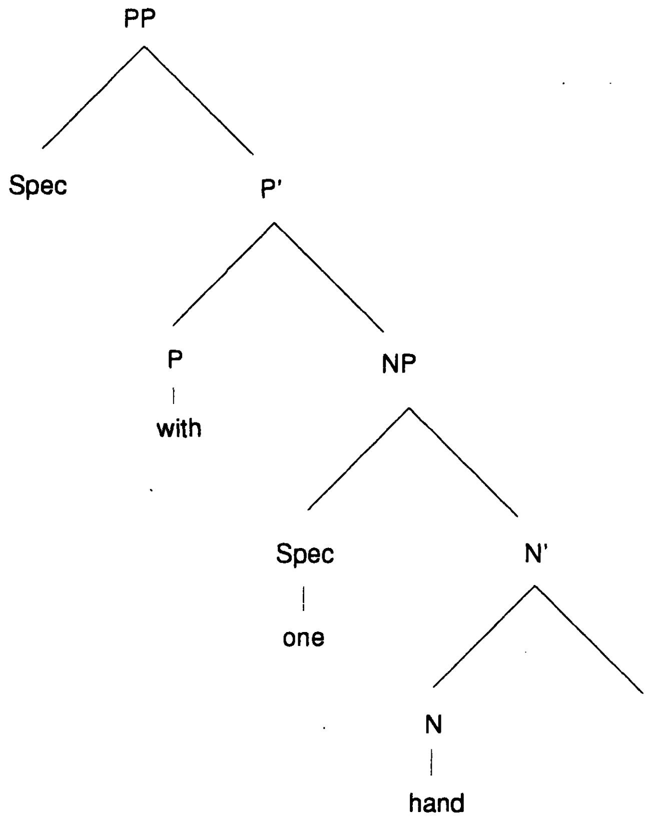

Figure 1-4: Analysis of the English sentence "with one hand" according to its parameterized X-bar grammar.

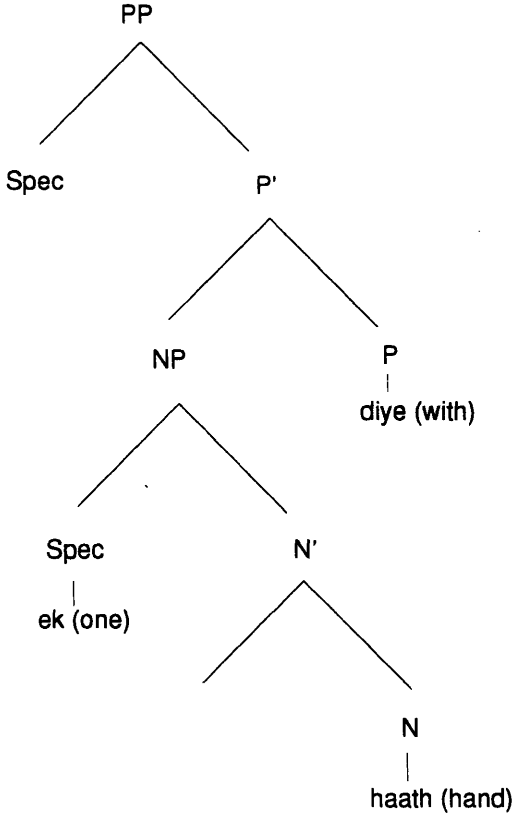

Figure 1-5: Analysis of the Bengali sentence "ek haath diye" a literal translation of "with one hand" according to its parameterized X-bar grammar. Notice the difference in word order.

be read as a unit; together they form another stand-alone part of this thesis.

Chapter 2 of this thesis examines the use of neural networks of a certain kind (the so called regularization networks) in solving pattern classification and regression problems. This corresponds to a point in the space of Figure 1.1 where the concept class is a Sobolev space of functions, the hypothesis class is the class of all feed forward regularization networks (with certain restrictions on their weights), the examples are drawn according to a fixed, unknown, arbitrary probability distribution, the distance metric is a $L_2(P)$ norm on the space of functions, the algorithm used to choose the best hypothesis is by training a finite sized network on labelled examples according to least-squares criterion. The concept class is infinite-dimensional; on using a finite network and finite amount of data, a certain amount of generalization error is made. We observe that the generalization error can be decomposed into an approximation error due to the finite number of parameters of the network and an estimation error due to the finite number of data points. Using techniques from approximation theory and VC theory, we obtain a bound on the generalization error in terms of the number of parameters and number of examples. Our main contributions in this chapter include:

- Formulation of the trade-off between hypothesis complexity and sample complexity when using Gaussian regularization networks.
- Combining results from approximation theory and the theory of empirical processes to obtain a specific bound on the total generalization error as a function of the number of examples and number of parameters.
- Using the bound above to provide guidelines for choosing an optimal network architecture to solve certain regression problems.

Chapter 3 explores the issue of active learning. We are specifically interested in investigating whether allowing the learner to choose examples helps in learning with fewer examples. This chapter consists of two parts which include several forays into this question. The first part explores this issue in a function approximation setting. It is not immediately clear that even if the learner were allowed to choose his/her own examples, there exist principled ways of doing this. We develop a framework within which meaningful adaptive sampling strategies can be obtained for arbitrary function classes. As specific examples we consider cases where the concept classes are real-valued classes like monotonic functions and functions with bounded first derivative, hypothesis classes are spline functions, there is no noise, the learner chooses an interpolating spline as a best hypothesis and examples are obtained passively (by random draw) or adaptively (according our strategy) by the active learner. We obtain theoretical and empirical bounds on the sample complexity and generalization error for this task. In the second part, we discuss the idea of epsilon-focusing; a strategy whereby the learner can adaptively focus on smaller and smaller regions of the domain to solve certain pattern classification problems. We derive conditions on function classes where epsilon-focusing would result in faster learning. Our main contributions here include:

- A formulation of active learning in approximation theoretic terms as an adaptive approximation problem.
- Development of active strategies for learning classes of real valued functions. These active strategies differ from traditional adaptive approximation strategies in optimal sampling theory in that examples are adaptively selected on the basis of previous examples as opposed to preselected on the basis of knowledge about the concept class.
- Explicit computation of theoretical upper and lower bounds on the sample complexity of PAC learning real classes using passive and active strategies. Simulations with some test target functions allows us to compare the empirical performance against the theoretical worst case bounds.
- Introduction of the idea of epsilon-focusing which provides a theoretical motivation for pattern classification schemes where more data is collected near the estimated class boundary. The computation of explicit sample complexity bounds for algorithms motivated by epsilon-focusing.

Chapters 4 and 5 of this thesis concentrate on a very different region of the possibility space of Figure 1.1. Here the concept class is a restricted subclass of natural languages, the hypothesis class consists of parameterized grammars including X-bar theory, verb movement and case theory, examples are assumed to be drawn according to some distribution on the sentences of the target, there might or might not be noise, there is a discrete distance metric which requires exact identification of the target, the algorithm used to choose the best hypothesis is the Triggering Learning Algorithm (Gibson and Wexler, 1993).

The TLA was proposed recently by Gibson and Wexler as a possible mechanism by which children set parameters and learned the language to which they were exposed. Chapter 4 originated as an attempt to analyze the TLA from the perspective of informational complexity and to derive conditions for convergence and rates of convergence of the TLA to the target. We explore the TLA and its variants under the diverse influence of noise, distributional assumptions on the data, and explore the linguistic consequences of this. In Chapter 5, we study another important facet of the language learning puzzle. Starting with a set of grammars and a learning algorithm, we are able to derive a dynamical system whose states correspond to the the linguistic composition of the population, i.e., the relative percentage of people in a community speaking a particular language. For the TLA, we give the precise update rules for the states of this system, analyze conditions for stability and carry out several simulations in linguistically plausible systems. This serves as a formal model for describing the historical evolution of languages and formalizes ideas inherent in Lightfoot (1991) and and Hawkins and Gell-Mann (1989) for the first time. These two chapters make several important contributions including:

- The development of a mathematical framework (a Markov structure) to formally study the issues relating to the learnability and sample complexity of the TLA.
- The investigation of variants of TLA, the effect of noise, distributional assumptions and parameterization of the space in a systematic manner on linguistically natural spaces.
- The derivation of algorithm-independent bounds on the sample complexity using results from computational learning theory.
- The derivation of a linguistic dynamical system starting from the TLA operating on parameterized grammars.
- Utilizing the dynamical system as a model for language change, running simulations on linguistically natural spaces and comparison of the results against historically observed patterns.
- Introduction of the diachronic criterion for deciding the plausibility of any learning algorithm.

1.3.1 A Final Word

Over the last decade, there has been a explosion of interest in formal learning theory (see the Proceedings of ACM COLT for a whiff of this). This has brought in its wake a perspective on learning paradigms which we greatly share and this thesis reflects that perspective strongly. In addition, as with all interdisciplinary pieces of work, we have an intellectual debt to many different fields. The areas of approximation theory and statistics, particularly the part of empirical process theory beautifully worked out by

Vapnik and Chervonenkis, model selection, pattern recognition, decision theory, and nonparametric regression play an important role in Chapter 2. Ideas from adaptive integration and numerical analysis play an important role in chapter 3. Chapters 4 and 5 have evolved from the application of our computational perspective to the analysis of learning paradigms which are considered worthwhile in linguistic theory (our decision of what is linguistically worthwhile has been influenced greatly by scholarly works in the Chomskyan tradition). Here, there is some use of Markov chain theory and dynamical systems theory. In all of this, we have brought to bear well known results and techniques from different areas of mathematics to formally pose and answer questions of interest in human and machine learning; questions previously unposed or unanswered or both. In this strict sense, there is little new mathematics here; though an abundant demonstration of its usefulness as a research tool in the cognitive and computer sciences. This reflects our purpose and our intended audience for this thesis, namely, all people interested in human or machine learning from a computational perspective.

Chapter 2

On the Relationship Between Generalization Error, Hypothesis Complexity, and Sample Complexity in Radial Basis Functions

Abstract

Feedforward networks are a class of approximation techniques that can be used to learn to perform some tasks from a finite set of examples. The question of the capability of a network to generalize from a finite training set to unseen data is clearly of crucial importance. In this chapter, we bound the generalization error of a class of Radial Basis Functions, for certain well defined function learning tasks, in terms of the number of parameters and number of examples. We show that the total generalization error is partly due to the insufficient representational capacity of the network (because of the finite size of the network being used) and partly due to insufficient information about the target function because of the finite number of samples. Prior research has looked at representational capacity or sample complexity in isolation. In the spirit of A. Barron, H. White and S. Geman we develop a framework to look at both. While the bound that we derive is specific for Radial Basis Functions, a number of observations deriving from it apply to any approximation technique. Our result also sheds light on ways to choose an appropriate network architecture for a particular problem and the kinds of problems which can be effectively solved with finite resources, i.e., with finite number of parameters and finite amounts of data.

2.1 Introduction

Many problems in learning theory can be effectively modelled as learning an input output mapping on the basis of limited evidence of what this mapping might be. The mapping usually takes the form of some unknown function between two spaces and the evidence is often a set of labelled, noisy, examples i.e., (x, y) pairs which are consistent with this function. On the basis of this data set, the learner tries to infer the true function.

We have discussed in Chapter 1, several examples from speech recognition, object recognition, and finance where such a scenario exists. At the risk of belaboring this point consider two more examples which illustrate this approach. In economics, it is sometimes of interest to predict the future foreign currency rates on the basis of the past time series. There might be a function which captures the dynamical relation between past and future currency rates and one typically tries to uncover this relation from data which has been appropriately processed. Similarly in medicine, one might be interested in predicting whether or not breast cancer will recur in a patient within five years after her treatment. The input space might involve dimensions like the age of the patient, whether she has been through menopause, the radiation treatment previously used etc. The output space would be single dimensional boolean taking on values depending upon whether breast cancer recurs or not. One might collect data from case histories of patients and try to uncover the underlying function.

The unknown target function is assumed to belong to some class $\mathcal{F}$ which using the terminology of computational learning theory we call the *concept class*. Typical examples of concept classes are classes of indicator functions, boolean functions, Sobolev spaces etc. The learner is provided with a finite data set. One can make many assumptions about how this data set is collected but a common assumption which would suffice for our purposes is that the data is drawn by sampling independently the input output space $(X \times Y)$ according to some unknown probability distribution. On the basis of this data, the learner then develops a hypothesis (another function) about the identity of the target function i.e., it comes up with a function chosen from some class, say H (the *hypothesis class*) which best fits the data and postulates this to be the target. Hypothesis classes could also be of different kinds. For example, they could be classes of boolean functions, polynomials, linear functions, spline functions and so on. One such class which is being increasingly used for learning problems is the class of feedforward networks ((Lippmann, 1987; Hertz, Krogh, and Palmer, 1991; Girosi, Jones, and Poggio, 1993). A typical feedforward network is a parameterized function of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i H(\mathbf{x}; \mathbf{w}_i)$$

where $\{c_i\}_{i=1}^{n}$ and $\{\mathbf{w}_i\}_{i=1}^{n}$ are free parameters and $H(\cdot;\cdot)$ is a given, fixed function (the "activation function"). Depending on the choice of the activation function one gets different network models, such as the most common form of "neural networks", the Multilayer Perceptron (Rumelhart, Hinton, and Williams, 1986; Cybenko, 1989; Lapedes, and Farmer, 1988; Hertz, Krogh, and Palmer, 1991; Hornik, Stinchcombe, and White, 1989; Funahashi, 1989; Mhaskar, and Micchelli, 1992; Mhaskar, 1993; Irie, and Miyake, 1988) , or the Radial Basis Functions network (Broomhead, and

Lowe, 1988; Dyn, 1987; Hardy, 1971,1990; Micchelli, 1986; Powell, 1990; Moody, and Darken, 1989; Poggio, and Girosi, 1990; Girosi, 1992; Girosi, Jones, and Poggio, 1993).

If, as more and more data becomes available, the learner's hypothesis becomes closer and closer to the target and converges to it in the limit, the target is said to be learnable. The error between the learner's hypothesis and the target function is defined to be the *generalization error* and for the target to be learnable the generalization error should go to zero as the data goes to infinity. While learnability is certainly a very desirable quality, it requires the fulfillment of two important criteria.

First, there is the issue of the representational capacity (or *hypothesis complexity*) of the hypothesis class. This must have sufficient power to represent or closely approximate the concept class. Otherwise for some target function f, the best hypothesis h in H might be far away from it. The error that this best hypothesis makes is formalized later as the *approximation error.* In this case, all the learner can hope to do is to converge to h in the limit of infinite data and so it will never recover the target. Second, we do not have infinite data but only some finite random sample set from which we construct a hypothesis. This hypothesis constructed from the finite data might be far from the best possible hypothesis, h, resulting in a further error. This additional error (caused by finiteness of data) is formalized later as the *estimation error.* The amount of data needed to ensure a small estimation error is referred to as the *sample complexity* of the problem. The hypothesis complexity, the sample complexity and the generalization error are related. If the class H is very large or in other words has high complexity, then for the same estimation error, the sample complexity increases. If the hypothesis complexity is small, the sample complexity is also small but now for the same estimation error the approximation error is high. This point has been developed in terms of the Bias-Variance trade-off in (Geman, Bienenstock, and Doursat, 1992) in the context of neural networks, and others (Rissanen, 1983; Grenander, 1951; Vapnik, 1982; Stone, 1974) in statistics in general.

The purpose of this chapter is two-fold. First, we formalize the problem of learning from examples so as to highlight the relationship between hypothesis complexity, sample complexity and total error. Second, we explore this relationship in the specific context of a particular hypothesis class. This is the class of Radial Basis function networks which can be considered to belong to the broader class of feed-forward networks. Specifically, we are interested in asking the following questions about radial basis functions.

Imagine you were interested in solving a particular problem (regression or pattern classification) using Radial Basis Function networks. Then, how large must the net-

work be and how many examples do you need to draw so that you are guaranteed with high confidence to do very well? Conversely, if you had a finite network and a finite amount of data, what are the kinds of problems you could solve effectively?

Clearly, if one were using a network with a finite number of parameters, then its representational capacity would be limited and therefore even in the best case we would make an approximation error. Drawing upon results in approximation theory (Lorentz, 1986) several researchers (Cybenko, 1989; Hartman, Keeler, and Kowalski, 1989; Barron, 1991; Hornik, Stinchcombe, and White, 1989; Chui, and Li, 1990; Arai, 1989; Mhaskar, and Micchelli, 1992; Mhaskar, 1993; Irie, and Miyake, 1988; Chen, Chen, and Liu, 1990) have investigated the approximating power of feedforward networks showing how as the number of parameters goes to infinity, the network can approximate any continuous function. These results assume infinite data and questions of learnability from finite data are ignored. For a finite network, due to finiteness of the data, we make an error in estimating the parameters and consequently have an estimation error in addition to the approximation error mentioned earlier. Using results from Vapnik and Chervonenkis (Vapnik, 1982; Vapnik, and Chervonenkis, 1971, 1981, 1991) and Pollard (Pollard, 1984) , work has also been done (Haussler, 1989; Baum, and Haussler, 1988) on the sample complexity of finite networks showing how as the data goes to infinity, the estimation error goes to zero i.e., the empirically optimized parameter settings converge to the optimal ones for that class. However, since the number of parameters are fixed and finite, even the optimal parameter setting might yield a function which is far from the target. This issue is left unexplored by Haussler (1989) in an excellent investigation of the sample complexity question.

In this chapter, we explore the errors due to both finite parameters and finite data in a common setting. In order for the total generalization error to go to zero, both the number of parameters and the number of data have to go to infinity, and we provide rates at which they grow for learnability to result. Further, as a corollary, we are able to provide a principled way of choosing the optimal number of parameters so as to minimize expected errors. It should be mentioned here that White (1990) and Barron (1991) have provided excellent treatments of this problem for different hypothesis classes. We will mention their work at appropriate points in this chapter.

The plan of the chapter is as follows: in section 2.2 we will formalize the problem and comment on issues of a general nature. We then provide in section 2.3 a precise statement of a specific problem. In section 2.4 we present our main result, whose proof is postponed to appendix 2-D for continuity of reading. The main result is qualified by several remarks in section 2.5. In section 2.6 we will discuss what could be the implications of our result in practice and finally we conclude in section 2.7

with a reiteration of our essential points.

2.2 Definitions and Statement of the Problem

In order to make a precise statement of the problem we first need to introduce some terminology and to define a number of mathematical objects. A summary of the most common notations and definitions used in this chapter can be found in appendix 2-A.

2.2.1 Random Variables and Probability Distributions

Let X and Y be two arbitrary sets. We will call $\mathbf{x}$ and y the *independent variable* and *response* respectively, where $\mathbf{x}$ and y range over the generic elements of X and Y. In most cases X will be a subset of a k-dimensional Euclidean space and Y a subset of the real line, so that the independent variable will be a k-dimensional vector and the response a real number. We assume that a probability distribution $P(\mathbf{x}, y)$ is defined on $X \times Y$. P is unknown, although certain assumptions on it will be made later in this section.

The probability distribution $P(\mathbf{x}, y)$ can also be written as[4]:

$$P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x}) \ , \tag{2.1}$$

where $P(y|\mathbf{x})$ is the conditional probability of the response y given the independent variable $\mathbf{x}$, and $P(\mathbf{x})$ is the marginal probability of the independent variable given by:

$$P(\mathbf{x}) = \int_Y dy \ P(\mathbf{x}, y) \ .$$

Expected values with respect to $P(\mathbf{x}, y)$ or $P(\mathbf{x})$ will be always indicated by $E[\cdot]$. Therefore, we will write:

$$E[g(\mathbf{x}, y)] \equiv \int_{X \times Y} d\mathbf{x}dy \ P(\mathbf{x}, y)g(\mathbf{x}, y)$$

and

$$E[h(\mathbf{x})] \equiv \int_X d\mathbf{x} \ P(\mathbf{x})h(\mathbf{x})$$

for any arbitrary function g or h.

[4] Note that we are assuming that the conditional distribution exists, but this is not a very restrictive assumption.

2.2.2 Learning from Examples and Estimators

The framework described above can be used to model the fact that in the real world we often have to deal with sets of variables that are related by a probabilistic relationship. For example, y could be the measured torque at a particular joint of a robot arm, and $\mathbf{x}$ the set of angular position, velocity and acceleration of the joints of the arm in a particular configuration. The relationship between $\mathbf{x}$ and y is probabilistic because there is noise affecting the measurement process, so that two different torques could be measured given the same configuration.

In many cases we are provided with *examples* of this probabilistic relationship, that is with a data set D_l, obtained by sampling l times the set $X \times Y$ according to $P(\mathbf{x}, y)$:

$$D_l \equiv \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{l} \ .$$

From eq. (2.1) we see that we can think of an element $(\mathbf{x}_i, y_i)$ of the data set D_l as obtained by sampling X according to $P(\mathbf{x})$, and then sampling Y according to $P(y|\mathbf{x})$. In the robot arm example described above, it would mean that one could move the robot arm into a random configuration $\mathbf{x}_1$, measure the corresponding torque y_1, and iterate this process l times.

The interesting problem is, given an instance of $\mathbf{x}$ that does not appear in the data set D_l, to give an estimate of what we expect y to be. For example, given a certain configuration of the robot arm, we would like to estimate the corresponding torque.

Formally, we define an *estimator* to be any function $f : X \to Y$. Clearly, since the independent variable $\mathbf{x}$ need not determine uniquely the response y, any estimator will make a certain amount of error. However, it is interesting to study the problem of finding the best possible estimator, given the knowledge of the data set D_l, and this problem will be defined as the problem of *learning from examples*, where the examples are represented by the data set D_l. Thus we have a probabilistic relation between $\mathbf{x}$ and y. One can think of this as an underlying deterministic relation corrupted with noise. Hopefully a good estimator will be able to recover this relation.

2.2.3 The Expected Risk and the Regression Function

In the previous section we explained the problem of learning from examples and stated that this is the same as the problem of finding the best estimator. To make sense of this statement, we now need to define a measure of how good an estimator is. Suppose we sample $X \times Y$ according to $P(\mathbf{x}, y)$, obtaining the pair $(\mathbf{x}, y)$. A measure[5] of the error of the estimator f at the point $\mathbf{x}$ is:

$$(y - f(\mathbf{x}))^2 .$$

In the example of the robot arm, $f(\mathbf{x})$ is our estimate of the torque corresponding to the configuration $\mathbf{x}$, and y is the measured torque of that configuration. The average error of the estimator f is now given by the functional

$$I[f] \equiv E[(y - f(\mathbf{x}))^2] = \int_{X \times Y} d\mathbf{x}dy \; P(\mathbf{x}, y)(y - f(\mathbf{x}))^2 ,$$

that is usually called the *expected risk* of f for the specific choice of the error measure.

Given this particular measure as our yardstick to evaluate different estimators, we are now interested in finding the estimator that minimizes the expected risk. In order to proceed we need to specify its domain of definition $\mathcal{F}$. Then using the expected risk as a criterion, we could obtain the best element of $\mathcal{F}$. Depending on the properties of the unknown probability distribution $P(\mathbf{x}, y)$ one could make different choices for $\mathcal{F}$. We will assume in the following that $\mathcal{F}$ is some space of differentiable functions. For example, $\mathcal{F}$ could be a space of functions with a certain number of bounded derivatives (the spaces $\Lambda^m(R^d)$ defined in appendix 2-A), or a Sobolev space of functions with a certain number of derivatives in L_p (the spaces $H^{m,p}(R^d)$ defined in appendix 2-A).

Assuming that the problem of minimizing $I[f]$ in $\mathcal{F}$ is well posed, it is easy to obtain its solution. In fact, the expected risk can be decomposed in the following way (see appendix 2-B):

$$I[f] = E[(f_0(\mathbf{x}) - f(\mathbf{x}))^2] + E[(y - f_0(\mathbf{x}))^2] \tag{2.2}$$

where $f_0(\mathbf{x})$ is the so called *regression function*, that is the conditional mean of the response given the independent variable:

$$f_0(\mathbf{x}) \equiv \int_Y dy \; yP(y|\mathbf{x}) . \tag{2.3}$$

From eq. (2.2) it is clear that the regression function is the function that minimizes the expected risk in $\mathcal{F}$, and is therefore the best possible estimator. Hence,

[5] Note that this is the familiar squared-error and when averaged over its domain yields the mean squared error for a particular estimator, a very common choice. However, it is useful to remember that there could be other choices as well.

$$f_0(\mathbf{x}) = \arg \min_{f \in \mathcal{F}} I[f] \ .$$

However, it is also clear that even the regression function will make an error equal to $E[(y - f_0(\mathbf{x}))^2]$, that is the variance of the response given a certain value for the independent variable, averaged over the values the independent variable can take. While the first term in eq. (2.2) depends on the choice of the estimator f, the second term is an intrinsic limitation that comes from the fact that the independent variable $\mathbf{x}$ does not determine uniquely the response y.

The problem of learning from examples can now be reformulated as the problem of reconstructing the regression function f_0, given the example set D_l. Thus we have some large class of functions $\mathcal{F}$ to which the target function f_0 belongs. We obtain noisy data of the form $(\mathbf{x}, y)$ where $\mathbf{x}$ has the distribution $P(\mathbf{x})$ and for each $\mathbf{x}$, y is a random variable with mean $f_0(\mathbf{x})$ and distribution $P(y|\mathbf{x})$. We note that y can be viewed as a deterministic function of $\mathbf{x}$ corrupted by noise. If one assumes the noise is additive, we can write $y = f_0(\mathbf{x}) + \eta_x$ where η_x[6] is zero-mean with distribution $P(y|\mathbf{x})$. We choose an estimator on the basis of the data set and we hope that it is close to the regression (target) function. It should also be pointed out that this framework includes pattern classification and in this case the regression (target) function corresponds to the Bayes discriminant function (Gish, 1990; Hampshire, and Pearlmutter, 1990; Richard, and Lippman, 1991) .

2.2.4 The Empirical Risk

If the expected risk functional $I[f]$ were known, one could compute the regression function by simply finding its minimum in $\mathcal{F}$, that would make the whole learning problem considerably easier. What makes the problem difficult and interesting is that in practice $I[f]$ is unknown because $P(\mathbf{x}, y)$ is unknown. Our only source of information is the data set D_l which consists of l independent random samples of $X \times Y$ drawn according to $P(\mathbf{x}, y)$. Using this data set, the expected risk can be approximated by the *empirical risk* I_{emp}:

$$I_{\text{emp}}[f] \equiv \frac{1}{l} \sum_{i=1}^{l} (y_i - f(\mathbf{x}_i))^2 \ .$$

For each given estimator f, the empirical risk is a random variable, and under fairly

[6] Note that the standard regression problem often assumes η_x is independent of x. Our case is distribution free because we make no assumptions about the nature of η_x.

general assumptions[7], by the law of large numbers (Dudley, 1989) it converges in probability to the expected risk as the number of data points goes to infinity:

$$\lim_{l \to \infty} P\{|I[f] - I_{\text{emp}}[f]| > \varepsilon\} = 0 \;\; \forall \varepsilon > 0 \; . \tag{2.4}$$

Therefore a common strategy consists in estimating the regression function as the function that minimizes the empirical risk, since it is "close" to the expected risk if the number of data is high enough. For the error metric we have used, this yields the least-squares error estimator. However, eq. (2.4) states only that the expected risk is "close" to the empirical risk *for each given* f, and not for all f *simultaneously.* Consequently the fact that the empirical risk converges in probability to the expected risk when the number, l, of data points goes to infinity does not guarantee that the minimum of the empirical risk will converge to the minimum of the expected risk (the regression function). As pointed out and analyzed in the fundamental work of Vapnik and Chervonenkis the notion of *uniform convergence* in probability has to be introduced, and it will be discussed in other parts of this chapter.

2.2.5 The Problem

The argument of the previous section suggests that an approximate solution of the learning problem consists in finding the minimum of the empirical risk, that is solving

$$\min_{f \in \mathcal{F}} I_{\text{emp}}[f] \; .$$

However this problem is clearly ill-posed, because, for most choices of $\mathcal{F}$, it will have an infinite number of solutions. In fact, all the functions in $\mathcal{F}$ that interpolate the data points $(\mathbf{x}_i, y_i)$, that is with the property

$$f(\mathbf{x}_i) = y_i \;\; 1, \ldots, l$$

will give a zero value for I_{emp}. This problem is very common in approximation theory and statistics and can be approached in several ways. A common technique consists in restricting the search for the minimum to a smaller set than $\mathcal{F}$. We consider the case in which this smaller set is a family of *parametric functions*, that is a family of functions defined by a certain number of real parameters. The choice of a parametric representation also provides a convenient way to store and manipulate the hypothesis function on a computer.

We will denote a generic subset of $\mathcal{F}$ whose elements are parametrized by a number

[7] For example, assuming the data is independently drawn and $I[f]$ is finite.

of parameters proportional to n, by H_n. Moreover, we will assume that the sets H_n form a nested family, that is

$$H_1 \subset H_2 \subset \ldots \subset H_n \subset \ldots \subset H.$$

For example, H_n could be the set of polynomials in one variable of degree $n-1$, Radial Basis Functions with n centers, multilayer perceptrons with n sigmoidal hidden units, multilayer perceptrons with n threshold units and so on. Therefore, we choose as approximation to the regression function the function $\hat{f}_{n,l}$ defined as:[8]

$$\hat{f}_{n,l} \equiv \arg \min_{f \in H_n} I_{\text{emp}}[f] \ . \tag{2.5}$$

Thus, for example, if H_n is the class of functions which can be represented as $f = \sum_{\alpha=1}^{n} c_\alpha H(\mathbf{x}; \mathbf{w}_\alpha)$ then eq. (2.5) can be written as

$$\hat{f}_{n,l} \equiv \arg \min_{c_\alpha, \mathbf{w}_\alpha} I_{\text{emp}}[f] \ .$$

A number of observations need to be made here. First, if the class $\mathcal{F}$ is small (typically in the sense of bounded VC-dimension or bounded metric entropy (Pollard, 1984)), then the problem is not necessarily ill-posed and we do not have to go through the process of using the sets H_n. However, as has been mentioned already, for most interesting choices of $\mathcal{F}$ (e.g. classes of functions in Sobolev spaces, continuous functions etc.) the problem might be ill posed. However, this might not be the only reason for using the classes H_n. It might be the case that that is all we have or for some reason it is something we would like to use. For example, one might want to use a particular class of feed-forward networks because of ease of implementation in VLSI. Also, if we were to solve the function learning problem on a computer as is typically done in practice, then the functions in $\mathcal{F}$ have to be represented somehow. We might consequently use H_n as a representation scheme. It should be pointed out that the sets H_n and $\mathcal{F}$ have to be matched with each other. For example, we would hardly use polynomials as an approximation scheme when the class $\mathcal{F}$ consists of indicator functions or for that matter use threshold units when the class $\mathcal{F}$ contains continuous

[8] Notice that we are implicitly assuming that the problem of minizing $I_{\text{emp}}[f]$ over H_n has a solution, which might not be the case. However the quantity

$$E_{n,l} \equiv \inf_{f \in H_n} I_{\text{emp}}[f]$$

is always well defined, and we can always find a function $\hat{f}_{n,l}$ for which $I_{\text{emp}}[\hat{f}_{n,l}]$ is arbitrarily close to $E_{n,l}$. It will turn out that this is sufficient for our purposes, and therefore we will continue, assuming that $\hat{f}_{n,l}$ is well defined by eq. (2.5) functions. In particular, if we are to recover the regression function, H must be dense in $\mathcal{F}$. One could look at this matching from both directions. For a class $\mathcal{F}$, one might be interested in an appropriate choice of H_n. Conversely, for a particular choice of H_n, one might ask what classes $\mathcal{F}$ can be effectively solved with this scheme. Thus, if we were to use multilayer perceptrons, this line of questioning would lead us to identify the class of problems which can be effectively solved by them.

Thus, we see that in principle we would like to minimize $I[f]$ over the large class $\mathcal{F}$ obtaining thereby the regression function f_0. What we do in practice is to minimize the empirical risk $I_{\mathrm{emp}}[f]$ over the smaller class H_n obtaining the function $\hat{f}_{n,l}$. Assuming we have solved all the computational problems related to the actual computation of the estimator $\hat{f}_{n,l}$, the main problem is now:

how good is $\hat{\mathbf{f}}_{\mathbf{n,l}}$?

Independently of the measure of performance that we choose when answering this question, we expect $\hat{f}_{n,l}$ to become a better and better estimator as n and l go to infinity. In fact, when l increases, our estimate of the expected risk improves and our estimator improves. The case of n is trickier. As n increases, we have more parameters to model the regression function, and our estimator should improve. However, at the same time, because we have more parameters to estimate with the same amount of data, our estimate of the expected risk deteriorates. Thus we now need more data and n and l have to grow as a function of each other for convergence to occur. At what rate and under what conditions the estimator $\hat{f}_{n,l}$ improves depends on the properties of the regression function, that is on $\mathcal{F}$, and on the approximation scheme we are using, that is on H_n.

2.2.6 Bounding the Generalization Error

At this stage it might be worthwhile to review and remark on some general features of the problem of learning from examples. Let us remember that our goal is to minimize the expected risk $I[f]$ over the set $\mathcal{F}$. If we were to use a finite number of parameters, then we have already seen that the best we could possibly do is to minimize our functional over the set H_n, yielding the estimator f_n:

$$f_n \equiv \arg \min_{f \in H_n} I[f] \ .$$

However, not only is the parametrization limited, but the data is also finite, and we can only minimize the empirical risk I_{emp}, obtaining as our final estimate the function $\hat{f}_{n,l}$. Our goal is to bound the distance from $\hat{f}_{n,l}$ that is our solution, from f_0, that is the "optimal" solution. If we choose to measure the distance in the $L^2(P)$ metric (see appendix 2-A), the quantity that we need to bound, that we will call *generalization error*, is:

$$E[(f_0 - \hat{f}_{n,l})^2] = \int_X d\mathbf{x}\ P(\mathbf{x})(f_0(\mathbf{x}) - \hat{f}_{n,l}(\mathbf{x}))^2 =$$

$$= \|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)}$$

There are 2 main factors that contribute to the generalization error, and we are going to analyze them separately for the moment.

1. A first cause of error comes from the fact that we are trying to approximate an infinite dimensional object, the regression function $f_0 \in \mathcal{F}$, with a finite number of parameters. We call this error *the approximation error*, and we measure it by the quantity $E[(f_0 - f_n)^2]$, that is the $L_2(P)$ distance between the best function in H_n and the regression function. The approximation error can be expressed in terms of the expected risk using the decomposition (2.2) as

   $$E[(f_0 - f_n)^2] = I[f_n] - I[f_0] \ . \tag{2.6}$$

   Notice that the approximation error does not depend on the data set D_l, but depends only on the approximating power of the class H_n. The natural framework to study it is approximation theory, that abound with bounds on the approximation error for a variety of choices of H_n and $\mathcal{F}$. In the following we will always assume that it is possible to bound the approximation error as follows:

   $$E[(f_0 - f_n)^2] \leq \varepsilon(n)$$

   where $\varepsilon(n)$ is a function that goes to zero as n goes to infinity if H is dense in $\mathcal{F}$. In other words, as shown in figure (2-6), as the number n of parameters gets larger the representation capacity of H_n increases, and allows a better and better approximation of the regression function f_0. This issue has been studied by a number of researchers (Cybenko, 1989; Hornik, Stinchcombe, and White, 1989; Barron, 1991, 1993; Funahashi, 1989; Mhaskar, and Micchelli, 1992; Mhaskar, 1993) in the neural networks community.

2. Another source of error comes from the fact that, due to finite data, we minimize the empirical risk $I_{\mathrm{emp}}[f]$, and obtain $\hat{f}_{n,l}$, rather than minimizing the expected risk $I[f]$, and obtaining f_n. As the number of data goes to infinity we hope that

$\hat{f}_{n,l}$ will converge to f_n, and convergence will take place if the empirical risk converges to the expected risk *uniformly in probability* (Vapnik, 1982) . The quantity

$$|I_{\text{emp}}[f] - I[f]|$$

is called *estimation error*, and conditions for the estimation error to converge to zero uniformly in probability have been investigated by Vapnik and Chervonenkis Pollard , Dudley (1987) , and Haussler (1989) . Under a variety of different hypothesis it is possible to prove that, with probability $1-\delta$, a bound of this form is valid:

$$|I_{\text{emp}}[f] - I[f]| \leq \omega(l, n, \delta) \quad \forall f \in H_n \tag{2.7}$$

The specific form of ω depends on the setting of the problem, but, in general, we expect $\omega(l, n, \delta)$ to be a decreasing function of l. However, we also expect it to be an increasing function of n. The reason is that, if the number of parameters is large then the expected risk is a very complex object, and then more data will be needed to estimate it. Therefore, keeping fixed the number of data and increasing the number of parameters will result, on the average, in a larger distance between the expected risk and the empirical risk.

The approximation and estimation error are clearly two components of the generalization error, and it is interesting to notice, as shown in the next statement, the generalization error can be bounded by the sum of the two:

Statement 2.2.1 *The following inequality holds:*

$$\|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)} \leq \varepsilon(n) + 2\omega(l, n, \delta) \ . \tag{2.8}$$

Proof: using the decomposition of the expected risk (2.2), the generalization error can be written as:

$$\|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)} = E[(f_0 - \hat{f}_{n,l})^2] = I[\hat{f}_{n,l}] - I[f_0] \ . \tag{2.9}$$

A natural way of bounding the generalization error is as follows:

$$E[(f_0 - \hat{f}_{n,l})^2] \leq |I[f_n] - I[f_0]| + |I[f_n] - I[\hat{f}_{n,l}]| \ . \tag{2.10}$$

In the first term of the right hand side of the previous inequality we recognize the approximation error (2.6). If a bound of the form (2.7) is known for the generalization error, it is simple to show (see appendix (2-C) that the second term can be bounded as

$$|I[f_n] - I[\hat{f}_{n,l}]| \leq 2\omega(l, n, \delta)$$

and statement (2.2.1) follows □.

Thus we see that the generalization error has two components: one, bounded by $\varepsilon(n)$, is related to the approximation power of the class of functions $\{H_n\}$, and is studied in the framework of approximation theory. The second, bounded by $\omega(l, n, \delta)$, is related to the difficulty of estimating the parameters given finite data, and is studied in the framework of statistics. Consequently, results from both these fields are needed in order to provide an understanding of the problem of learning from examples. Figure (2-6) also shows a picture of the problem.

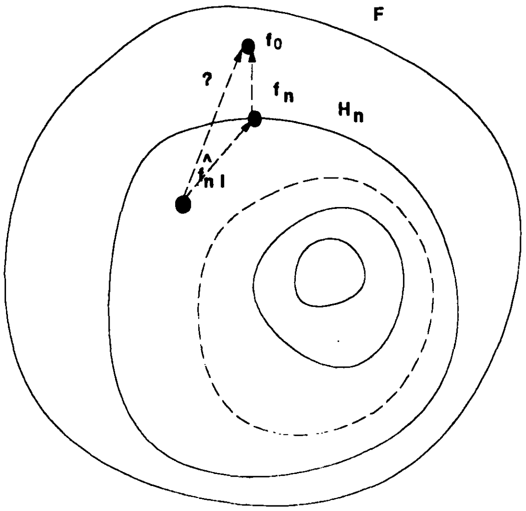

Figure 2-6: This figure shows a picture of the problem. The outermost circle represents the set F. Embedded in this are the nested subsets, the H_n's. f_0 is an arbitrary target function in $\mathcal{F}$, f_n is the closest element of H_n and $\hat{f}_{n,l}$ is the element of H_n which the learner hypothesizes on the basis of data.

2.2.7 A Note on Models and Model Complexity

From the form of eq. (2.8) the reader will quickly realize that there is a trade-off between n and l for a certain generalization error. For a fixed l, as n increases, the approximation error $\varepsilon(n)$ decreases but the estimation error $\omega(l, n, \delta)$ increases. Consequently, there is a certain n which might optimally balance this trade-off. Note that the classes H_n can be looked upon as models of increasing complexity and the search for an optimal n amounts to a search for the right model complexity. One typically wishes to match the model complexity with the sample complexity (measured by how much data we have on hand) and this problem is well studied (Eubank, 1988; Stone, 1974; Linehart, and Zucchini, 1986, Rissanen, 1989; Barron, and Cover, 1989; Efron, 1982; Craven, and Wahba, 1979) in statistics.

Broadly speaking, simple models would have high approximation errors but small estimation errors while complex models would have low approximation errors but high estimation errors. This might be true even when considering qualitatively different models and as an illustrative example let us consider two kinds of models we might use to learn regression functions in the space of bounded continuous functions. The class of linear models, i.e., the class of functions which can be expressed as $f = \mathbf{w} \cdot \mathbf{x} + \theta$, do not have much approximating power and consequently their approximation error is rather high. However, their estimation error is quite low. The class of models which can be expressed in the form $H = \sum_{i=1}^{n} c_i \sin(\mathbf{w}_i \cdot \mathbf{x} + \theta_i)$ have higher approximating power (Jones, 1990) resulting in low approximation errors. However this class has an infinite VC-dimension and its estimation error can not therefore be bounded.

So far we have provided a very general characterization of this problem, without stating what the sets $\mathcal{F}$ and H_n are. As we have already mentioned before, the set $\mathcal{F}$ could be a set of bounded differentiable or integrable functions, and H_n could be polynomials of degree n, spline functions with n knots, multilayer perceptrons with n hidden units or any other parametric approximation scheme with n parameters. In the next section we will consider a specific choice for these sets, and we will provide a bound on the generalization error of the form of eq. (2.8).

2.3 Stating the Problem for Radial Basis Functions

As mentioned before the problem of learning from examples reduces to estimating some target function from a set X to a set Y. In most practical cases, such as character recognition, motor control, time series prediction, the set X is the k-dimensional

Euclidean space R^k, and the set Y is some subset of the real line, that for our purposes we will assume to be the interval $[-M, M]$, where M is some positive number. In fact, there is a probability distribution $P(\mathbf{x}, y)$ defined on the space $R^k \times [-M, M]$ according to which the labelled examples are drawn independently at random, and from which we try to estimate the regression (target) function. It is clear that the regression function is a real function of k variables.

In this chapter we focus our attention on the Radial Basis Functions approximation scheme (also called Hyper-Basis Functions; Poggio and Girosi, 1990). This is the class of approximating functions that can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i G(\frac{\|\mathbf{x} - \mathbf{t}_i\|}{\sigma_i}) \tag{2.11}$$

where G is some given basis function (in our case Gaussian) and the β_i, $\mathbf{t}_i$, and σ_i are free parameters. We would like to understand what classes of problems can be solved "well" by this technique, where "well" means that both approximation and estimation bounds need to be favorable. It is possible to show that a favorable approximation bound can be obtained if we assume that the class of functions $\mathcal{F}$ to which the regression function belongs is defined as follows:

$$\mathcal{F} \equiv \{f | f = \lambda * G_m, m > k/2, |\lambda|_{R^k} \leq M\} \ . \tag{2.12}$$

Here λ is a signed Radon measure on the Borel sets of R^k, G_m is the Bessel-Macdonald kernel, i.e., the inverse fourier transform of

$$\tilde{G}_m(\mathbf{s}) = \frac{1}{(1 + 4\pi^2 \|\mathbf{s}\|^2)^{m/2}}$$

The symbol $*$ stands for the convolution operation, $|\lambda|_{R^k}$ is the total variation[9] of the measure λ and M is a positive real number. The space $\mathcal{F}$ as defined in eq. 2.12 is the so-called *Liouville Space* of order m. If m is even, this contains the *Sobolev Space* $H^{m,1}$ of functions whose derivatives upto order m are integrable.

We point out that the class $\mathcal{F}$ is non-trivial to learn in the sense that it has infinite pseudo-dimension (Pollard, 1984).

In order to obtain an estimation bound we need the approximating class to have bounded variation, and the following constraint will be imposed:

[9] A signed measure λ can be decomposed by the Hahn-Jordan decomposition into $\lambda = \lambda^+ - \lambda^-$. Then $|\lambda| = \lambda^+ + \lambda^-$ is called the total variation of λ. See Dudley (1989) for more information.

$$\sum_{i=1}^{n} |\beta_i| \leq M \ .$$

This constraint does not affect the approximation bound, and the two pieces fit together nicely. Thus the set H_n is defined now as the set of functions belonging to L_2 such that

$$f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i G(\frac{\|\mathbf{x} - \mathbf{t}_i\|}{\sigma_i}), \ \sum_{i=1}^{n} |\beta_i| \leq M \ , \quad \mathbf{t}_i \in R^k \ , \ \sigma_i \in R \qquad (2.13)$$

Having defined the sets H_n and $\mathcal{F}$ we remind the reader that our goal is to recover the regression function, that is the minimum of the expected risk over $\mathcal{F}$. What we end up doing is to draw a set of l examples and to minimize the empirical risk I_{emp} over the set H_n, that is to solve the following non-convex minimization problem:

$$\hat{f}_{n,l} \equiv \arg \min_{\beta_\alpha, \mathbf{t}_\alpha, \sigma_\alpha} \sum_{i=1}^{l} (y_i - \sum_{\alpha=1}^{n} \beta_\alpha G(\frac{\|\mathbf{x}_i - \mathbf{t}_\alpha\|}{\sigma_\alpha}))^2 \qquad (2.14)$$

Notice that assumption that the regression function

$$f_0(\mathbf{x}) \equiv E[y|\mathbf{x}]$$

belongs to the class $\mathcal{F}$ correspondingly implies an assumption on the probability distribution $P(y|\mathbf{x})$, viz., that P must be such that $E[y|\mathbf{x}]$ belongs to $\mathcal{F}$. Notice also that since we assumed that Y is a closed interval, we are implicitly assuming that $P(y|\mathbf{x})$ has compact support.

Assuming now that we have been able to solve the minimization problem of eq. (2.14), the main question we are interested in is "how far is $\hat{f}_{n,l}$ from f_0?". We give an answer in the next section.

2.4 Main Result

The main theorem is:

Theorem 2.4.1 *For any $0 < \delta < 1$, for n nodes, l data points, input dimensionality of k, and $H_n, \mathcal{F}, f_0, \hat{f}_{n,l}$ also as defined in the statement of the problem above, with probability greater than $1 - \delta$,*

$$\|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)} \leq O\left(\frac{1}{n}\right) + O\left(\left[\frac{nk\ln(nl) - \ln\delta}{l}\right]^{1/2}\right)$$

Proof: The proof requires us to go through a series of propositions and lemmas which have been relegated to appendix (2-D) for continuity of ideas.□

2.5 Remarks

There are a number of comments we would like to make on the formulation of our problem and the result we have obtained. There is a vast body of literature on approximation theory and the theory of empirical risk minimization. In recent times, some of the results in these areas have been applied by the computer science and neural network community to study formal learning models. Here we would like to make certain observations about our result, suggest extensions and future work, and to make connections with other work done in related areas.

2.5.1 Observations on the Main Result

- The theorem has a PAC (Valiant, 1984) like setting. It tells us that if we draw enough data points (labelled examples) and have enough nodes in our Radial Basis Functions network, we can drive our error arbitrarily close to zero with arbitrarily high probability. Note however that our result is not entirely distribution-free. Although no assumptions are made on the form of the underlying distribution, we do have certain constraints on the kinds of distributions for which this result holds. In particular, the distribution is such that its conditional mean $E[y|\mathbf{x}]$ (this is also the regression function $f_0(x)$) must belong to a the class of functions $\mathcal{F}$ defined by eq. (2.12). Further the distribution $P(y|\mathbf{x})$ must have compact support [10].

- The error bound consists of two parts, one ($O(1/n)$) coming from approximation theory, and the other $O(((nk\ln(nl) + \ln(1/\delta))/l)^{1/2})$ from statistics. It is noteworthy that for a given approximation scheme (corresponding to $\{H_n\}$), a certain class of functions (corresponding to $\mathcal{F}$) suggests itself. So we have gone from the class of networks to the class of problems they can perform as opposed to the other way around, i.e., from a class of problems to an optimal class of networks.

[10] This condition, that is related to the problem of large deviations , could be relaxed, and will be subject of further investigations.

- This sort of a result implies that if we have the prior knowledge that f_0 belongs to class $\mathcal{F}$, then by choosing the number of data points, l, and the number of basis functions, n, appropriately, we can drive the misclassification error arbitrarily close to Bayes rate. In fact, for a fixed amount of data, even before we have started looking at the data, we can pick a starting architecture, i.e., the number of nodes. n, for optimal performance. After looking at the data, we might be able to do some structural risk minimization (Vapnik, 1982) to further improve architecture selection. For a fixed architecture, this result sheds light on how much data is required for a certain error performance. Moreover, it allows us to choose the number of data points and number of nodes simultaneously for guaranteed error performances. Section 2.6 explores this question in greater detail.

2.5.2 Extensions

- There are certain natural extensions to this work. We have essentially proved the consistency of the estimated network function $\hat{f}_{n,l}$. In particular we have shown that $\hat{f}_{n,l}$ converges to f_0 with probability 1 as l and n grow to infinity. It is also possible to derive conditions for almost sure convergence. Further, we have looked at a specific class of networks ($\{H_n\}$) which consist of weighted sums of Gaussian basis functions with moving centers but fixed variance. This kind of an approximation scheme suggests a class of functions $\mathcal{F}$ which can be approximated with guaranteed rates of convergence as mentioned earlier. We could prove similar theorems for other kinds of basis functions which would have stronger approximation properties than the class of functions considered here. The general principle on which the proof is based can hopefully be extended to a variety of approximation schemes.

- We have used notions of metric entropy and covering number (Dudley, 1987; Pollard, 1984) in obtaining our uniform convergence results. Haussler (1989) uses the results of Pollard and Dudley to obtain uniform convergence results and our techniques closely follow his approach. It should be noted here that Vapnik deals with exactly the same question and uses the VC-dimension instead. It would be interesting to compute the VC-dimension of the class of networks and use it to obtain our results.

- While we have obtained an upper bound on the error in terms of the number of nodes and examples, it would be worthwhile to obtain lower bounds on the same. Such lower bounds do not seem to exist in the neural network literature to the best of our knowledge.

- We have considered here a situation where the estimated network i.e., $\hat{f}_{n,l}$ is obtained by minimizing the empirical risk over the class of functions H_n. Very often, the estimated network is obtained by minimizing a somewhat different objective function which consists of two parts. One is the fit to the data and the other is some complexity term which favors less complex (according to the defined notion of complexity) functions over more complex ones. For example the regularization approach (Tikhonov, 1963; Poggio and Girosi, 1992; Wahba, 1990) minimizes a cost function of the form

$$H[f] = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i) + \lambda\Phi[f]$$

  over the class $H = \cup_{n\geq 1} H_n$. Here λ is the so called "regularization parameter" and $\Phi[f]$ is a functional which measures smoothness of the functions involved. It would be interesting to obtain convergence conditions and rates for such schemes. Choice of an optimal λ is an interesting question in regularization techniques and typically cross-validation or other heuristic schemes are used. A result on convergence rate potentially offers a principled way to choose λ.

- Structural risk minimization is another method to achieve a trade-off between network complexity (corresponding to n in our case) and fit to data. However it does not guarantee that the architecture selected will be the one with minimal parameterization[11]. In fact, it would be of some interest to develop a sequential growing scheme. Such a technique would at any stage perform a sequential hypothesis test (Govindarajulu, 1975) . It would then decide whether to ask for more data, add one more node or simply stop and output the function it has as its ϵ-good hypothesis. In such a process, one might even incorporate active learning (Angluin, 1988) so that if the algorithm asks for more data, then it might even specify a region in the input domain from where it would like to see this data. It is conceivable that such a scheme would grow to minimal parameterization (or closer to it at any rate) and require less data than classical structural risk minimization.

[11] Neither does regularization for that matter. The question of minimal parameterization is related to that of order determination of systems, a very difficult problem!

- It should be noted here that we have assumed that the empirical risk $\sum_{i=1}^{l}(y_i - f(x_i))^2$ can be minimized over the class H_n and the function $\hat{f}_{n,l}$ be effectively computed. While this might be fine in principle, in practice only a locally optimal solution to the minimization problem is found (typically using some gradient descent schemes). The computational complexity of obtaining even an approximate solution to the minimization problem is an interesting one and results from computer science (Judd, 1988; Blum and Rivest, 1988) suggest that it might in general be NP-hard.

2.5.3 Connections with Other Results

- In the neural network and computational learning theory communities results have been obtained pertaining to the issues of generalization and learnability. Some theoretical work has been done (Baum and Haussler, 1989; Haussler, 1989; Ji and Psaltis, 1992) in characterizing the sample complexity of finite sized networks. Of these, it is worthwhile to mention again the work of Haussler from which this chapter derives much inspiration. He obtains bounds for a fixed hypothesis space i.e. a fixed finite network architecture. Here we deal with families of hypothesis spaces using richer and richer hypothesis spaces as more and more data becomes available. Later we will characterize the trade-off between hypothesis complexity and error rate. Others (Levin, Tishby, and Solla, 1990; Opper, and Haussler, 1991) attempt to characterize the generalization abilities of feed-forward networks using theoretical formalizations from statistical mechanics. Yet others (Botros, and Atkeson, 1991; Moody, 1992; Cohn and Tesauro, 1991; Weigand, Rumelhart, and Huberman, 1991) attempt to obtain empirical bounds on generalization abilities.

- This is an attempt to obtain rate-of-convergence bounds in the spirit of Barron's work , but using a different approach. We have chosen to combine theorems from approximation theory (which gives us the $O(1/n)$ term in the rate, and uniform convergence theory (which gives us the other part). Note that at this moment, our rate of convergence is worse than Barron's. In particular, he obtains a rate of convergence of $O(1/n + (nk\ln(l))/l)$. Further, he has a different set of assumptions on the class of functions (corresponding to our $\mathcal{F}$). Finally, the approximation scheme is a class of networks with sigmoidal units as opposed to radial-basis units and a different proof technique is used. It should be mentioned here that his proof relies on a discretization of the networks into a countable family, while no such assumption is made here.

- It would be worthwhile to make a reference to (Geman, Bienenstock, and Doursat, 1992) which talks of the Bias-Variance dilemma. This is another way of formulating the trade-off between the approximation error and the estimation error. As the number of parameters (proportional to n) increases, the bias (which can be thought of as analogous to the approximation error) of the estimator decreases and its variance (which can be thought of as analogous to the estimation error) increases for a fixed size of the data set. Finding the right bias-variance trade-off is very similar in spirit to finding the trade-off between network complexity and data complexity.

- Given the class of radial basis functions we are using, a natural comparison arises with kernel regression (Krzyzak, 1986; Devroye, 1981) and results on the convergence of kernel estimators. It should be pointed out that, unlike our scheme, Gaussian-kernel regressors require the variance of the Gaussian to go to zero as a function of the data. Further the number of kernels is always equal to the number of data points and the issue of trade-off between the two is not explored to the same degree.

- In our statement of the problem, we discussed how pattern classification could be treated as a special case of regression. In this case the function f_0 corresponds to the Bayes *a-posteriori* decision function. Researchers (Richard, and Lippman, 1991; Hampshire, and Pearlmutter, 1990; Gish, 1990) in the neural network community have observed that a network trained on a least square error criterion and used for pattern classification was in effect computing the Bayes decision function. This chapter provides a rigorous proof of the conditions under which this is the case.

2.6 Implications of the Theorem in Practice: Putting In the Numbers

We have stated our main result in a particular form. We have provided a provable upper bound on the error (in the $\| \cdot \|_{L^2(P)}$ metric) in terms of the number of examples and the number of basis functions used. Further we have provided the order of the convergence and have not stated the constants involved. The same result could be stated in other forms and has certain implications. It provides us rates at which the number of basis functions (n) should increase as a function of the number of examples (l) in order to guarantee convergence(Section 2.6.1). It also provides us with the trade-offs between the two as explored in Section 2.6.2.

2.6.1 Rate of Growth of n for Guaranteed Convergence

From our theorem (2.4.1) we see that the generalization error converges to zero only if n goes to infinity more slowly than l. In fact, if n grows too quickly the estimation error $\omega(l, n, \delta)$ will diverge, because it is proportional to n. In fact, setting $n = l^r$, we obtain

$$\lim_{l \to +\infty} \omega(l, n, \delta) =$$

$$= \lim_{l \to +\infty} O\left(\left[\frac{l^r k \ln(l^{r+1}) + \ln(1/\delta)}{l}\right]^{1/2}\right) =$$

$$= \lim_{l \to +\infty} l^{r-1} \ln l \ .$$

Therefore the condition $r < 1$ should hold in order to guarantee convergence to zero.

2.6.2 Optimal Choice of n

In the previous section we made the point that the number of parameters n should grow more slowly than the number of data points l, in order to guarantee the consistency of the estimator $\hat{f}_{n,l}$. It is quite clear that there is an *optimal* rate of growth of the number of parameters, that, for any fixed amount of data points l, gives the best possible performance with the least number of parameters. In other words, for any fixed l there is an optimal number of parameters $n^*(l)$ that minimizes the generalization error. That such a number should exist is quite intuitive: for a fixed number of data, a small number of parameters will give a low estimation error $\omega(l, n, \delta)$, but very high approximation error $\varepsilon(n)$, and therefore the generalization error will be high. If the number of parameters is very high the approximation error $\varepsilon(n)$ will be very small, but the estimation error $\omega(l, n, \delta)$ will be high, leading to a large generalization error again. Therefore, somewhere in between there should be a number of parameters high enough to make the approximation error small, but not too high, so that these parameters can be estimated reliably, with a small estimation error. This phenomenon is evident from figure (2-7), where we plotted the generalization error as a function of the number of parameters n for various choices of sample size l. Notice that for a fixed sample size, the error passes through a minimum. Notice that the location of the minimum shifts to the right when the sample size is increased.

In order to find out exactly what is the optimal rate of growth of the network size we simply find the minimum of the generalization error as a function of n keeping the sample size l fixed. Therefore we have to solve the equation:

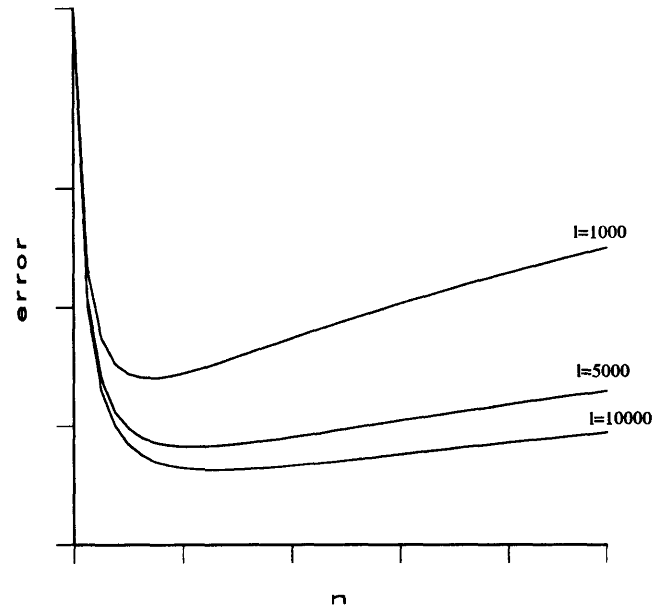

Figure 2-7: Bound on the generalization error as a function of the number of basis functions n keeping the sample size l fixed. This has been plotted for a few different choices of sample size. Notice how the generalization error goes through a minimum for a certain value of n. This would be an appropriate choice for the given (constant) data complexity. Note also that the minimum is broader for larger l, that is, an accurate choice of n is less critical when plenty of data is available.

$$\frac{\partial}{\partial n}E[(f_0 - \hat{f}_{n,l})^2] = 0$$

for n as a function of l. Substituting the bound given in theorem (2.4.1) in the previous equation, and setting all the constants to 1 for simplicity, we obtain:

$$\frac{\partial}{\partial n}\left[\frac{1}{n} + (\frac{nk\ln(nl) - \ln(\delta)}{l})^{\frac{1}{2}}\right] = 0 \ .$$

Performing the derivative the expression above can be written as

$$\frac{1}{n^2} = \frac{1}{2}\left[\frac{kn\ln(nl) - \ln\delta}{l}\right]^{-\frac{1}{2}}\frac{k}{l}[\ln(nl) + 1] \ .$$

We now make the assumption that l is big enough to let us perform the approximation $\ln(nl) + 1 \approx \ln(nl)$. Moreover, we assume that

$$\frac{1}{\delta} << (nl)^{nk}$$

in such a way that the term including δ in the equation above is negligible. After some algebra we therefore conclude that the optimal number of parameters $n^*(l)$ satisfies, for large l, the equation:

$$n^*(l) = \left[\frac{4l}{k\ln(n^*(l)l)}\right]^{\frac{1}{3}} \ .$$

From this equation is clear that n^* is roughly proportional to a power of l, and therefore we can neglect the factor n^* in the denominator of the previous equation, since it will only affect the result by a multiplicative constant. Therefore we conclude that the optimal number of parameters $n^*(l)$ for a given number of examples behaves as

$$n^*(l) \propto \left[\frac{l}{k\ln l}\right]^{\frac{1}{3}} \ . \qquad (2.15)$$

In order to show that this is indeed the optimal rate of growth we reported in figure (2-8) the generalization error as function of the number of examples l for different rate of growth of n, that is setting $n = l^r$ for different values of r. Notice that the exponent $r = \frac{1}{3}$, that is very similar to the optimal rate of eq. (2.15), performs better than larger ($r = \frac{1}{2}$) and smaller ($r = \frac{1}{10}$) exponents.

While a fixed sample size suggests the scheme above for choosing an optimal network size, it is important to note that for a certain confidence rate (δ) and for a fixed error rate (ϵ), there are various choices of n and l which are satisfactory. Fig. 2-9 shows n

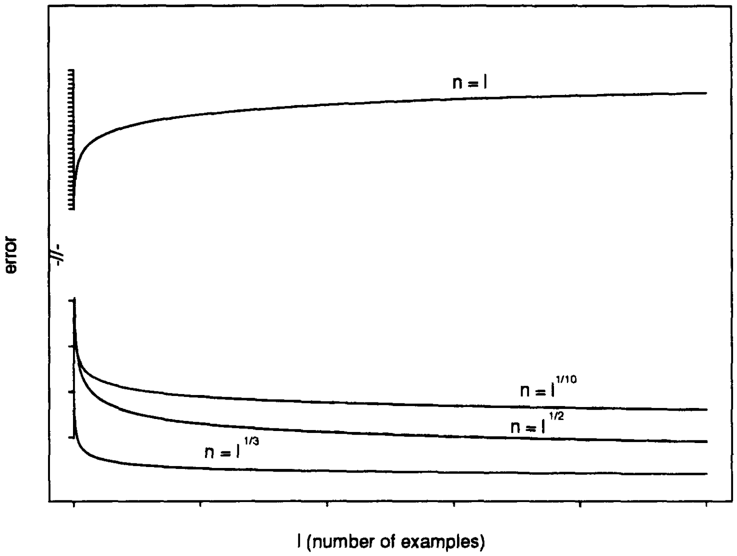

Figure 2-8: The bound on the generalization error as a function of the number of examples for different choices of the rate at which network size n increases with sample size l. Notice that if $n = l$, then the estimator is not guaranteed to converge, i.e., the bound on the generalization error diverges. While this is a distribution free-upper bound, we need distribution-free lower bounds as well to make the stronger claim that $n = l$ will never converge.

as a function of l, in other words (l, n) pairs which yield the same error rate with the same confidence.

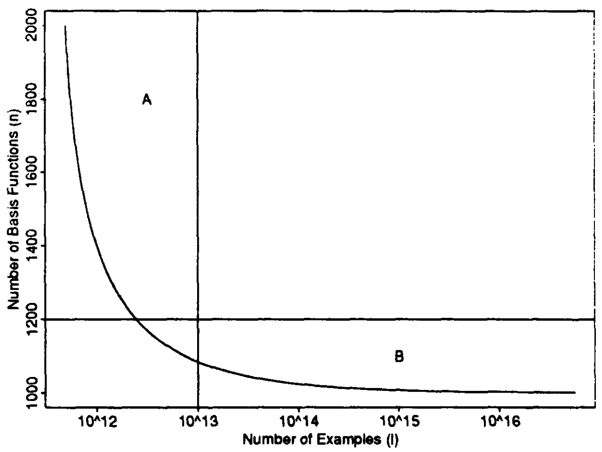

Figure 2-9: This figures shows various choices of (l, n) which give the same generalization error. The x-axis has been plotted on a log scale. The interesting observation is that there are an infinite number of choices for number of basis functions and number of data points all of which would guarantee the same generalization error (in terms of its worst case bound).

If data are expensive for us, we could operate in region A of the curve. If network size is expensive we could operate in region B of the curve. In particular the economics of trading off network and data complexity would yield a suitable point on this curve and thus would allow us to choose the right combination of n and l to solve our regression problem with the required accuracy and confidence.

Of course we could also plot the error as a function of data size l for a fixed network size (n) and this has been done for various choices of n in Fig. 2-10.
We see as expected that the error monotonically decreases as a function of l. However it asymptotically decreases not to the Bayes error rate but to some value above it (the approximation error) which depends upon the the network complexity.

Finally figure (2-11) shows the result of theorem (2.4.1) in a 3-dimensional plot. The generalization error, the network size, and the sample size are all plotted as a function of each other.

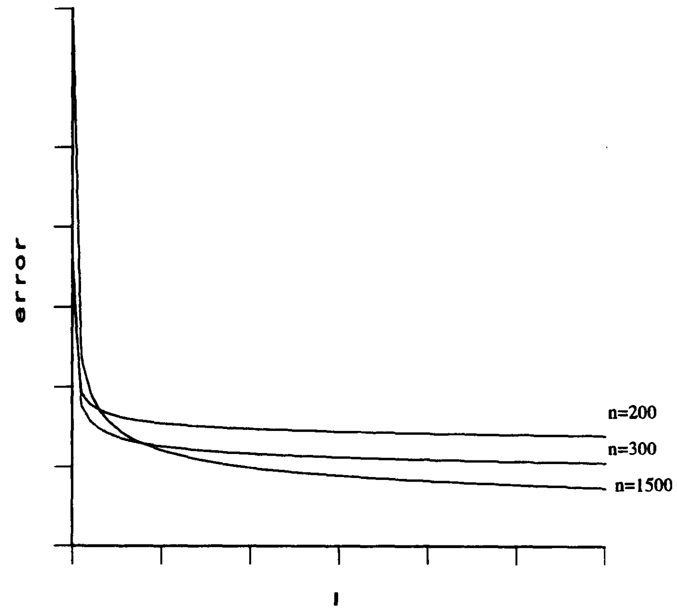

Figure 2-10: The generalization error as a function of number of examples keeping the number of basis functions (n) fixed. This has been done for several choices of n. As the number of examples increases to infinity the generalization error asymptotes to a minimum which is not the Bayes error rate because of finite hypothesis complexity (finite n).

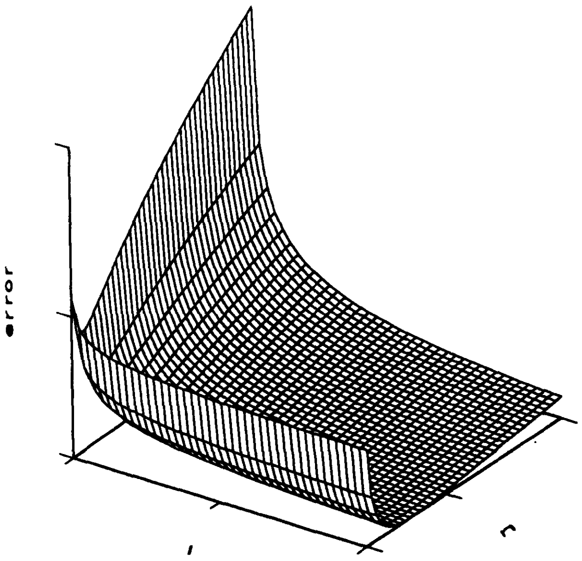

Figure 2-11: The generalization error, the number of examples (l) and the number of basis functions (n) as a function of each other.

2.7 Conclusion

For the task of learning some unknown function from labelled examples where we have multiple hypothesis classes of varying complexity, choosing the class of right complexity and the appropriate hypothesis within that class poses an interesting problem. We have provided an analysis of the situation and the issues involved and in particular have tried to show how the hypothesis complexity, the sample complexity and the generalization error are related. We proved a theorem for a special set of hypothesis classes, the radial basis function networks and we bound the generalization error for certain function learning tasks in terms of the number of parameters and the number of examples. This is equivalent to obtaining a bound on the rate at which the number of parameters must grow with respect to the number of examples for convergence to take place. Thus we use richer and richer hypothesis spaces as more and more data become available. We also see that there is a tradeoff between hypothesis complexity and generalization error for a certain fixed amount of data and our result allows us a principled way of choosing an appropriate hypothesis complexity (network architecture). The choice of an appropriate model for empirical data is a problem of long-standing interest in statistics and we provide connections between our work and other work in the field.

2-A Notations

- $\mathcal{A}$: a set of functions defined on S such that, for any $a \in \mathcal{A}$,

$$0 \leq a(\xi) \leq U^2 \quad \forall \xi \in S \ .$$

- $\mathcal{A}_{\vec{\xi}}$: the restriction of $\mathcal{A}$ to the data set, see eq. (2.23).
- $\mathcal{B}$: it will usually indicate the set of all possible l-dimensional Boolean vectors.
- B: a generic ϵ-separated set in S.
- $\mathcal{C}(\epsilon, \mathcal{A}, d_{L^1})$: the metric capacity of a set $\mathcal{A}$ endowed with the metric $d_{L^1(P)}$.
- $d(\cdot,\cdot)$: a metric on a generic metric space S.
- $d_{L^1}(\cdot,\cdot)$, $d_{L^1(P)}(\cdot,\cdot)$: L^1 metrics in vector spaces. The definition depends on the space on which the metric is defined (k-th dimensional vectors, real valued functions, vector valued functions).

  1. In a vector space R^k we have

  $$d_{L^1}(\mathbf{x},\mathbf{y}) = \frac{1}{l}\sum_{\mu=1}^{l} |x^\mu - y^\mu|$$

  where $\mathbf{x}$, $\mathbf{y} \in R^k$, x^μ and y^μ denote their μ-th components.

  2. In an infinite dimensional space $\mathcal{F}$ of real valued functions in k variables we have

  $$d_{L^1(P)}(f,g) = \int_{R^k} |f(\mathbf{x}) - g(\mathbf{x})| dP(\mathbf{x})$$

  where $f, g \in \mathcal{F}$ and $dP(\mathbf{x})$ is a probability measure on R^k.

  3. In an infinite dimensional space $\mathcal{F}$ of functions in k variables with values in R^n we have

  $$d_{L^1(P)}(\mathbf{f},\mathbf{g}) = \frac{1}{n}\sum_{i=1}^{n}\int_{R^k} |f_i(\mathbf{x}) - g_i(\mathbf{x})| dP(\mathbf{x})$$

  where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots f_i(\mathbf{x}), \ldots f_n(\mathbf{x}))$, $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots g_i(\mathbf{x}), \ldots g_n(\mathbf{x}))$ are elements of $\mathcal{F}$ and $dP(\mathbf{x})$ is a probability measure on R^k.

- D_l: it will always indicate a data set of l points:

$$D_l \equiv \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{l} .$$

The points are drawn according to the probability distribution $P(\mathbf{x}, y)$.

- $E[\cdot]$: it denotes the expected value with respect to the probability distribution $P(\mathbf{x}, y)$. For example

$$I[f] = E[(y - f(\mathbf{x}))^2] ,$$

and

$$\|f_0 - f\|^2_{L^2(P)} = E[(f_0(\mathbf{x}) - f(\mathbf{x}))^2] .$$

- f: a generic estimator, that is any function from X to Y:

$$f : X \Rightarrow Y .$$

- $f_0(\mathbf{x})$: the regression function, it is the conditional mean of the response given the predictor:

$$f_0(\mathbf{x}) \equiv \int_Y dy \; yP(y|\mathbf{x}) .$$

It can also be defined as the function that minimizes the expected risk $I[f]$ in $\mathcal{U}$, that is

$$f_0(\mathbf{x}) \equiv \arg \inf_{f \in \mathcal{U}} I[f] .$$

Whenever the response is obtained sampling a function h in presence of zero mean noise the regression function coincides with the sampled function h.

- f_n: it is the function that minimizes the expected risk $I[f]$ in H_n:

$$f_n \equiv \arg \inf_{f \in H_n} I[f]$$

Since

$$I[f] = \|f_0 - f\|^2_{L^2(P)} + I[f_0]$$

f_n it is also the best $L^2(P)$ approximation to the regression function in H_n (see figure 2-6).

- $\hat{f}_{n,l}$: is the function that minimizes the empirical risk $I_{\rm emp}[f]$ in H_n:

  $$\hat{f}_{n,l} \equiv \arg \inf_{f \in H_n} I_{\rm emp}[f]$$

  In the neural network language it is the output of the network after training has occurred.

- $\mathcal{F}$: the space of functions to which the regression function belongs, that is the space of functions we want to approximate.

  $$\mathcal{F} : X \Rightarrow Y$$

  where $X \in R^d$ and $Y \in R$. $\mathcal{F}$ could be for example a set of differentiable functions, or some Sobolev space $H^{m,p}(R^k)$

- $\mathcal{G}$: it is a class of functions of k variables

  $$g : R^k \rightarrow [0, V]$$

  defined as

  $$\mathcal{G} == \{g : g(\mathbf{x}) = G(\|\mathbf{x} - \mathbf{t}\|),\ \mathbf{t} \in R^k\}.$$

  where G is the gaussian function.

- G_1: it is a $k+2$-dimensional vector space of functions from R^k to R defined as

  $$G_1 \equiv \text{span}\{1, x^1, x^2, \cdot, x^k, \|\mathbf{x}\|^2\}$$

  where $\mathbf{x} \in R^k$ and x^μ is the μ-th component of the vector $\mathbf{x}$.

- G_2: it is a set of real valued functions in k variables defined as

  $$G_2 = \{\alpha e^{-f} : f \in G_1,\quad \alpha = \frac{1}{\sqrt{2\pi}\sigma}\}$$

where σ is the standard deviation of the Gaussian G.

- H_I: it is a class of vector valued functions

$$\mathbf{g}(\mathbf{x}) : R^k \to R^n$$

of the form

$$\mathbf{g}(\mathbf{x}) = (G(\|\mathbf{x} - \mathbf{t}_1\|), G(\|\mathbf{x} - \mathbf{t}_2\|), \ldots, G(\|\mathbf{x} - \mathbf{t}_n\|))$$

where G is the gaussian function and the $\mathbf{t}_i$ are arbitrary k-dimensional vectors.

- H_F: it is a class of real valued functions in n variables:

$$f : [0, V]^n \to R$$

of the form

$$f(\mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{x}$$

where $\boldsymbol{\beta} \equiv (\beta_1, \ldots, \beta_n)$ is an arbitrary n-dimensional vector that satisfies the constraint

$$\sum_{i=1}^{n} |\beta_i| \leq M \ .$$

- H_n: a subset of $\mathcal{F}$, whose elements are parametrized by a number of parameters proportional to n. We will assume that the sets H_n form a nested family, that is

$$H_1 \subset H_2 \subset \ldots \subset H_n \subset \ldots \ .$$

For example H_n could be the set of polynomials in one variable of degree $n - 1$, Radial Basis Functions with n centers or multilayer perceptrons with n hidden units. Notice that for Radial Basis Functions with moving centers and Multilayer perceptrons the number of parameters of an element of H_n is not n, but it is proportional to n (respectively $n(k+1)$ and $n(k+2)$, where k is the number of variables).

- H: it is defined as $H = \bigcup_{n=1}^{\infty} H_n$, and it is identified with the approximation scheme. If H_n is the set of polynomials in one variable of degree $n-1$, H is the set of polynomials of any degree.

- $H^{m,p}(R^k)$: the Sobolev space of functions in k variables whose derivatives up to order m are in $L^p(R^k)$.

- $I[f]$: the expected risk, defined as

$$I[f] \equiv \int_{X \times Y} d\mathbf{x}dy \; P(\mathbf{x}, y)(y - f(\mathbf{x}))^2 \; .$$

  where f is any function for which this expression is well defined. It is a measure of how well the function f predicts the response y.

- $I_{\text{emp}}[f]$: the empirical risk. It is a functional on $\mathcal{U}$ defined as

$$I_{\text{emp}}[f] \equiv \frac{1}{l} \sum_{i=1}^{l} (y_i - f(\mathbf{x}_i))^2 \; ,$$

  where $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ is a set of data randomly drawn from $X \times Y$ according to the probability distribution $P(\mathbf{x}, y)$. It is an approximate measure of the expected risk, since it converges to $I[f]$ in probability when the number of data points l tends to infinity.

- k: it will always indicate the number of independent variables, and therefore the dimensionality of the set X.

- l: it will always indicate the number of data points drawn from X according to the probability distribution $P(\mathbf{x})$.

- $L^2(P)$: the set of function whose square is integrable with respect to the measure defined by the probability distribution P. The norm in $L^2(P)$ is therefore defined by

$$\|f\|^2_{L^2(P)} \equiv \int_{R^k} d\mathbf{x} \; P(\mathbf{x}) f^2(\mathbf{x}) \; .$$

- $\Lambda^m(R^k)(M_0, M_1, M_2, \ldots, M_m)$: the space of functions in k variables whose derivatives up to order m are bounded:

$$|D^\alpha f| \leq M_{|\alpha|} \quad |\alpha| = 1, 2, \ldots, m$$

where α is a multi-index.

- M: a bound on the coefficients of the gaussian Radial Basis Functions technique considered in this paper, see eq. (2.13).
- $\mathcal{M}(\epsilon, \mathcal{S}, d)$: the packing number of the set $\mathcal{S}$, with metric d.
- $\mathcal{N}(\epsilon, \mathcal{S}, d)$: the covering number of the set $\mathcal{S}$, with metric d.
- n: a positive number proportional to the number of parameters of the approximating function. Usually will be the number of basis functions for the RBF technique or the number of hidden units for a multilayer perceptron.
- $P(\mathbf{x})$: a probability distribution defined on X. It is the probability distribution according to which the data are drawn from X.
- $P(y|\mathbf{x})$: the conditional probability of the response y given the predictor $\mathbf{x}$. It represents the probabilistic dependence of y from $\mathbf{x}$. If there is no noise in the system it has the form $P(y|\mathbf{x}) = \delta(y - h(\mathbf{x}))$, for some function h, indicating that the predictor $\mathbf{x}$ uniquely determines the response y.
- $P(\mathbf{x}, y)$: the joint distribution of the predictors and the response. It is a probability distribution on $X \times Y$ and has the form

$$P(\mathbf{x}, y) \equiv P(\mathbf{x})P(y|\mathbf{x}) \ .$$

- S: it will usually denote a metric space, endowed with a metric d.
- $\mathcal{S}$: a generic subset of a metric space S.
- $\mathcal{T}$: a generic ϵ-cover of a subset $\mathcal{S} \subset S$.
- U: it gives a bound on the elements of the class $\mathcal{A}$. In the specific case of the class $\mathcal{A}$ considere in the proof we have $U = 1 + MV$.
- $\mathcal{U}$: the set of all the functions from X to Y for which the expected risk is well defined.
- V: a bound on the Gaussian basis function G:

$$0 \leq G(\mathbf{x}) \leq V \ , \quad \forall \mathbf{x} \in R^k \ .$$

- X: a subset of R^k, not necessarily proper. It is the set of the independent variables, or predictors, or, in the language of neural networks, input variables.
- $\mathbf{x}$: a generic element of X, and therefore a k-dimensional vector (in the neural network language is the input vector).
- Y: a subset of R, whose elements represent the response variable, that in the neural networks language is the output of the network. Unless otherwise stated it will be assumed to be compact, implying that $\mathcal{F}$ is a set of bounded functions. In pattern recognition problem it is simply the set $\{0, 1\}$.
- y: a generic element of Y, it denotes the response variable.

2-B A Useful Decomposition of the Expected Risk

We now show that the function that minimizes the expected risk

$$I[f] = \int_{X \times Y} P(\mathbf{x}, y) d\mathbf{x} dy (y - f(\mathbf{x}))^2 \ .$$

is the regression function defined in eq. (2.3). It is sufficient to add and subtract the regression function in the definition of expected risk:

$$\begin{aligned} I[f] &= \int_{X \times Y} d\mathbf{x} dy P(\mathbf{x}, y)(y - f_0(\mathbf{x}) + f_0(\mathbf{x}) - f(\mathbf{x}))^2 = \\ &= \int_{X \times Y} d\mathbf{x} dy P(\mathbf{x}, y)(y - f_0(\mathbf{x}))^2 + \\ &+ \int_{X \times Y} d\mathbf{x} dy P(\mathbf{x}, y)(f_0(\mathbf{x}) - f(\mathbf{x}))^2 + \\ &+ 2 \int_{X \times Y} d\mathbf{x} dy P(\mathbf{x}, y)(y - f_0(\mathbf{x}))(f_0(\mathbf{x}) - f(\mathbf{x})) \end{aligned}$$

By definition of the regression function $f_0(\mathbf{x})$, the cross product in the last equation is easily seen to be zero, and therefore

$$I[f] = \int_X d\mathbf{x} P(\mathbf{x})(f_0(\mathbf{x}) - f(\mathbf{x}))^2 + I[f_0] \ .$$

Since the last term of $I[f]$ does not depend on f, the minimum is achieved when the first term is minimum, that is when $f(\mathbf{x}) = f_0(\mathbf{x})$.

In the case in which the data come from randomly sampling a function f in presence of additive noise, ϵ, with probability distribution $\mathcal{P}(\epsilon)$ and zero mean, we have $P(y|\mathbf{x}) = \mathcal{P}(y - f(\mathbf{x}))$ and then

$$I[f_0] = \int_{X\times Y} d\mathbf{x}dy\, P(\mathbf{x},y)(y - f_0(\mathbf{x}))^2 = \tag{2.16}$$

$$= \int_X d\mathbf{x}P(\mathbf{x}) \int_Y (y - f(\mathbf{x}))^2 \mathcal{P}(y - f(\mathbf{x})) = \tag{2.17}$$

$$= \int_X d\mathbf{x}P(\mathbf{x}) \int_Y \epsilon^2 \mathcal{P}(\epsilon) d\epsilon \;=\; \sigma^2 \tag{2.18}$$

where σ^2 is the variance of the noise. When data are noisy, therefore, even in the most favourable case we cannot expect the expected risk to be smaller than the variance of the noise.

2-C A Useful Inequality

Let us assume that, with probability $1 - \delta$ a uniform bound has been established:

$$|I_{\text{emp}}[f] - I[f]| \le \omega(l, n, \delta) \quad \forall f \in H_n \ .$$

We want to prove that the following inequality also holds:

$$|I[f_n] - I[\hat{f}_{n,l}]| \le 2\omega(l, n, \delta) \ . \tag{2.19}$$

This fact is easily established by noting that since the bound above is uniform, then it holds for both f_n and $\hat{f}_{n,l}$, and therefore the following inequalities hold:

$$I[\hat{f}_{n,l}] \le I_{\text{emp}}[\hat{f}_{n,l}] + \omega$$

$$I_{\text{emp}}[f_n] \le I[f_n] + \omega$$

Moreover, by definition, the two following inequalities also hold:

$$I[f_n] \le I[\hat{f}_{n,l}]$$

$$I_{\text{emp}}[\hat{f}_{n,l}] \le I_{\text{emp}}[f_n]$$

Therefore tha following chain of inequalities hold, proving inequality (2.19):

$$I[f_n] \le I[\hat{f}_{n,l}] \le I_{\text{emp}}[\hat{f}_{n,l}] + \omega \le I_{\text{emp}}[f_n] + \omega \le I[f_n] + 2\omega \ .$$

An intutitive explanation of these inequalities is also explained in figure (2-12).

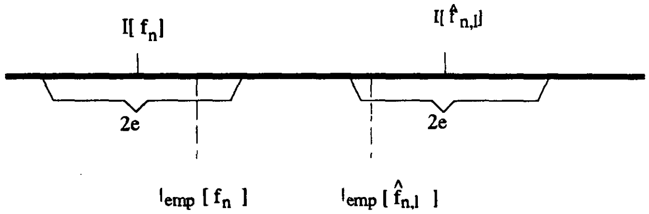

Figure 2-12: If the distance between $I[f_n]$ and $I[\hat{f}_{n,l}]$ is larger than 2ϵ, the condition $I_{\text{emp}}[\hat{f}_{n,l}] \leq I_{\text{emp}}[f_n]$ is violated.

2-D Proof of the Main Theorem

The theorem will be proved in a series of steps. Conceptually, there are four major steps in the proof outlined in the proof structure below.

Structure of Proof

Step 1

The total generalization error is decomposed into its approximation and estimation components. Using the derivations outlined in appendices 2-B, and 2-C, we are able to show that the decomposition has the form of statement 2.2.1 of section 2.2, viz., with probability $1-\delta$,

$$\|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)} \leq \varepsilon(n) + 2\omega(l,n,\delta) \ . \tag{2.20}$$

We now need to compute $\varepsilon(n)$ and $\omega(l,n,\delta)$ and these constitute steps 2 and 3 of the proof structure.

Step 2

We obtain a bound on $\varepsilon(n)$ (the approximation error) in section 2-D.1. The fundamental lemma used here is the Maurey-Jones-Barron lemma (Lemma 2-D.1) and the approximation bound is obtained.

Step 3

We obtain a bound on the estimation error $\omega(l,n,\delta)$ in section 2-D.2. Recall that we need to be able to prove a uniform law of large numbers of the form:

$$\forall f \ \in \ H_n, \ |I[f] - I_emp[f]| \leq \omega(l,n,\delta)$$

with probability greater than $1 - \delta$.

Starting with a uniform law of the form stated in Claim 2-D.1 and refining it further we arrive at Claim 2-D.3. In doing this, we introduce notions of *covering numbers* and *metric entropy*. The form of this refined uniform law of large numbers is:

$$P(\forall h \in H_n, |I_{\text{emp}}[h] - I[h]| \leq \epsilon) \geq$$

$$\geq 1 - 4\mathcal{C}(\epsilon/16, \mathcal{A}, d_{L^1})]e^{-\frac{1}{128U^4}\epsilon^2 l} .$$

In order to let $1 - 4\mathcal{C}(\epsilon/16, \mathcal{A}, d_{L^1})]e^{-\frac{1}{128U^4}\epsilon^2 l}$ be greater than $1 - \delta$, we need to obtain an expression for $\mathcal{C}(\epsilon/16, \mathcal{A}, d_{L^1})]$ in terms of the number of parameters. Claims 2-D.4 through 2-D.9 go through this computation.

Finally, in claim 2-D.10, we show how to use this result to compute an expression for $\omega(l, n, \delta)$ which is what we originally set out to do.

Step 4

Putting together the approximation and estimation bounds of steps 2 and 3, we obtain in section 2-D.3 how the expression for the total generalization error in the appropriate form in order to prove the main theorem.

2-D.1 Bounding the approximation error

In this part we attempt to bound the approximation error. In section 2.3 we assumed that the class of functions to which the regression function belongs, that is the class of functions that we want to approximate, is

$$\mathcal{F} \equiv \{f | f = \lambda * G_m, m > k/2, |\lambda|_{R^k} \leq M\} .$$

where λ is a signed Radon measure on the Borel sets of R^k, G_m is the Bessel-Macdonald kernel as defined in section 2.3 and M is a positive real number. Our approximating family is the class:

$$H_n = \{f \in L_2 | f = \sum_{i=1}^{n} \beta_i G(\frac{\|\mathbf{x} - \mathbf{t}_i\|}{\sigma_i}),\ \sum_{i=1}^{n} |\beta_i| \leq M\ ,\quad \mathbf{t}_i \in R^k\}$$

It has been shown in [50, 51] that the class H_n uniformly approximate elements of $\mathcal{F}$, and that the following bound is valid:

$$E[(f_0 - f_n)^2] \leq O\left(\frac{1}{n}\right) . \tag{2.21}$$

This result is based on a lemma by Jones [71] on the convergence rate of an iterative approximation scheme in Hilbert spaces. A formally similar lemma, brought to our attention by R. Dudley [38] is due to Maurey and was published by Pisier [105]. Here we report a version of the lemma due to Barron [8, 9] that contains a slight refinement of Jones' result:

Lemma 2-D.1 (Maurey-Jones-Barron) *If f is in the closure of the convex hull of a set $\mathcal{G}$ in a Hilbert space H with $\|g\| \leq b$ for each $g \in \mathcal{G}$, then for every $n \geq 1$ and for $c > b^2 - \|f\|^2$ there is a f_n in the convex hull of n points in $\mathcal{G}$ such that*

$$\|f - f_n\|^2 \leq \frac{c}{n} .$$

In order to exploit this result one needs to define suitable classes of functions which are the closure of the convex hull of some subset $\mathcal{G}$ of a Hilbert space H. One way to approach the problem consists in utilizing the *integral representation* of functions. Suppose that the functions in a Hilbert space H can be represented by the integral

$$f(\mathbf{x}) = \int_{\mathcal{M}} G_m(\mathbf{x}; \mathbf{t}) d\alpha(\mathbf{t}) \tag{2.22}$$

where α is some measure on the parameter set $\mathcal{M}$, and $G_m(\mathbf{x}; \mathbf{t})$ is a function of H parametrized by the parameter $\mathbf{t}$, whose norm $\|G_m(\mathbf{x}; \mathbf{t})\|$ is bounded by the same number for any value of $\mathbf{t}$. In particular, if we let $G_m(\mathbf{x}; \mathbf{t})$ be translates of G_m by $\mathbf{t}$, i.e., $G_m(\mathbf{x} - \mathbf{t})$, and α be a finite measure, the integral (2.22) can be seen as an infinite convex combination of translates of G_m.

We now make the following two observations. First, it is clear that elements of $\mathcal{F}$ have an integral representation of the type (2.22) and are members of the Hilbert space H. Second, since λ is a finite measure (bounded by M) elements of $\mathcal{F}$ are infinite convex combinations of translates of G_m. We now make use of the important fact that convex combinations of translates of G_m can be represented as convex combinations of translates and dilates of Gaussians (in other words sets of the form of H_n for some n).

This allows us to define $\mathcal{G}$ of lemma 2-D.1 to be the parametrized set $\mathcal{G} = \{g | g(\mathbf{x}) = G(\frac{\|\mathbf{x}-\mathbf{t}\|}{\sigma})\}$. Clearly, elements of $\mathcal{F}$ lie in the convex hull of $\mathcal{G}$ as defined above and therefore, applying lemma (2-D.1) one can prove ([50, 51]) that there exist n coefficients c_i, n parameter vectors $\mathbf{t}_i$, and n choices for σ_i such that

$$\|f - \sum_{i=1}^{n} c_i G(\mathbf{x}; \mathbf{t_i}; \sigma_i)\|^2 \leq O(\frac{1}{n})$$

Notice that the bound (2.21), that is similar in spirit to the result of A. Barron on multilayer perceptrons [8, 10], is interesting because the rate of convergence does not depend on the dimension d of the input space. This is apparently unusual in approximation theory, because it is known, from the theory of linear and nonlinear widths [121, 104, 87, 88, 32, 31, 33, 91], that, if the function that has to be approximated has d variables and a degree of smoothness s, we should not expect to find an approximation technique whose approximation error goes to zero faster than $O(n^{-\frac{s}{d}})$. Here "degree of smoothness" is a measure of how constrained the class of functions we consider is, for example the number of derivatives that are uniformly bounded, or the number of derivatives that are integrable or square integrable. Therefore, from classical approximation theory, we expect that, *unless certain constraints are imposed on the class of functions to be approximated*, the rate of convergence will dramatically slow down as the number of dimensions increases, showing the phenomenon known as "the curse of dimensionality" [13].

In the case of class $\mathcal{F}$ we consider here, the constraint of considering functions that are convolutions of Radon measures with Gaussians seems to impose on this class of functions an amount of smoothness that is sufficient to guarantee that the rate of convergence does not become slower and slower as the dimension increases. A longer discussion of the "curse of dimensionality" can be found in [51].

We notice also that, since the rate (2.21) is independent of the dimension, the class $\mathcal{F}$, together with the approximating class H_n, defines a class of problems that are "tractable" even in a high number of dimensions.

2-D.2 Bounding the estimation error

In this part we attempt to bound the estimation error $|I[f] - I_{\text{emp}}[f]|$. In order to do that we first need to introduce some basic concepts and notations.

Let $\mathcal{S}$ be a subset of a metric space S with metric d. We say that an **ϵ-cover** with respect to the metric d is a set $\mathcal{T} \in S$ such that for every $s \in \mathcal{S}$, there exists some $t \in \mathcal{T}$ satisfying $d(s,t) \leq \epsilon$. The size of the smallest ϵ-cover is $\mathcal{N}(\epsilon, \mathcal{S}, d)$ and is called the **covering number** of $\mathcal{S}$. In other words

$$\mathcal{N}(\epsilon, \mathcal{S}, d) = \min_{\mathcal{T} \subset \mathcal{S}} |\mathcal{T}| \ ,$$

where $\mathcal{T}$ runs over all the possible ϵ-cover of $\mathcal{S}$ and $|\mathcal{T}|$ denotes the cardinality of $\mathcal{T}$.

A set B belonging to the metric space S is said to be **ϵ-separated** if for all $x, y \in B$, $d(x,y) > \epsilon$. We define the the *packing number* $\mathcal{M}(\epsilon, \mathcal{S}, d)$ as the size of the largest ϵ-separated subset of $\mathcal{S}$. Thus

$$\mathcal{M}(\epsilon, \mathcal{S}, d) = \max_{B \subset \mathcal{S}} |B| \ ,$$

where B runs over all the ϵ-separated subsets of $\mathcal{S}$. It is easy to show that the covering number is always less than the packing number, that is $\mathcal{N}(\epsilon, \mathcal{S}, d) \leq \mathcal{M}(\epsilon, \mathcal{S}, d)$.

Let now $P(\xi)$ be a probability distribution defined on S, and $\mathcal{A}$ be a set of real-valued functions defined on S such that, for any $a \in \mathcal{A}$,

$$0 \leq a(\xi) \leq U^2 \quad \forall \xi \in S \ .$$

Let also $\bar{\xi} = (\xi_1, .., \xi_l)$ be a sequence of l examples drawn independently from S according to $P(\xi)$. For any function $a \in \mathcal{A}$ we define the empirical and true expectations of a as follows:

$$\hat{E}[a] = \frac{1}{l} \sum_{i=1}^{l} a(\xi_i)$$

$$E[a] = \int_S d\xi P(\xi) a(\xi)$$

The difference between the empirical and true expectation can be bounded by the following inequality, whose proof can be found in [109] and [63], that will be crucial in order to prove our main theorem.

Claim 2-D.1 ([109], [63]) *Let $\mathcal{A}$ and $\bar{\xi}$ be as defined above. Then, for all $\epsilon > 0$,*

$$P\left(\exists a \in \mathcal{A} : |\hat{E}[a] - E[a]| > \epsilon\right) \leq$$

$$\leq 4E\left[\mathcal{N}(\tfrac{\epsilon}{16}, \mathcal{A}_{\bar{\xi}}, d_{L^1})\right] e^{-\frac{1}{128U^4}\epsilon^2 l}$$

In the above result, $\mathcal{A}_{\bar{\xi}}$ is the restriction of $\mathcal{A}$ to the data set, that is:

$$\mathcal{A}_{\bar{\xi}} \equiv \{(a(\xi_1), \ldots, a(\xi_l)) : a \in \mathcal{A}\} \ . \tag{2.23}$$

The set $\mathcal{A}_{\bar{\xi}}$ is a collection of points belonging to the subset $[0, U]^l$ of the l-dimensional euclidean space. Each function a in $\mathcal{A}$ is represented by a point in $\mathcal{A}_{\bar{\xi}}$, while every point in $\mathcal{A}_{\bar{\xi}}$ represents all the functions that have the same values at the points $\xi_1, \ldots, \xi_l$. The distance metric d_{L^1} in the inequality above is the standard L^1 metric in R^l, that is

$$d_{L^1}(\mathbf{x}, \mathbf{y}) = \frac{1}{l} \sum_{\mu=1}^{l} |x^\mu - y^\mu|$$

where $\mathbf{x}$ and $\mathbf{y}$ are points in the l-dimensional euclidean space and x^μ and y^μ are their μ-th components respectively.

The above inequality is a result in the theory of uniform convergence of empirical measures to their underlying probabilities, that has been studied in great detail by Pollard and Vapnik, and similar inequalities can be found in the work of Vapnik [125, 126, 124], although they usually involve the VC dimension of the set $\mathcal{A}$, rather than its covering numbers.

Suppose now we choose $S = X \times Y$, where X is an arbitrary subset of R^k and $Y = [-M, M]$ as in the formulation of our original problem. The generic element of S will be written as $\xi = (\mathbf{x}, y) \in X \times Y$. We now consider the class of functions $\mathcal{A}$ defined as:

$$\mathcal{A} = \{a : X \times Y \rightarrow R \mid a(\mathbf{x}, y) = (y - h(\mathbf{x}))^2, \ h \in H_n(R^k)\}$$

where $H_n(R^k)$ is the class of k-dimensional Radial Basis Functions with n basis functions defined in eq. 2.13 in section 2.3. Clearly,

$$|y - h(\mathbf{x})| \leq |y| + |h(\mathbf{x})| \leq M + MV,$$

and therefore

$$0 \leq a \leq U^2$$

where we have defined

$$U \equiv M + MV \ .$$

We notice that, by definition of $\hat{E}(a)$ and $E(a)$ we have

$$\hat{E}(a) = \frac{1}{l} \sum_{i=1}^{l} (y_i - h(\mathbf{x}_i))^2 = I_{\text{emp}}[h]$$

and

$$E(a) = \int_{X \times Y} d\mathbf{x} dy \ P(\mathbf{x}, y)(y - h(\mathbf{x}))^2 = I[h] \ .$$

Therefore, applying the inequality of claim 2-D.1 to the set $\mathcal{A}$, and noticing that the elements of $\mathcal{A}$ are essentially defined by the elements of H_n, we obtain the following result:

$$
\begin{aligned}
&P(\forall h \in H_n, |I_{\text{emp}}[h] - I[h]| \leq \epsilon) \geq \\
&\geq 1 - 4E[\mathcal{N}(\epsilon/16, \mathcal{A}_{\bar{\xi}}, d_{L^1})] e^{-\frac{1}{128U^4}\epsilon^2 l} \ .
\end{aligned}
\tag{2.24}
$$

so that the inequality of claim 2-D.1 gives us a bound on the estimation error. However, this bound depends on the specific choice of the probability distribution $P(\mathbf{x}, y)$, while we are interested in bounds that do not depend on P. Therefore it is useful to define some quantity that does not depend on P, and give bounds in terms of that.

We then introduce the concept of **metric capacity** of $\mathcal{A}$, that is defined as

$$
\mathcal{C}(\epsilon, \mathcal{A}, d_{L^1}) = \sup_P \{\mathcal{N}(\epsilon, \mathcal{A}, d_{L^1(P)})\}
$$

where the supremum is taken over all the probability distributions P defined over S, and $d_{L^1(P)}$ is standard $L^1(P)$ distance[12]

induced by the probability distribution P:

$$
d_{L^1(P)}(a_1, a_2) = \int_S d\xi P(\xi) |a_1(\xi) - a_2(\xi)| \quad a_1, a_2 \in \mathcal{A} \ .
$$

The relationship between the covering number and the metric capacity is showed in the following

Claim 2-D.2

$$
E[\mathcal{N}(\epsilon, \mathcal{A}_{\bar{\xi}}, d_{L^1})] \leq \mathcal{C}(\epsilon, \mathcal{A}, d_{L^1}) \ .
$$

Proof: For any sequence of points $\bar{\xi}$ in S, there is a trivial isometry between $(\mathcal{A}_{\bar{\xi}}, d_{L^1})$ and $(\mathcal{A}, d_{L^1(P_{\bar{\xi}})})$ where $P_{\bar{\xi}}$ is the empirical distribution on the space S given by $\frac{1}{l}\sum_{i=1}^{l} \delta(\xi - \xi_i)$. Here δ is the Dirac delta function, $\xi \in S$, and ξ_i is the i-th element of the data set. To see that this isometry exists, first note that for every element $a \in \mathcal{A}$, there exists a unique point $(a(\xi_1), \ldots, a(\xi_l)) \in \mathcal{A}_{\bar{\xi}}$. Thus a simple bijective mapping exists between the two spaces. Now consider any two elements g and h of $\mathcal{A}$. The distance between them is given by

[12] Note that here $\mathcal{A}$ is a class of real-valued functions defined on a general metric space S. If we consider an arbitrary $\mathcal{A}$ defined on S and taking values in R^n, the $d_{L^1(P)}$, norm is appropriately adjusted to be

$$
d_{L^1(P)}(\mathbf{f}, \mathbf{g}) = \frac{1}{n} \sum_{i=1}^{n} \int_S |f_i(\mathbf{x}) - g_i(\mathbf{x})| P(\mathbf{x}) d\mathbf{x}
$$

where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots f_i(\mathbf{x}), \ldots f_n(\mathbf{x}))$, $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots g_i(\mathbf{x}), \ldots g_n(\mathbf{x}))$ are elements of $\mathcal{A}$ and $P(\mathbf{x})$ is a probability distribution on S. Thus d_{L^1} and $d_{L^1(P)}$ should be interpreted according to the context.

$$d_{L^1(P_{\bar{\xi}})}(g,h) = \int_S |g(\xi) - h(\xi)| P_{\bar{\xi}}(\xi) d\xi = \frac{1}{l}\sum_{i=1}^{l} |g(\xi_i) - h(\xi_i)|.$$

This is exactly what the distance between the two points $(g(\xi_1),..,g(\xi_l))$ and $(h(\xi_1),...h(\xi_l))$, which are elements of $\mathcal{A}_{\bar{\xi}}$, is according to the d_{L^1} distance. Thus there is a one-to-one correspondence between elements of $\mathcal{A}$ and $\mathcal{A}_{\bar{\xi}}$ and the distance between two elements in $\mathcal{A}$ is the same as the distance between their corresponding points in $\mathcal{A}_{\bar{\xi}}$. Given this isometry, for every ϵ-cover in $\mathcal{A}$, there exists an ϵ-cover of the same size in $\mathcal{A}_{\bar{\xi}}$, so that

$$\mathcal{N}(\epsilon, \mathcal{A}_{\bar{\xi}}, d_{L^1}) = \mathcal{N}(\epsilon, \mathcal{A}, d_{L^1(P_{\bar{\xi}})}) \leq \mathcal{C}(\epsilon, \mathcal{A}, d_{L^1}).$$

and consequently $E[\mathcal{N}(\epsilon, \mathcal{A}_{\bar{\xi}}, d_{L^1})] \leq \mathcal{C}(\epsilon, \mathcal{A}, d_{L^1})$. □

The result above, together with eq. (2.24) shows that the following proposition holds:

Claim 2-D.3

$$\begin{gathered} P(\forall h \in H_n, |I_{\text{emp}}[h] - I[h]| \leq \epsilon) \geq \\ \geq 1 - 4\mathcal{C}(\epsilon/16, \mathcal{A}, d_{L^1})]e^{-\frac{1}{128U^4}\epsilon^2 l} \, . \end{gathered} \tag{2.25}$$

Thus in order to obtain a uniform bound ω on $|I_{\text{emp}}[h] - I[h]|$, our task is reduced to computing the metric capacity of the functional class $\mathcal{A}$ which we have just defined. We will do this in several steps. In Claim 2-D.4, we first relate the metric capacity of $\mathcal{A}$ to that of the class of radial basis functions H_n. Then Claims 2-D.5 through 2-D.9 go through a computation of the metric capacity of H_n.

Claim 2-D.4

$$C(\epsilon, \mathcal{A}, d_{L^1}) \leq C(\epsilon/4U, H_n, d_{L^1})$$

Proof: Fix a distribution P on $S = X \times Y$. Let P_X be the marginal distribution with respect to X. Suppose K is an $\epsilon/4U$-cover for H_n with respect to this probability distribution P_X, i.e. with respect to the distance metric $d_{L^1(P_X)}$ on H_n. Further let the size of K be $\mathcal{N}(\epsilon/4U, H_n, d_{L^1(P_X)})$. This means that for any $h \in H_n$, there exists a function h^* belonging to K, such that:

$$\int |h(\mathbf{x}) - h^*(\mathbf{x})| P_X(\mathbf{x}) d\mathbf{x} \leq \epsilon/4U$$

Now we claim the set $H(K) = \{(y - h(\mathbf{x}))^2 : h \in K\}$ is an ϵ cover for $\mathcal{A}$ with respect to the distance metric $d_{L^1(P)}$. To see this, it is sufficient to show that

$$\int |(y-h(\mathbf{x}))^2-(y-h^*(\mathbf{x}))^2|P(\mathbf{x},y)d\mathbf{x}dy \leq$$

$$\leq \int 2|(2y-h-h^*)||(h-h^*)|P(\mathbf{x},y)d\mathbf{x}dy \leq$$

$$\leq \int 2(2M+2MV)|h-h^*|P(\mathbf{x},y)d\mathbf{x}dy \leq \epsilon$$

which is clearly true. Now

$$\mathcal{N}(\epsilon,\mathcal{A},d_{L^1(P)}) \leq |H(K)| =$$

$$= calN(\epsilon/4U,H_n,d_{L^1(P_X)}) \leq$$

$$\leq \mathcal{C}(\epsilon/4U,H_n,d_{L^1})$$

Taking the supremum over all probability distributions, the result follows. □

So the problem reduces to finding $C(\epsilon,H_n,d_{L^1})$, i.e. the metric capacity of the class of appropriately defined Radial Basis Functions networks with n centers. To do this we will decompose the class H_n to be the composition of two classes defined as follows.

Definitions/Notations

H_I is a class of functions defined from the metric space (R^k,d_{L^1}) to the metric space (R^n,d_{L^1}). In particular,

$$H_I = \{\mathbf{g}(\mathbf{x}) = (G(\|\mathbf{x}-\mathbf{t}_1\|),G(\|\mathbf{x}-\mathbf{t}_2\|),\ldots,G(\|\mathbf{x}-\mathbf{t}_n\|))\}$$

where G is a Gaussian and $\mathbf{t}_i$ are k-dimensional vectors.
Note here that G is the same Gaussian that we have been using to build our Radial-Basis-Function Network. Thus H_I is parametrized by the n centers t_i and the variance of the Gaussian σ^2, in other words $nk+1$ parameters in all.

H_F is a class defined from the metric space $([0,V]^n,d_{L^1})$ to the metric space (R,d_{L^1}). In particular,

$$H_F = \{h(\mathbf{x}) = \boldsymbol{\beta}\cdot\mathbf{x},\ \ \mathbf{x}\in[0,V]^n \text{ and } \sum_{i=1}^{n}|\beta_i| \leq M\}$$

where $\boldsymbol{\beta} \equiv (\beta_1,\ldots,\beta_n)$ is an arbitrary n-dimensional vector.

Thus we see that

$$H_n = \{h_F \circ h_I : h_F \in H_F \text{ and } h_I \in H_I\}$$

where $\circ$ stands for the composition operation, i.e., for any two functions f and g, $f \circ g = f(g(\mathbf{x}))$. It should be pointed out that H_n as defined above is defined from R^k to R.

Claim 2-D.5

$$C(\epsilon, H_I, d_{L^1}) \leq 2^n \left(\frac{2eV}{\epsilon} \ln\left(\frac{2eV}{\epsilon}\right)\right)^{n(k+2)}$$

Proof: Fix a probability distribution P on R^k. Consider the class

$$\mathcal{G} = \{g : g(\mathbf{x}) = G(\|\mathbf{x} - \mathbf{t}\|),\ \mathbf{t} \in R^k\}.$$

Let K be an $\mathcal{N}(\epsilon, \mathcal{G}, d_{L^1(P)})$-sized ϵ cover for this class. We first claim that

$$T = \{(h_1, .., h_n) : h_i \in K\}$$

is an ϵ-cover for H_I with respect to the $d_{L^1(P)}$ metric.

Remember that the $d_{L^1(P)}$ distance between two vector-valued functions $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), .., g_n(\mathbf{x}))$ and $\mathbf{g}^*(\mathbf{x}) = (g_1^*(\mathbf{x}), .., g_n^*(\mathbf{x}))$ is defined as

$$d_{L^1(P)}(\mathbf{g}, \mathbf{g}^*) = \frac{1}{n}\sum_{i=1}^{n} \int |g_i(\mathbf{x}) - g_i^*(\mathbf{x})| P(\mathbf{x}) d\mathbf{x}$$

To see this, pick an arbitrary $\mathbf{g} = (g_1, \ldots, g_n) \in H_I$. For each g_i, there exists a $g_i^* \in K$ which is ϵ-close in the appropriate sense for real-valued functions, i.e. $d_{L^1(P)}(g_i, g_i^*) \leq \epsilon$. The function $\mathbf{g} = (g_1^*, .., g_n^*)$ is an element of T. Also, the distance between $(g_1, .., g_n)$ and $(g_1^*, .., g_n^*)$ in the $d_{L^1(P)}$ metric is

$$d_{L^1(P)}(\mathbf{g}, \mathbf{g}^*) \leq \frac{1}{n}\sum_{i=1}^{n} \epsilon = \epsilon\ .$$

Thus we obtain that

$$\mathcal{N}(\epsilon, H_I, d_{L^1(P)}) \leq [\mathcal{N}(\epsilon, \mathcal{G}, d_{L^1(P)})]^n$$

and taking the supremum over all probability distributions as usual, we get

$$\mathcal{C}(\epsilon, H_I, d_{L^1}) \leq (\mathcal{C}(\epsilon, \mathcal{G}, d_{L^1}))^n\ .$$

Now we need to find the capacity of $\mathcal{G}$. This is done in the Claim 2-D.6. From this the result follows. □

Definitions/Notations

Before we proceed to the next step in our proof, some more notation needs to be defined. Let $\mathcal{A}$ be a family of functions from a set S into R. For any sequence $\bar{\xi} = (\xi_1, .., \xi_d)$ of points in S, let $\mathcal{A}_{\bar{\xi}}$ be the restriction of $\mathcal{F}$ to the data set, as per our previously introduced notation. Thus $\mathcal{A}_{\bar{\xi}} = \{(a(\xi_1), \ldots, a(\xi_d)) : a \in \mathcal{A}\}$. If there exists some translation of the set $\mathcal{A}_{\bar{\xi}}$, such that it intersects all 2^d orthants of the space R^d, then $\bar{\xi}$ is said to be **shattered** by $\mathcal{A}$. Expressing this a little more formally, let $\mathcal{B}$ be the set of all possible l-dimensional boolean vectors. If there exists a translation $\mathbf{t} \in R^d$ such that for every $\mathbf{b} \in \mathcal{B}$, there exists some function $a_{\mathbf{b}} \in \mathcal{A}$ satisfying $a_{\mathbf{b}}(\xi_i) - t_i \geq b_i \Leftrightarrow b_i = 1$ for all $i = 1$ to d, then the set $(\xi_1, .., \xi_d)$ is shattered by $\mathcal{A}$. Note that the inequality could easily have been defined to be strict and would not have made a difference. The largest d such that there exists a sequence of d points which are shattered by $\mathcal{A}$ is said to be the pseudo-dimension of $\mathcal{A}$ denoted by pdim$\mathcal{A}$. □

In this context, there are two important theorems which we will need to use. We give these theorems without proof.

Theorem 2-D.1 (Dudley) *Let F be a k-dimensional vector space of functions from a set S into R. Then pdim$(F) = k$.*

The following theorem is stated and proved in a somewhat more general form by Pollard. Haussler, using techniques from Pollard has proved the specific form shown here.

Theorem 2-D.2 (Pollard, Haussler) *Let F be a family of functions from a set S into $[M_1, M_2]$, where pdim$(F) = d$ for some $1 \leq d < \infty$. Let P be a probability distribution on S. Then for all $0 < \epsilon \leq M_2 - M_1$,*

$$\mathcal{M}(\epsilon, F, d_{L^1(P)}) < 2\left(\frac{1}{\epsilon}2e(M_2 - M_1)\log\frac{1}{\epsilon}2e(M_2 - M_1)\right)^d$$

Here $\mathcal{M}(\epsilon, F, d_{L^1(P)})$ is the packing number of F according to the distance metric $d_{L^1(P)}$.

Claim 2-D.6

$$C(\epsilon, \mathcal{G}, d_{L^1}) \leq 2\left(\frac{2eV}{\epsilon}\ln\left(\frac{2eV}{\epsilon}\right)\right)^{(k+2)}$$

Proof: Consider the $k+2$-dimensional vector space of functions from R^k to R defined as

$$G_1 \equiv \text{span}\{1, x^1, x^2, \cdot, x^k, \|\mathbf{x}\|^2\}$$

where $\mathbf{x} \in R^k$ and x^μ is the μ-th component of the vector $\mathbf{x}$. Now consider the class

$$G_2 = \{\alpha e^{-f} : f \in G_1, \quad \alpha = \frac{1}{\sqrt{2\pi}\sigma}\}$$

where σ is the standard deviation of the Gaussian, and $f \in G_1$. We claim that the pseudo-dimension of $\mathcal{G}$ denoted by pdim$(\mathcal{G})$ fulfills the following inequality,

$$\text{pdim } (\mathcal{G}) \leq \text{ pdim } (G_2) = \text{ pdim } (G_1) = (k+2).$$

To see this consider the fact that $\mathcal{G} \subset G_2$. Consequently, for every sequence of points $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$, $\mathcal{G}_{\bar{\mathbf{x}}} \subset (G_2)_{\bar{\mathbf{x}}}$. Thus if $(\mathbf{x}_1, \ldots, \mathbf{x}_d)$ is shattered by $\mathcal{G}$, it will be shattered by G_2. This establishes the first inequality.

We now show that pdim$(G_2) \leq$ pdim(G_1). It is enough to show that every set shattered by G_2 is also shattered by G_1. Suppose there exists a sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$ which is shattered by G_2. This means that by our definition of shattering, there exists a translation $\mathbf{t} \in R^d$ such that for every boolean vector $\mathbf{b} \in \{0,1\}^d$ there is some function $g_{\mathbf{b}} = \alpha e^{-f_{\mathbf{b}}}$ where $f_{\mathbf{b}} \in G_1$ satisfying $g_{\mathbf{b}}(x_i) \geq t_i$ if and only if $b_i = 1$, where t_i and b_i are the i-th components of $\mathbf{t}$ and $\mathbf{b}$ respectively. First notice that every function in G_2 is positive. Consequently, we see that every t_i has to be greater than 0, for otherwise, $g_b(\mathbf{x}_i)$ could never be less than t_i which it is required to be if $b_i = 0$. Having established that every t_i is greater than 0, we now show that the set $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$ is shattered by G_1. We let the translation in this case be $\mathbf{t}' = (\log(t_1/\alpha), \log(t_2/\alpha), \ldots, \log(t_d/\alpha))$. We can take the log since the t_i/α's are greater than 0. Now for every boolean vector $\mathbf{b}$, we take the function $-f_b \in G_1$ and we see that since

$$g_b = \alpha e^{-f_b} \geq t_i \Leftrightarrow b_i = 1.$$

if follows that

$$-f_b \geq \log(t_i/\alpha) = \mathbf{t}'_i \Leftrightarrow b_i = 1.$$

Thus we see that the set $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$ can be shattered by G_1. By a similar argument, it is also possible to show that pdim$(G_1) \geq$ pdim(G_2).

Since G_1 is a vector space of dimensionality $k+2$, an application of Dudley's Theorem

[37] yields the value $k+2$ for its pseudo-dimension. Further, functions in the class $\mathcal{G}$ are in the range $[0, V]$. Now we see (by an application of Pollard's theorem) that

$$\mathcal{N}(\epsilon, \mathcal{G}, d_{L^1(P)}) \leq \mathcal{M}(\epsilon, \mathcal{G}, d_{L^1(P)}) \leq$$

$$\leq 2\left(\tfrac{2eV}{\epsilon}\ln\left(\tfrac{2eV}{\epsilon}\right)\right)^{\text{pdim}(\mathcal{G})} \leq$$

$$\leq 2\left(\tfrac{2eV}{\epsilon}\ln\left(\tfrac{2eV}{\epsilon}\right)\right)^{(k+2)}$$

Taking the supremum over all probability distributions, the result follows.□

Claim 2-D.7

$$C(\epsilon, H_F, d_{L^1}) \leq 2\left(\frac{4MeV}{\epsilon}\ln\left(\frac{4MeV}{\epsilon}\right)\right)^n$$

Proof: The proof of this runs in very similar fashion. First note that

$$H_F \subset \{\boldsymbol{\beta}\cdot\mathbf{x} : \mathbf{x},\ \boldsymbol{\beta} \in R^n\}.$$

The latter set is a vector space of dimensionality n and by Dudley's theorem[37], we see that its pseudo-dimension pdim is n. Also, clearly by the same argument as in the previous proposition, we have that $\text{pdim}(H_F) \leq n$. To get bounds on the functions in H_F, notice that

$$|\sum_{i=1}^{n}\beta_i x_i| \leq \sum_{i=1}^{n}|\beta_i||x_i| \leq V\sum_{i=1}^{n}|\beta_i| \leq MV.$$

Thus functions in H_F are bounded in the range $[-MV, MV]$. Now using Pollard's result [63], [109], we have that

$$\mathcal{N}(\epsilon, H_F, d_{L^1(P)}) \leq \mathcal{M}(\epsilon, H_F, d_{L^1(P)}) \leq$$

$$\leq 2\left(\tfrac{4MeV}{\epsilon}\ln\left(\tfrac{4MeV}{\epsilon}\right)\right)^n.$$

Taking supremums over all probability distributions, the result follows. □

Claim 2-D.8 *A uniform first-order Lipschitz bound of H_F is Mn.*

Proof: Suppose we have $\mathbf{x},\ \mathbf{y} \in R^n$ such that

$$d_{L^1}(\mathbf{x}, \mathbf{y}) \leq \epsilon.$$

The quantity Mn is a uniform first-order Lipschitz bound for H_F if, for any element of H_F, parametrized by a vector $\boldsymbol{\beta}$, the following inequality holds:

$$|\mathbf{x} \cdot \boldsymbol{\beta} - \mathbf{y} \cdot \boldsymbol{\beta}| \leq Mn\epsilon$$

Now clearly,

$$|\mathbf{x} \cdot \boldsymbol{\beta} - \mathbf{y} \cdot \boldsymbol{\beta}| = |\sum_{i=1}^{n} \beta_i(x_i - y_i)| \leq$$

$$\leq \sum_{i=1}^{n} |\beta_i||(x_i - y_i)| \leq$$

$$\leq M \sum_{i=1}^{n} |(x_i - y_i)| \leq Mn\epsilon$$

The result is proved. □

Claim 2-D.9

$$C(\epsilon, H_n, d_{L^1}) \leq C(\frac{\epsilon}{2Mn}, H_I, d_{L^1})C(\frac{\epsilon}{2}, H_F, d_{L^1})$$

Proof: Fix a distribution P on R^k. Assume we have an $\epsilon/(2Mn)$-cover for H_I with respect to the probability distribution P and metric $d_{L^1(P)}$. Let it be K where

$$|K| = \mathcal{N}(\epsilon/2Mn, H_I, d_{L^1(P)}).$$

Now each function $f \in K$ maps the space R^k into R^n, thus inducing a probability distribution P_f on the space R^n. Specifically, P_f can be defined as the distribution obtained from the measure μ_f defined so that any measurable set $A \subset R^n$ will have measure

$$\mu_f(A) = \int_{f^{-1}(A)} P(\mathbf{x})d\mathbf{x} .$$

Further, there exists a cover K_f which is an $\epsilon/2$-cover for H_F with respect to the probability distribution P_f. In other words

$$|K_f| = \mathcal{N}(\epsilon/2, H_F, d_{L^1(P_f)}).$$

We claim that

$$H(K) = \{f \circ g : g \in K \text{ and } f \in K_g\}$$

is an ϵ cover for H_n. Further we note that

$$|H(K)| = \textstyle\sum_{f\in K} |K_f| \leq \sum_{f\in K} \mathcal{C}(\epsilon/2, H_F, d_{L^1}) \leq$$

$$\leq \mathcal{N}(\epsilon/(2Mn), H_I, d_{L^1(P)})\mathcal{C}(\epsilon/2, H_F, d_{L^1})$$

To see that $H(K)$ is an ϵ-cover, suppose we are given an arbitrary function $h_f \circ h_i \in H_n$. There clearly exists a function $h_i^* \in K$ such that

$$\int_{R^k} d_{L^1}(h_i(\mathbf{x}), h_i^*(\mathbf{x}))P(\mathbf{x})d\mathbf{x} \leq \epsilon/(2Mn)$$

Now there also exists a function $h_f^* \in K_{h_i^*}$ such that

$$\textstyle\int_{R^k} |h_f \circ h_i^*(\mathbf{x}) - h_f^* \circ h_i^*(\mathbf{x})|P(\mathbf{x})d\mathbf{x} =$$

$$= \textstyle\int_{R^n} |h_f(\mathbf{y}) - h_f^*(\mathbf{y})|P_{h_i^*}(\mathbf{y})d\mathbf{y} \leq \epsilon/2\ .$$

To show that $H(K)$ is an ϵ-cover it is sufficient to show that

$$\int_{R^k} |h_f \circ h_i(\mathbf{x}) - h_f^* \circ h_i^*(\mathbf{x})|P(\mathbf{x})d\mathbf{x} \leq \epsilon.$$

Now

$$\textstyle\int_{R^k} |h_f \circ h_i(\mathbf{x}) - h_f^* \circ h_i^*(\mathbf{x})|P(\mathbf{x})d\mathbf{x} \leq$$

$$\leq \textstyle\int_{R^k}\{|h_f \circ h_i(\mathbf{x}) - h_f \circ h_i^*(\mathbf{x})|+$$

$$+|h_f \circ h_i^*(\mathbf{x}) - h_f^* \circ h_i^*(\mathbf{x})|P(\mathbf{x})d\mathbf{x}\}$$

by the triangle inequality. Further, since h_f is Lipschitz bounded,

$$\textstyle\int_{R^k} |h_f \circ h_i(\mathbf{x}) - h_f \circ h_i^*(\mathbf{x})|P(\mathbf{x})d\mathbf{x} \leq$$

$$\leq \textstyle\int_{R^k} Mnd_{L^1}(h_i(\mathbf{x}), h_i^*(\mathbf{x}))P(\mathbf{x})d\mathbf{x} \leq Mn(\epsilon/2Mn) \leq \epsilon/2\ .$$

Also,

$$\textstyle\int_{R^k} |h_f \circ h_i^*(\mathbf{x}) - h_f^* \circ h_i^*(\mathbf{x})|P(\mathbf{x})d\mathbf{x} =$$

$$= \textstyle\int_{R^n} |h_f(\mathbf{y}) - h_f^*(\mathbf{y})|P_{h_i^*}(\mathbf{y})d\mathbf{y} \leq \epsilon/2\ .$$

Consequently both sums are less than $\epsilon/2$ and the total integral is less than ϵ. Now we see that

$$\mathcal{N}(\epsilon, H_n, d_{L^1(P)}) \leq \mathcal{N}\left(\epsilon/(2Mn), H_I, d_{L^1(P)}\right) \mathcal{C}(\epsilon/2, H_F, d_{L^1}).$$

Taking supremums over all probability distributions, the result follows. □

Having obtained the crucial bound on the metric capacity of the class H_n, we can now prove the following

Claim 2-D.10 *With probability* $1-\delta$, *and* $\forall h \in H_n$, *the following bound holds:*

$$|I_{\mathrm{emp}}[h] - I[h]| \leq O\left(\left[\frac{nk\ln(nl) + \ln(1/\delta)}{l}\right]^{1/2}\right)$$

Proof: We know from the previous claim that

$$C(\epsilon, H_n, d_{L^1}) \leq$$

$$\leq 2^{n+1}\left[\tfrac{4MeVn}{\epsilon}\ln\left(\tfrac{4MeVn}{\epsilon}\right)\right]^{n(k+2)}\left[\tfrac{8MeV}{\epsilon}\ln\left(\tfrac{8MeV}{\epsilon}\right)\right]^{n} \leq$$

$$\leq \left[\tfrac{8MeVn}{\epsilon}\ln(\tfrac{8MeVn}{\epsilon})\right]^{n(k+3)}.$$

From claim (2-D.3), we see that

$$P(\forall h \in H_n, |I_{\mathrm{emp}}[h] - I[h]| \leq \epsilon) \geq \tag{2.26}$$

$$\geq 1-\delta$$

as long as

$$\mathcal{C}(\epsilon/16, \mathcal{A}, d_{L^1})e^{-\frac{1}{128U^4}\epsilon^2 l} \leq \frac{\delta}{4}$$

which in turn is satisfied as long as (by Claim 2-D.4)

$$\mathcal{C}(\epsilon/64U, H_n, d_{L^1})e^{-\frac{1}{128U^2}\epsilon^2 l} \leq \frac{\delta}{4}$$

which implies

$$\left(\tfrac{1}{\epsilon}256MeVUn\ \ln\left(\tfrac{1}{\epsilon}256MeVUn\right)\right)^{n(k+3)}\cdot$$

$$\cdot e^{-\frac{1}{128U^2}\epsilon^2 l} \leq \tfrac{\delta}{4}$$

In other words,

$$\left(\frac{An}{\epsilon}\ln\left(\frac{An}{\epsilon}\right)\right)^{n(k+3)} e^{-\epsilon^2 l/B} \leq \frac{\delta}{4}$$

for constants A, B. The latter inequality is satisfied as long as

$$(An/\epsilon)^{2n(k+3)} e^{-\epsilon^2 l/B} \leq \frac{\delta}{4}$$

which implies

$$2n(k+3)(\ln(An) - \ln(\epsilon)) - \epsilon^2 l/B \leq \ln(\delta/4)$$

and in turn implies

$$\epsilon^2 l > B\ln(4/\delta) + 2Bn(k+3)(\ln(An) - \ln(\epsilon)).$$

We now show that the above inequality is satisfied for

$$\epsilon = \left(\frac{B\left[\ln(4/\delta) + 2n(k+3)\ln(An) + n(k+3)\ln(l)\right]}{l}\right)^{1/2}$$

Putting the above value of ϵ in the inequality of interest, we get

$$\epsilon^2(l/B) = \ln(4/\delta) + 2n(k+3)\ln(An) + n(k+3)\ln(l) \geq$$

$$\geq \ln(4/\delta) + 2n(k+3)\ln(An) +$$

$$+ 2n(k+3)\tfrac{1}{2}\ln\left(\tfrac{l}{B[\ln(4/\delta)+2n(k+3)\ln(An)+n(k+3)\ln(l)]}\right)$$

In other words,

$$n(k+3)\ln(l) \geq$$

$$\geq n(k+3)\ln\left(\tfrac{l}{B[\ln(4/\delta)+2n(k+3)\ln(An)+n(k+3)\ln(l)]}\right)$$

Since

$$B\left[\ln(4/\delta) + 2n(k+3)\ln(An) + n(k+3)\ln(l)\right] \geq 1$$

the inequality is obviously true for this value of ϵ. Taking this value of ϵ then proves our claim. □

2-D.3 Bounding the generalization error

Finally we are able to take our results in Parts II and III to prove our main result:

Theorem 2-D.3 *With probability greater than $1-\delta$ the following inequality is valid:*

$$\|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)} \leq O\left(\frac{1}{n}\right) + O\left(\left[\frac{nk\ln(nl) - \ln\delta}{l}\right]^{1/2}\right)$$

Proof: We have seen in statement (2.2.1) that the generalization error is bounded as follows:

$$\|f_0 - \hat{f}_{n,l}\|^2_{L^2(P)} \leq \varepsilon(n) + 2\omega(l,n,\delta) \;.$$

In section (2-D.1) we showed that

$$\varepsilon(n) = O\left(\frac{1}{n}\right)$$

and in claim (2-D.10) we showed that

$$\omega(l,n,\delta) = O\left(\left[\frac{nk\ln(nl) - \ln\delta}{l}\right]^{1/2}\right) \;.$$

Therefore the theorem is proved putting these results together. □

Chapter 3

Investigating the Sample Complexity of Active Learning Schemes

Abstract

In the classical learning framework of the previous chapter (akin to PAC) examples were randomly drawn and presented to the learner. In this chapter, we consider the possibility of a more active learner who is allowed to choose his/her own examples. Our investigations can be divided into two natural parts. The first, is in a function approximation setting, and develops an adaptive sampling strategy (equivalent to adaptive approximation) motivated from the standpoint of optimal recovery (Micchelli and Rivlin, 1976). We provide a general formulation of the problem. This can be regarded as sequential optimal recovery. We demonstrate the application of this general formulation to two special cases of functions on the real line 1) monotonically increasing functions and 2) functions with bounded derivative. An extensive investigation of the sample complexity of approximating these functions is conducted yielding both theoretical and empirical results on test functions. Our theoretical results (stated in PAC-style), along with the simulations demonstrate the superiority of our active scheme over both passive learning as well as classical optimal recovery. The second part of this chapter is in a concept learning framework and discusses the idea of ϵ-focusing: a scheme where the active learner can iteratively draw examples from smaller and smaller regions of the input space thereby gaining vast improvements in sample complexity.

In Chapter 2, we considered a learning paradigm where the learner's hypothesis was constrained to belong to a class of functions which can be represented by a sum of radial basis functions. It was assumed that the examples ((x, y) pairs) were drawn according to some fixed, unknown, arbitrary, probability distribution. In this important sense, the learner was merely a *passive* recipient of information about the target function. In this chapter, we consider the possibility of a more *active* learner. There are of course a myriad of ways in which a learner could be more active. Consider, for example, the extreme pathological case where the learner simply asks for the true target function which is duly provided by an obliging oracle. This, the reader will quickly realize is hardly interesting. Such pathological cases aside, this theme of activity on the part of the learner has been explored (though it is not always conceived as such) in a number of different settings (PAC-style concept learning, boundary-hunting pattern recognition schemes, adaptive integration, optimal sampling etc.) in more principled ways and we will comment on these in due course.

For our purposes, we restrict our attention in this chapter to the situation where the learner is allowed to choose its own examples[13], in other words, decide where in the domain D (for functions defined from D to Y) it would like to sample the target function. Note that this is in direct contrast to the passive case where the learner is presented with randomly drawn examples. Keeping other factors in the learning paradigm unchanged, we then compare in this chapter, the active and passive learners who differ only in their method of collecting examples. At the outset, we are particularly interested in whether there exist principled ways of collecting examples in the first place. A second important consideration is whether these ways allow the learner to learn with a fewer number of examples. This latter question is particularly in keeping with the spirit of this thesis, viz., the informational complexity of learning from examples.

This chapter can be divided into two parts which are roughly self-contained. In Part I, we consider active learning in an approximation-theoretic setting. We develop a general framework for collecting examples for approximating (learning) real-valued functions. We then demonstrate the application of these to some specific classes of functions. We obtain theoretical bounds on the sample complexity of the active and passive learners, and perform some empirical simulations to demonstrate the superiority of the active learner. Part II discusses the idea of ϵ-focusing–a paradigm in which the learner iteratively focuses in on specific "interesting" regions of the input space to collect its examples. This is largely in a concept learning (alternatively, pattern classification) setting. We are able to show how using this idea, one can get large gains in sample complexity for some concept classes.

Part I: Active Learning for Approximation of Real Valued Functions

[13] This can be regarded as a computational instantiation of the psychological practice of *selective attention* where a human might choose to selectively concentrate on interesting or confusing regions of the feature space in order to better grasp the underlying concept. Consider, for example, the situation when one encounters a speaker with a foreign accent. One cues in to this foreign speech by focusing on and then adapting to its distinguishing properties. This is often accomplished by asking the speaker to repeat words which are confusing to us.

3.1 A General Framework For Active Approximation

3.1.1 Preliminaries

We need to develop the following notions:
$\mathcal{F}$: Let $\mathcal{F}$ denote a class of functions from some domain D to Y where Y is a subset of the real line. The domain D is typically a subset of R^k though it could be more general than that. There is some unknown target function $f \in \mathcal{F}$ which has to be approximated by an approximation scheme.

$\mathcal{D}$: This is a data set obtained by sampling the target $f \in \mathcal{F}$ at a number of points in its domain. Thus,

$$\mathcal{D} = \{(x_i, y_i) | x_i \in D, y_i = f(x_i),\ i = 1 \dots n\}$$

Notice that the data is uncorrupted by noise.

$\mathcal{H}$: This is a class of functions (also from D to Y) from which the learner will choose one in an attempt to approximate the target f. Notationally, we will use $\mathcal{H}$ to refer not merely to the class of functions (hypothesis class) but also the algorithm by means of which the learner picks an approximating function $h \in \mathcal{H}$ on the basis of the data set $\mathcal{D}$. In other words, $\mathcal{H}$ denotes an approximation scheme which is really a tuple $< \mathcal{H}, A >$. A is an algorithm that takes as its input the data set $\mathcal{D}$, and outputs an $h \in \mathcal{H}$.
Examples: If we consider real-valued functions from R^k to R, some typical examples of $\mathcal{H}$ are the class of polynomials of a fixed order (say q), splines of some fixed order, radial basis functions with some bound on the number of nodes, etc. As a concrete example, consider functions from $[0, 1]$ to R. Imagine a data set is collected which consists of examples, i.e., (x_i, y_i) pairs as per our notation. Without loss of generality, one could assume that $x_i \leq x_{i+1}$ for each i. Then a cubic (degree-3) spline is obtained by interpolating the data points by polynomial pieces (with the pieces tied together at the data points or "knots") such that the overall function is twice-differentiable at the knots. Fig. 3-13 shows an example of an arbitrary data set fitted by cubic splines.

$\mathbf{d_C}$: We need a metric to determine how good the approximation learner's approximation is. Specifically, the metric d_C measures the approximation error on the region

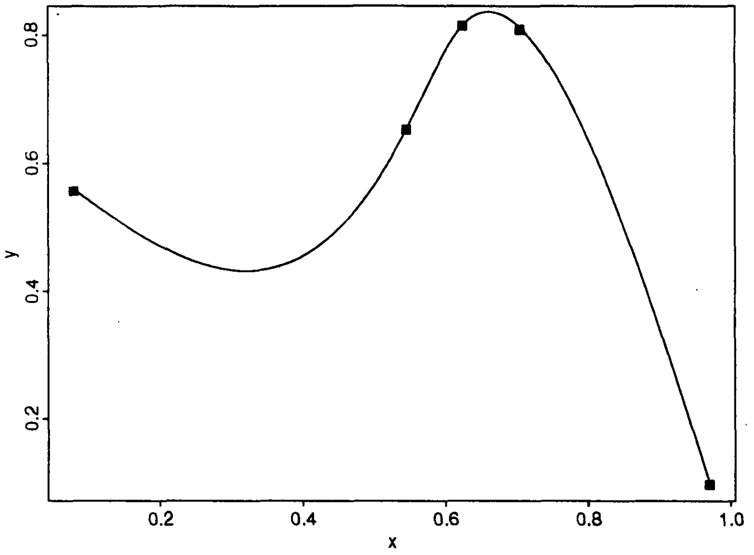

Figure 3-13: An arbitrary data set fitted with cubic splines

C of the domain D. In other words, d_C, takes as its input any two functions (say f_1 and f_2) from D to R and outputs a real number. It is assumed that d_C satisfies all the requisites for being a real distance metric on the appropriate space of functions. Since the approximation error on a larger domain is obviously going to be greater than that on the smaller domain, we can make the following two observations: 1) for any two sets C_1 and C_2 such that $C_1 \subset C_2$, $d_{C_1}(f_1, f_2) \leq d_{C_2}(f_1, f_2)$, 2) $d_D(f_1, f_2)$ is the total approximation on the entire domain; this is our basic criterion for judging the "goodness" of the learner's hypothesis.

Examples: For real-valued functions from R^k to R, the L^p_C metric defined as $d_C(f_1, f_2) = (\int_C |f_1 - f_2|^p dx)^{1/p}$ serves as a natural example of an error metric.

$\mathcal{C}$: This is a collection of subsets C of the domain. We are assuming that points in the domain where the function is sampled, divide (partition) the domain into a collection of disjoint sets $C_i \in \mathcal{C}$ such that $\cup_{i=1}^n C_i = D$.

Examples: For the case of functions from $[0,1]$ to R, and a data set $\mathcal{D}$, a natural way in which to partition the domain $[0,1]$ is into the intervals $[x_i, x_{i+1})$, (here again, without loss of generality we have assumed that $x_i \leq x_{i+1}$). The set $\mathcal{C}$ could be the set of all (closed, open, or half-open and half-closed) intervals $[a,b] \subset [0,1]$.

The goal of the learner (operating with an approximation scheme $\mathcal{H}$) is to provide a hypothesis $h \in \mathcal{H}$ (which it chooses on the basis of its example set $\mathcal{D}$) as an approximator of the unknown target function $f \in \mathcal{F}$. We now need to formally lay down a criterion for assessing the competence of a learner (approximation scheme). In recent times, there has been much use of PAC (Valiant 1984) like criteria to assess learning algorithms. Such a criterion has been used largely for concept learning but some extensions to the case of real valued functions exist (Haussler 1989). We adapt here for our purposes a PAC like criterion to judge the efficacy of approximation schemes of the kind described earlier.

Definition 3.1.1 *An approximation scheme is said to P-PAC learn the function $f \in \mathcal{F}$ if for every $\epsilon > 0$ and $1 > \delta > 0$, and for an arbitrary distribution P on D, it collects a data set $\mathcal{D}$, and computes a hypothesis $h \in \mathcal{H}$ such that $d_D(h, f) < \epsilon$ with probability greater than $1 - \delta$. The function class $\mathcal{F}$ is P-PAC learnable if the approximation scheme can P-PAC learn every function in $\mathcal{F}$. The class $\mathcal{F}$ is PAC learnable if the approximation scheme can P-PAC learn the class for every distribution P.*

There is an important clarification to be made about our definition above. Note that the distance metric d is arbitrary. It need not be naturally related to the distribution P according to which the data is drawn. Recall that this is not so in typical distance metrics used in classical PAC formulations. For example, in concept learning, where the set $\mathcal{F}$ consists of indicator functions, the metric used is the $L_1(P)$ metric given by $d(1_A, 1_B) = \int_D |1_A - 1_B| P(x)dx$. Similarly, extensions to real-valued functions typically use an $L_2(P)$ metric. The use of such metrics imply that the training error is an empirical average of the true underlying error. One can then make use of convergence of empirical means to true means (Vapnik, 1982) and prove learnability. In our case, this is not necessarily the case. For example, one could always come up with a distribution P which would never allow a passive learner to see examples in a certain region of the domain. However, the arbitrary metric d might weigh this region heavily. Thus the learner would never be able to learn such a function class for this metric. In this sense, our model is more demanding than classical PAC. To make matters easy, we will consider here the case of $P - PAC$ learnability alone, where P is a known distribution (uniform in the example cases studied). However, there is a sense in which our notion of PAC is easier —the learner knows the true metric d and given any two functions, can compute their relative distance. This is not so in classical PAC, where the learner cannot compute the distance between two functions since it does not know the underlying distribution.

We have left the mechanism of data collection undefined. Our goal here is the investigation of different methods of data collection. A baseline against which we will compare all such schemes is the *passive* method of data collection where the learner collects its data set by sampling D according to P and receiving the point $(x, f(x))$. If the learner were allowed to draw its own examples, are there principled ways in which it could do this? Further, as a consequence of this flexibility accorded to the learner in its data gathering scheme, could it learn the class $\mathcal{F}$ with fewer examples? These are the questions we attempt to resolve in this chapter, and we begin by motivating and deriving in the next section, a general framework for *active* selection of data for arbitrary approximation schemes.

3.1.2 The Problem of Collecting Examples

We have introduced in the earlier section, our baseline algorithm for collecting examples. This corresponds to a *passive* learner that draws examples according to the probability distribution P on the domain D. If such a passive learner collects examples and produces an output h such that $d_D(h, f)$ is less than ϵ with probability greater than $1 - \delta$, it P-PAC learns the function. The number of examples that a learner needs before it produces such an (ϵ-good,δ-confidence) hypothesis is called its *sample complexity*.

Against this baseline passive data collection scheme, lies the possibility of allowing the learner to choose its own examples. At the outset it might seem reasonable to believe that a data set would provide the learner with some information about the target function; in particular, it would probably inform it about the "interesting" regions of the function, or regions where the approximation error is high and need further sampling. On the basis of this kind of information (along with other information about the class of functions in general) one might be able to decide where to sample next. We formalize this notion as follows:

Let $\mathcal{D} = \{(x_i, y_i); i = 1 \ldots n\}$ be a data set (containing n data points) which the learner has access to. The approximation scheme acts upon this data set and picks an $h \in \mathcal{H}$ (which best fits the data according to the specifics of the algorithm A inherent in the approximation scheme). Further, let $C_i; i = 1, \ldots, K(n)$[14] be a partition of the domain D into different regions on the basis of this data set. Finally let

$$\mathcal{F}_{\mathcal{D}} = \{f \in \mathcal{F} | f(x_i) = y_i \; \forall (x_i, y_i) \in \mathcal{D}\}$$

[14] The number of regions $K(n)$ into which the domain D is partitioned by n data points depends upon the geometry of D and the partition scheme used. For the real line partitioned into intervals as in our example, $K(n) = n + 1$. For k-cubes, one might obtain Voronoi partitions and compute $K(n)$ accordingly.

This is the set of all functions in $\mathcal{F}$ which are consistent with the data seen so far. The target function could be any one of the functions in $\mathcal{F}_{\mathcal{D}}$.

We first define an error criterion e_C (where C is any subset of the domain) as follows:

$$e_C(\mathcal{H}, \mathcal{D}, \mathcal{F}) = \sup_{f \in \mathcal{F}_{\mathcal{D}}} d_C(h, f)$$

Essentially, e_C is a measure of the maximum possible error the approximation scheme could have (over the region C) given the data it has seen so far. It clearly depends on the data, the approximation scheme, and the class of functions being learned. It does not depend upon the target function (except indirectly in the sense that the data is generated by the target function after all, and this dependence is already captured in the expression). We thus have a scheme to measure *uncertainty* (maximum possible error) over the different regions of the input space D. One possible strategy to select a new point might simply be to sample the function in the region C_i where the error bound is the highest. Let us assume we have a procedure $\mathcal{P}$ to do this. $\mathcal{P}$ could be to sample the region C at the centroid of C, or sampling C according to some distribution on it, or any other method one might fancy. This can be described as follows:

Active Algorithm A

1. **[Initialize]** Collect one example (x_1, y_1) by sampling the domain D once according to procedure $\mathcal{P}$.
2. **[Obtain New Partitions]** Divide the domain D into regions $C_1, \ldots, C_{K(1)}$ on the basis of this data point.
3. **[Compute Uncertainties]** Compute e_{C_i} for each i.
4. **[General Update and Stopping Rule]** In general, at the jth stage, suppose that our partition of the domain D is into $C_i, i = 1 \ldots K(j)$. One can compute e_{C_i} for each i and sample the region with maximum uncertainty (say C_l) according to procedure $\mathcal{P}$. This would provide a new data point (x_{j+1}, y_{j+1}). The new data point would re-partition the domain D into new regions. At any stage, if the maximum uncertainty over the entire domain e_D is less than ϵ stop.

The above algorithm is one possible active strategy. However, one can carry the argument a little further and obtain an optimal sampling strategy which would give us a precise location for the next sample point. Imagine for a moment, that the learner asks for the value of the function at a point $x \in D$. The value returned obviously belongs to the set

$$\mathcal{F}_{\mathcal{D}}(x) = \{f(x) | f \in \mathcal{F}_{\mathcal{D}}\}$$

Assume that the value observed was $y \in \mathcal{F}_{\mathcal{D}}(x)$. In effect, the learner now has one more example, the pair (x, y), which it can add to its data set to obtain a new, larger data set $\mathcal{D}'$ where

$$\mathcal{D}' = \mathcal{D} \cup (x, y)$$

Once again, the approximation scheme $\mathcal{H}$ would map the new data set $\mathcal{D}'$ into a new hypothesis h'. One can compute

$$e_C(\mathcal{H}, \mathcal{D}', \mathcal{F}) = \sup_{f \in \mathcal{F}_{\mathcal{D}'}} d(h', f)$$

Clearly, $e_D(\mathcal{H}, \mathcal{D}', \mathcal{F})$ now measures the maximum possible error after seeing this new data point. This depends upon (x, y) (in addition to the usual $\mathcal{H}, \mathcal{D}$, and $\mathcal{F}$). For a fixed x, we don't know the value of y we would observe if we had chosen to sample at that point. Consequently, a natural thing to do at this stage is to again take a worst case bound, i.e., assume we would get the most unfavorable y and proceed. This would provide the maximum possible error we could make if we had chosen to sample at x. This error (over the entire domain) is

$$\sup_{y \in \mathcal{F}_{\mathcal{D}}(x)} e_D(\mathcal{H}, \mathcal{D}', \mathcal{F}) = \sup_{y \in \mathcal{F}_{\mathcal{D}}(x)} e_D(\mathcal{H}, \mathcal{D} \cup (x, y), \mathcal{F})$$

Naturally, we would like to sample the point x for which this maximum error is minimized. Thus, the optimal point to sample by this argument is

$$x_{new} = \arg\min_{x \in D} \sup_{y \in \mathcal{F}_{\mathcal{D}}(x)} e_D(\mathcal{H}, \mathcal{D} \cup (x, y), \mathcal{F}) \tag{3.27}$$

This provides us with a principled strategy to choose our next point. The following optimal active learning algorithm follows:

Active Algorithm B (Optimal)

1. **[Initialize]** Collect one example (x_1, y_1) by sampling the domain D once according to procedure $\mathcal{P}$. We do this because without any data, the approximation scheme would not be able to produce any hypothesis.

2. **[Compute Next Point to Sample]** Apply eq. 3.27 and obtain x_2. Sampling the function at this point yields the next data point (x_2, y_2) which is added to the data set.

3. **[General Update and Stopping Rule]** In general, at the jth stage, assume we have in place a data set $\mathcal{D}_j$ (consisting of j data). One can compute x_{j+1} according to eq. 3.27 and sampling the function here one can obtain a new hypothesis and a new data set $\mathcal{D}_{j+1}$. In general, as in Algorithm A, stop whenever the total error $e_D(\mathcal{H}, \mathcal{D}_k, \mathcal{F})$ is less than ϵ.

By the process of derivation, it should be clear that if we chose to sample at some point other than that obtained by eq. 3.27, an adversary could provide a y value and a function consistent with all the data provided (including the new data point), that would force the learner to make a larger error than if the learner chose to sample at x_{new}. In this sense, algorithm B is optimal. It also differs from algorithm A, in that it does not require a partition scheme, or a procedure $\mathcal{P}$ to choose a point in some region. However, the computation of x_{new} inherent in algorithm B is typically more intensive than computations required by algorithm A. Finally, it is worthwhile to observe that crucial to our formulation is the derivation of the error bound $e_D(\mathcal{H}, \mathcal{D}, \mathcal{F})$. As we have noted earlier, this is a measure of the maximum possible error the approximation scheme $\mathcal{H}$ could be forced to make in approximating functions of $\mathcal{F}$ using the data set $\mathcal{D}$. Now, if one wanted an approximation scheme independent bound, this would be obtained by minimizing e_D over all possible schemes, i.e.,

$$\inf_{\mathcal{H}} e_D(\mathcal{H}, \mathcal{D}, \mathcal{F})$$

Any approximation scheme can be forced to make at least as much error as the above expression denotes. Another bound of some interest is obtained by removing the dependence of e_D on the data. Thus given an approximation scheme $\mathcal{H}$, if data $\mathcal{D}$ is drawn randomly, one could compute

$$P\{e_D(\mathcal{H}, \mathcal{D}, \mathcal{F}) > \epsilon\}$$

or in an approximation scheme-independent setting, one computes

$$P\{\inf_{\mathcal{H}} e_D(\mathcal{H}, \mathcal{D}, \mathcal{F}) > \epsilon\}$$

The above expressions would provide us PAC-like bounds which we will make use of later in this chapter.

3.1.3 In Context

Having motivated and derived two possible active strategies, it is worthwhile at this stage to comment on the formulation and its place in the context of previous work in similar vein executed across a number of disciplines.

1) Optimal Recovery: The question of choosing the location of points where the unknown function will be sampled has been studied within the framework of optimal recovery (Micchelli and Rivlin, 1976; Micchelli and Wahba, 1981; Athavale and Wahba, 1979). While work of this nature has strong connections to our formulation, there remains a crucial difference. Sampling schemes motivated by optimal recovery are not adaptive. In other words, given a class of functions $\mathcal{F}$ (from which the target f is selected), optimal sampling chooses the points $x_i \in D, i = 1, \ldots, n$ by optimizing over the entire function space $\mathcal{F}$. Once these points are obtained, then they remain fixed irrespective of the target (and correspondingly the data set $\mathcal{D}$). Thus, if we wanted to sample the function at n points, and had an approximation scheme $\mathcal{H}$ with which we wished to recover the true target, a typical optimal recovery formulation would involve sampling the function at the points obtained as a result of optimizing the following objective function:

$$\arg \min_{x_1, \ldots, x_n} \sup_{f \in \mathcal{F}} d(f, h(\mathcal{D} = \{(x_i, f(x_i))_{i=1 \ldots n}\})) \tag{3.28}$$

where $h(\mathcal{D} = \{(x_i, f(x_i))_{i=1 \ldots n}\}) \in \mathcal{H}$ is the learner's hypothesis when the target is f and the function is sampled at the x_i's. Given no knowledge of the target, these points are the optimal to sample.

In contrast, our scheme of sampling can be conceived as an iterative application of optimal recovery (one point at a time) by conditioning on the data seen so far. Making this absolutely explicit, we start out by asking for one point using optimal recovery. We obtain this point by

$$\arg \min_{x_1} \sup_{f \in \mathcal{F}} d(f, h(\mathcal{D}_1 = \{(x_1, f(x_1))\}))$$

Having sampled at this point (and obtained y_1 from the true target), we can now reduce the class of candidate target functions to $\mathcal{F}_1$, the elements of $\mathcal{F}$ which are consistent with the data seen so far. Now we obtain our second point by

$$\arg \min_{x_2} \sup_{f \in \mathcal{F}_1} d(f, h(\mathcal{D}_2 = \{(x_1, y_1), (x_2, f(x_2))\}))$$

Note that the supremum is done over a restricted set $\mathcal{F}_1$ the second time. In this fashion, we perform optimal recovery at each stage, reducing the class of functions over which the supremum is performed. It should be made clear that this sequential optimal recovery is *not* a greedy technique to arrive at the solution of eq. 3.28. It will give us a different set of points. Further, this set of points will depend upon the target function. In other words,the sampling strategy adapts itself to the unknown target f as it gains more information about that target through the data. We know of no similar sequential sampling scheme in the literature.

While classical optimal recovery has the formulation of eq. 3.28, imagine a situation where a "teacher" who knows the target function and the learner, wishes to communicate to the learner the best set of points to minimize the error made by the learner. Thus given a function g, this best set of points can be obtained by the following optimization

$$\arg \min_{x_1,\ldots,x_n} d(g, h(\{(x_i, g(x_i))\}_{i=1\ldots n})) \tag{3.29}$$

Eq. 3.28 and eq. 3.29 provide two bounds on the performance of the active learner following the strategy of Algorithm B in the previous section. While eq. 3.28 chooses optimal points without knowing anything about the target, and, eq. 3.29 chooses optimal points knowing the target completely, the active learner chooses points optimally on the basis of partial information about the target (information provided by the data set).

2) Concept Learning: The PAC learning community (which has traditionally focused on concept learning) typically incorporates activity on the part of the learner by means of queries, the learner can make of an oracle. Queries (Angluin, 1988) range from membership queries (is x an element of the target concept c) to statistical queries (Kearns, 1993 ; where the learner can not ask for data but can ask for estimates of functionals of the function class) to arbitrary boolean valued queries (see Kulkarni etal for an investigation of query complexity). Our form of activity can be considered as a natural adaptation of membership queries to the case of learning real-valued functions in our modified PAC model. It is worthwhile to mention relevant work which touches the contents of this chapter at some points. The most significant of these is an investigation of the sample complexity of active versus passive learning conducted by Eisenberg and Rivest (1990) for a simple class of unit step functions. It was found that a binary search algorithm could vastly outperform a passive learner in terms of the number of examples it needed to (ϵ, δ) learn the target function. This chapter is very much in the spirit of that work focusing as it does on the sample complexity question. Another interesting direction is the transformation of PAC-learning algorithms from a batch to online mode. While Littlestone etal (1991) consider online learning of linear functions, Kimber and Long (1992) consider functions with bounded derivatives which we examine later in this chapter. However the question of choosing one's data is not addressed at all. Kearns and Schapire (1990) consider the learning of p-concepts (which are essentially equivalent to learning classes of real-valued functions with noise) and address the learning of monotone functions in this context. Again, there is no active component on the part of the learner.

3)Adaptive Integration: The novelty of our formulation lies in its adaptive nature. There are some similarities to work in adaptive numerical integration which are worth mentioning. Roughly speaking, an adaptive integration technique (Berntsen et al 1991) divides the domain of integration into regions over which the integration is done. Estimates are then obtained of the error on each of these regions. The region with maximum error is subdivided. Though the spirit of such an adaptive approach is close to ours, specific results in the field naturally differ because of differences between the integration problem (and its error bounds) and the approximation problem.

4) Bayesian and other formulations: It should be noted that we have a worst-case formulation (the supremum in our formulation represents the maximum possible error the scheme might have). Alternate bayesian schemes have been devised (Mackay, 1991; Cohn, 1994) from the perspective of optimal experiment design (Fedorov). Apart from the inherently different philosophical positions of the two schemes, an indepth treatment of the sample complexity question is not done. We will soon give two examples where we address this sample complexity question closely. In a separate piece of work (Sung and Niyogi, 1994) , the author has also investigated such bayesian formulations from such an information-theoretic perspective. Yet another average-case formulation comes from the information-complexity viewpoint of Traub and Wozniakovski (see Traub etal (1988) for details). Various interesting sampling strategies are suggested by research in that spirit. We do not attempt to compare them due to the difficulty in comparing worst-case and average-case bounds.

Thus, we have motivated and derived in this section, two possible active strategies. The formulation is general. We now demonstrate the usefulness of such a formulation by considering two classes of real-valued functions as examples and deriving specific active algorithms from this perspective. At this stage, the important question of sample complexity of active versus passive learning still remains unresolved. We investigate this more closely by deriving theoretical bounds and performing empirical simulation studies in the case of the specific classes we consider.

3.2 Example 1: A Class of Monotonically Increasing Bounded Functions

Consider the following class of functions from the interval $[0,1] \subset \Re$ to $\Re$:

$$\mathcal{F} = \{f : 0 \leq f \leq M, \text{ and } f(x) \geq f(y) \forall x \geq y\}$$

Note that the functions belonging to this class need not be continuous though they do need to be measurable. This class is PAC- learnable (with an $L_1(P)$ norm, in which case our notion of PAC reduces to the classical notion) though it has infinite pseudo-dimension[15](in the sense of Pollard (1984)). Thus, we observe:

Observation 1 *The class $\mathcal{F}$ has infinite pseudo-dimension (in the sense of Pollard (1984); Haussler (1989),).*

Proof: To have infinite pseudo-dimension, it must be the case that for every $n > 0$, there exists a set of points $\{x_1, \ldots, x_n\}$ which is shattered by the class $\mathcal{F}$. In other words, there must exist a fixed translation vector $\mathbf{t} = (t_1, \ldots, t_n)$ such that for every boolean vector $\mathbf{b} = (b_1, \ldots, b_n)$, there exists a function $f \in \mathcal{F}$ which satisfies $f(x_i) - t_i > 0 \Leftrightarrow b_i = 1$. To see that this is indeed the case, let the n points be $x_i = i/(n+1)$ for i going from 1 to n. Let the translation vector then be given by $t_i = x_i$. For an arbitrary boolean vector $\mathbf{b}$ we can always come up with a monotonic function such that $f(x_i) = i/(n+1) - 1/3(n+1)$ if $b_i = 0$ and $f(x_i) = i/(n+1) + 1/3(n+1)$ if $b_i = 1$. □

We also need to specify the terms $\mathcal{H}$, d_C, the procedure $\mathcal{P}$ for partitioning the domain $D = [0,1]$ and so on. For our purposes, we assume that the approximation scheme $\mathcal{H}$ is first order splines. This is simply finding the monotonic function which interpolates the data in a piece-wise linear fashion. A natural way to partition the domain is to divide it into the intervals $[0, x_1)$, $[x_1, x_2), \ldots, [x_i, x_{i+1}), \ldots, [x_n, 1]$. The metric d_C is an L_p metric given by $d_C(f_1, f_2) = (\int_0^1 |f_1 - f_2|pdx)^{1/p}$.

Note that we are specifically interested in comparing the sample complexities of passive and active learning. We will do this under a uniform distributional assumption, i.e., the *passive* learner draws its examples by sampling the target function uniformly at random on its domain $[0,1]$. In contrast, we will show how our general formulation in the earlier section translates into a specific *active* algorithm for choosing points, and we derive bounds on its sample complexity. We begin by first

[15] Finite pseudo-dimension is only a sufficient and not necessary condition for PAC learnability as this example demonstrates.

providing a lower bound for the number of examples a passive PAC learner would need to draw to learn this class $\mathcal{F}$.

3.2.1 Lower Bound for Passive Learning

Theorem 3.2.1 *Any passive learning algorithm (more specifically, any approximation scheme which draws data uniformly at random and interpolates the data by any arbitrary bounded function) will have to draw at least $\frac{1}{2}(M/2\epsilon)^p \ln(1/\delta)$ examples to P-PAC learn the class where P is a uniform distribution.*

Proof: Consider the uniform distribution on $[0,1]$ and a subclass of functions which have value 0 on the region A $= [0, 1-(2\epsilon)^p]$ and belong to $\mathcal{F}$. Suppose the passive learner draws l examples uniformly at random. Then with probability $(1-(2\epsilon/M)^p)^l$, all these examples will be drawn from region A. It only remains to show that for the subclass considered, whatever be the function hypothesized by the learner, an adversary can force it to make a large error.

Suppose the learner hypothesizes that the function is h. Let the value of $(\int_{(1-(2\epsilon/M)^p,1)} |h(x)|^p dx)^{1/p}$ be χ. Obviously $0 \leq \chi \leq (M^p(2\epsilon/M)^p)^{1/p} = 2\epsilon$. If $\chi < \epsilon$, then the adversary can claim that the target function was really

$$g(x) = \begin{cases} 0 & \text{for } x \in [0, 1-(2\epsilon/M)^p] \\ M & \text{for } x \in (1-(2\epsilon/M)^p, 1] \end{cases}$$

If, on the other hand $\chi \geq \epsilon$, then the adversary can claim the function was really $g = 0$.

In the first case, by the triangle inequality,

$$d(h,g) = (\int_{[0,1]} |g-h|^p dx)^{1/p} \geq (\int_{[1-(2\epsilon/M)^p,1]} |g-h|^p dx)^{1/p}$$

$$\geq (\int_{(1-(2\epsilon/M)^p,1)} M^p dx)^{1/p} - (\int_{(1-(2\epsilon/M)^p,1)} |h|^p dx)^{1/p} = 2\epsilon - \chi > \epsilon$$

In the second case,

$$d(h,g) = (\int_{[0,1]} |g-h|^p dx)^{1/p} \geq (\int_{(1-(2\epsilon/M)^p,1)} |0-h|^p dx)^{1/p} = \chi > \epsilon$$

Now we need to find out how large l must be so that this particular event of drawing all examples in A is not very likely, in particular, it has probability less than δ.

For $(1-(2\epsilon/M)^p)^l$ to be greater than δ, we need $l < \frac{1}{-\ln(1-(2\epsilon/M)^p)} \ln(\frac{1}{\delta})$. It is a fact that for $\alpha < 1/2$, $\frac{1}{2\alpha} \leq \frac{1}{-\ln(1-\alpha)}$. Making use of this fact (and setting $\alpha = (2\epsilon/M)^p$, we see that for $\epsilon < (\frac{M}{2})(\frac{1}{2})^{1/p}$, we have $\frac{1}{2}(M/2\epsilon)^p \ln(1/\delta) < \frac{1}{-\ln(1-(2\epsilon/M)^p)} \ln(\frac{1}{\delta})$. So unless l is greater than $\frac{1}{2}(M/2\epsilon)^p \ln(1/\delta)$, the probability that all examples are chosen from A is greater than δ. Consequently, with probability greater than δ, the passive learner is forced to make an error of atleast ϵ, and PAC learning cannot take place. □

3.2.2 Active Learning Algorithms

In the previous section we computed a lower bound for passively PAC learning this class for a uniform distribution[16]. Here we derive an active learning strategy (the CLA algorithm) which would meaningfully choose new examples on the basis of information gathered about the target from previous examples. This is a specific instantiation of the general formulation, and interestingly yields a "divide and conquer" binary searching algorithm starting from a different philosophical standpoint. We formally prove an upper bound on the number of examples it requires to PAC learn the class. While this upper bound is a worst case bound and holds for all functions in the class, the actual number of queries (examples) this strategy takes differs widely depending upon the target function. We demonstrate empirically the performance of this strategy for different kinds of functions in the class in order to get a feel for this difference. We derive a classical non-sequential optimal sampling strategy and show that this is equivalent to uniformly sampling the target function. Finally, we are able to empirically demonstrate that the active algorithm outperforms both the passive and uniform methods of data collection.

Derivation of an optimal sampling strategy

Consider an approximation scheme of the sort described earlier attempting to approximate a target function $f \in \mathcal{F}$ on the basis of a data set $\mathcal{D}$. Shown in fig. 3-14 is a picture of the situation. We can assume without loss of generality that we start out by knowing the value of the function at the points $x = 0$ and $x = 1$. The points $\{x_i; i = 1, \ldots, n\}$ divide the domain into $n + 1$ intervals C_i (i going from 0 to n) where $C_i = [x_i, x_{i+1}] (x_0 = 0, x_{n+1} = 1)$.The monotonicity constraint on $\mathcal{F}$ permits us to obtain rectangular boxes showing the values that the target function could take at the points on its domain. The set of all functions which lie within these boxes as shown is $\mathcal{F}_{\mathcal{D}}$.

Let us first compute $e_{C_i}(\mathcal{H}, \mathcal{D}, \mathcal{F})$ for some interval C_i. On this interval, the func-

[16] Naturally, this is a distribution-free lower bound as well. In other words, we have demonstrated the existence of a distribution for which the passive learner would have to draw at least as many examples as the theorem suggests.

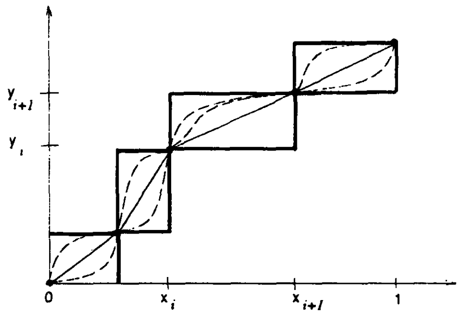

Figure 3-14: A depiction of the situation for an arbitrary data set. The set $\mathcal{F}_\mathcal{D}$ consists of all functions lying in the boxes and passing through the datapoints (for example, the dotted lines). The approximating function h is a linear interpolant shown by a solid line.

tion is constrained to lie in the appropriate box. We can zoom in on this box as shown in fig. 3-15.

The maximum error the approximation scheme could have (indicated by the shaded region) is clearly given by

$$(\int_{C_i} |h - f(x_i)|^p dx)^{1/p} = (\int_0^B (\frac{A}{B}x)^p dx)^{1/p} = AB^{1/p}/(p+1)^{1/p}$$

where $A = f(x_{i+1}) - f(x_i)$ and $B = (x_{i+1} - x_i)$.

Clearly the error over the entire domain e_D is given by

$$e_D^p = \sum_{i=0}^{n} e_{C_i}^p \tag{3.30}$$

The computation of e_C is all we need to implement an active strategy motivated by Algorithm A in section 3.1. All we need to do is sample the function in the interval with largest error; recall that we need a procedure $\mathcal{P}$ to determine how to sample this interval to obtain a new data point. We choose (arbitrarily) to sample the midpoint of the interval with the largest error yielding the following algorithm.

The Choose and Learn Algorithm (CLA)

1. **[Initial Step]** Ask for values of the function at points $x = 0$ and $x = 1$. At this stage, the domain $[0, 1]$ is composed of one interval only, viz., $[0, 1]$. Compute $E_1 = \frac{1}{(p+1)^{1/p}}(1 - 0)^{1/p}|(f(1) - f(0))|$ and $T_1 = E_1$. If $T_1 < \epsilon$, stop and output the linear interpolant of the samples as the hypothesis, otherwise query the midpoint of the interval to get a partition of the domain into two subintervals $[0, 1/2)$ and $[1/2, 1]$.

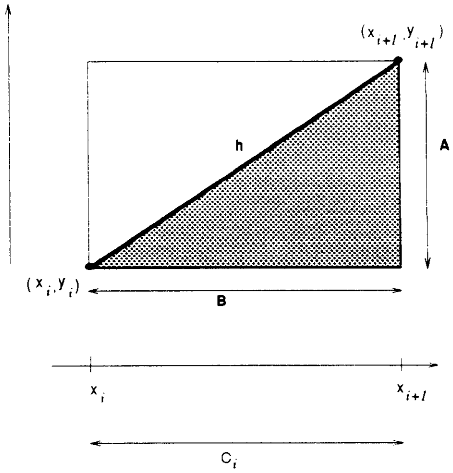

Figure 3-15: Zoomed version of interval. The maximum error the approximation scheme could have is indicated by the shaded region. This happens when the adversary claims the target function had the value y_i throughout the interval.

2. **[General Update and Stopping Rule]** In general, at the kth stage, suppose that our partition of the interval $[0, 1]$ is $[x_0 = 0, x_1)$,$[x_1, x_2)$, $\ldots$, $[x_{k-1}, x_k = 1]$. We compute the normalized error $E_i = \frac{1}{(p+1)^{1/p}}(x_i - x_{i-1})^{1/p}|(f(x_i) - f(x_{i-1}))|$ for all $i = 1,..,k$. The midpoint of the interval with maximum E_i is queried for the next sample. The total normalized error $T_k = (\sum_{i=1}^{k} E_i^p)^{1/p}$ is computed at each stage and the process is terminated when $T_k \leq \epsilon$. Our hypothesis h at every stage is a linear interpolation of all the points sampled so far and our final hypothesis is obtained upon the termination of the whole process.

Now imagine that we chose to sample at a point $x \in C_i = [x_i, x_{i+1}]$ and received the value $y \in \mathcal{F}_{\mathcal{D}}(x)$ (i.e., y in the box) as shown in the fig. 3-16. This adds one more interval and divides C_i into two subintervals C_{i1} and C_{i2} where $C_{i1} = [x_i, x]$ and $C_{i2} = [x, x_{i+1}]$. We also correspondingly obtain two smaller boxes inside the larger box within which the function is now constrained to lie. The uncertainty measure e_C can be recomputed taking this into account.

Observation 2 *The addition of the new data point (x, y) does not change the uncertainty value on any of the other intervals. It only affects the interval C_i which got subdivided. The total uncertainty over this interval is now given by*

$$e_{C_i}(\mathcal{H}, \mathcal{D}', \mathcal{F}) = (\tfrac{1}{p+1})^{1/p}((x - x_i)(y - f(x_i))^p + (x_{i+1} - x))((f(x_{i+1}) - f(x_i)) - y)^p)^{1/p} =$$

$$= G\,(zr^p + (B - z)(A - r)^p)^{1/p}$$

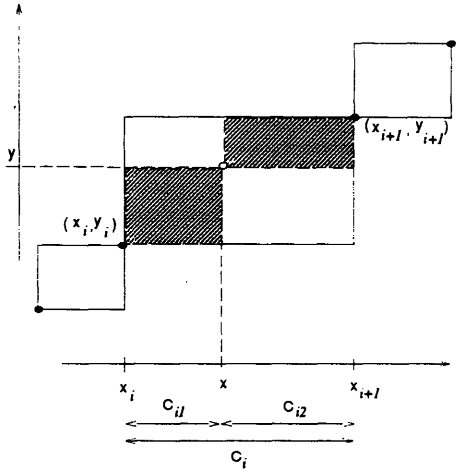

Figure 3-16: The situation when the interval C_i is sampled yielding a new data point. This subdivides the interval into two subintervals and the two shaded boxes indicate the new constraints on the function.

where for convenience we have used the substitution $z = x - x_i$, $r = y - f(x_i)$, and A and B are $f(x_{i+1}) - f(x_i)$ and $x_{i+1} - x_i$ as above. Clearly z ranges from 0 to B while r ranges from 0 to A.

We first prove the following lemma:

Lemma 3.2.1

$$B/2 = \arg \min_{z\in[0,B]} \sup_{r\in[0,A]} G\,(zr^p + (B-z)(A-r)^p)^{1/p}$$

Proof: Consider any $z \in [0, B]$. There are three cases to consider:
Case I $z > B/2$: let $z = B/2 + \alpha$ where $\alpha > 0$. We find

$$\sup_{r\in[0,A]} G\,(zr^p + (B-z)(A-r)^p)^{1/p} = \left(\sup_{r\in[0,A]} G\,(zr^p + (B-z)(A-r)^p)\right)^{1/p}$$

Now,

$$\sup\nolimits_{r\in[0,A]} G\,(zr^p + (B-z)(A-r)^p) =$$

$$\sup\nolimits_{r\in[0,A]} G\,((B/2+\alpha)r^p + (B/2-\alpha)(A-r)^p)$$

$$= G \sup\nolimits_{r\in[0,A]} B/2(r^p + (A-r)^p) + \alpha(r^p - (A-r)^p)$$

Now for $r = A$, the expression within the supremum $B/2(r^p+(A-r)^p)+\alpha(r^p-(A-r)^p)$

is equal to $(B/2+\alpha)A^p$. For any other $r \in [0, A]$, we need to show that

$$B/2(r^p + (A-r)^p) + \alpha(r^p - (A-r)^p) \leq (B/2+\alpha)A^p$$

or

$$B/2((r/A)^p + (1-(r/A))^p) + \alpha((r/A)^p - (1-r/A)^p) \leq B/2+\alpha$$

Putting $\beta = r/A$ (clearly $\beta \in [0,1]$, and noticing that $(1-\beta)^p \leq 1-\beta^p$ and $\beta^p - (1-\beta)^p \leq 1$ the inequality above is established. Consequently, we are able to see that

$$\sup_{r\in[0,A]} G(zr^p + (B-z)(A-r)^p)^{1/p} = G(B/2+\alpha)^{1/p}A$$

Case II Let $z = B/2 - \alpha$ for $\alpha > 0$. In this case, by a similar argument as above, it is possible to show that again,

$$\sup_{r\in[0,A]} G(zr^p + (B-z)(A-r)^p)^{1/p} = G(B/2+\alpha)^{1/p}A$$

Case III Finally, let $z = B/2$. Here

$$\sup_{r\in[0,A]} G(zr^p + (B-z)(A-r)^p)^{1/p} = G(B/2)^{1/p} \sup_{r\in[0,A]} (r^p + (A-r)^p)^{1/p}$$

Clearly, then for this case, the above expression is reduced to $GA(B/2)^{1/p}$. Considering the three cases, the lemma is proved.□

The above lemma in conjunction with eq. 3.30 and observation 2 proves that if we choose to sample a particular interval C_i then sampling the midpoint is the optimal thing to do. In particular, we see that

$$\min_{x\in[x_i,x_{i+1}]} \sup_{y\in[f(x_i),f(x_{i+1})]} e_{C_i}(\mathcal{H}, \mathcal{D} \cup (x,y), \mathcal{F}) =$$

$$(\tfrac{1}{p+1})^{1/p}(\tfrac{x_{i+1}-x_i}{2})^{1/p}(f(x_{i+1}) - f(x_i)) = e_{C_i}(\mathcal{H}, \mathcal{D}, \mathcal{F})/2^{1/p}$$

In other words, if the learner were constrained to pick its next sample in the interval C_i, then by sampling the midpoint of this interval C_i, the learner ensures that the maximum error it could be forced to make by a malicious adversary is minimized. In particular, if the uncertainty over the interval C_i with its current data set $\mathcal{D}$ is e_{C_i}, the uncertainty over this region will be reduced after sampling its midpoint and can have a maximum value of $e_{C_i}/2^{1/p}$.

Now which interval must the learner sample to minimize the maximum possible uncertainty over the entire domain $D = [0,1]$. Noting that if the learner chose to sample the interval C_i then

$$\min_{x \in C_i = [x_i, x_{i+1}]} \sup_{y \in \mathcal{F}_\mathcal{D}} e_{D=[0,1]}(\mathcal{H}, \mathcal{D} \cup (x,y), \mathcal{F}) = \left(\sum_{j=0, j \neq i}^{n} e^p_{C_j}(\mathcal{H}, \mathcal{D}, \mathcal{F}) + \frac{e^p_{C_i}(\mathcal{H}, \mathcal{D}, \mathcal{F})}{2} \right)^{1/p}$$

From the decomposition above, it is clear that the optimal point to sample according to the principle embodied in Algorithm B is the midpoint of the interval C_j which has the maximum uncertainty $e_{C_j}(\mathcal{H}, \mathcal{D}, \mathcal{F})$ on the basis of the data seen so far, i.e., the data set $\mathcal{D}$. Thus we can state the following theorem

Theorem 3.2.2 *The CLA is the optimal algorithm for the class of monotonic functions*

Having thus established that our binary searching algorithm (CLA) is optimal, we now turn our efforts to determining the number of examples the CLA would need in order to learn the unknown target function to ϵ accuracy with δ confidence. In particular, we can prove the following theorem.

Theorem 3.2.3 *The CLA converges in at most $(M/\epsilon)^p$ steps. Specifically, after collecting at most $(M/\epsilon)^p$ examples, its hypothesis is ϵ close to the target with probability 1.*

Proof Sketch: The proof of convergence for this algorithm is a little tedious. However, to convince the reader, we provide the proof of convergence for a slight variant of the active algorithm. It is possible to show (not shown here) that convergence times for the active algorithm described earlier is bounded by the convergence time for the variant. First, consider a uniform grid of points $(\epsilon/M)^p$ apart on the domain $[0, 1]$. Now imagine that the active learner works just as described earlier but with a slight twist, viz., it can only query points on this grid. Thus at the kth stage, instead of querying the true midpoint of the interval with largest uncertainty, it will query the gridpoint closest to this midpoint. Obviously the intervals at the kth stage are also separated by points on the grid (i.e. previous queries). If it is the case that the learner has queried all the points on the grid, then the maximum possible error it could make is less than ϵ. To see this, let $\alpha = \epsilon/M$ and let us first look at a specific small interval $[k\alpha, (k+1)\alpha]$. We know the following to be true for this subinterval:

$$f(k\alpha) = h(k\alpha) \leq f(x), h(x) \leq f((k+1)\alpha) = h((k+1)\alpha)$$

Thus

$$|f(x) - h(x)| \leq f((k+1)\alpha) - f(k\alpha)$$

and so over the interval $[k\alpha, (k+1)\alpha]$

$$\int_{k\alpha}^{(k+1)\alpha} |f(x) - h(x)|^p dx \leq \int_{k\alpha}^{(k+1)\alpha} (f((k+1)\alpha) - f(k\alpha))^p dx$$

$$\leq (f((k+1)\alpha) - f(k\alpha))^p \alpha$$

It follows that

$$\int_{[0,1]} |f - h|^p dx = \int_{[0,\alpha)} |f - h|^p dx + \ldots + \int_{[1-\alpha,1]} |f - h|^p dx \leq$$

$$\alpha\left((f(\alpha) - f(0))^p + (f(2\alpha) - f(\alpha))^p + \ldots + (f(1) - f(1-\alpha))^p\right) \leq$$

$$\alpha(f(\alpha) - f(0) + f(2\alpha) - f(\alpha) + \ldots + f(1) - f(1-\alpha))^p \leq$$

$$\leq \alpha(f(1) - f(0))^p \leq \alpha M^p$$

So if $\alpha = (\epsilon/M)^p$, we see that the L_p error would be at most $\left(\int_{[0,1]} |f - h|^p dx\right)^{1/p} \leq \epsilon$. Thus the active learner moves from stage to stage collecting examples at the grid points. It could converge at any stage, but clearly after it has seen the value of the unknown target at all the gridpoints, its error is provably less than ϵ and consequently it must stop by this time. □

3.2.3 Empirical Simulations, and other Investigations

Our aim here is to characterize the performance of CLA as an active learning strategy. Remember that CLA is an adaptive example choosing strategy and the number of samples it would take to converge depends upon the specific nature of the target function. We have already computed an upper bound on the number of samples it would take to converge in the worst case. In this section we try to provide some intuition as to how this sampling strategy differs from random draw of points (equivalent to passive learning) or drawing points on a uniform grid (equivalent to optimal recovery following eq. 3.28 as we shall see shortly). We perform simulations on arbitrary monotonic increasing functions to better characterize conditions under which the active strategy could outperform both a passive learner as well as a uniform learner.

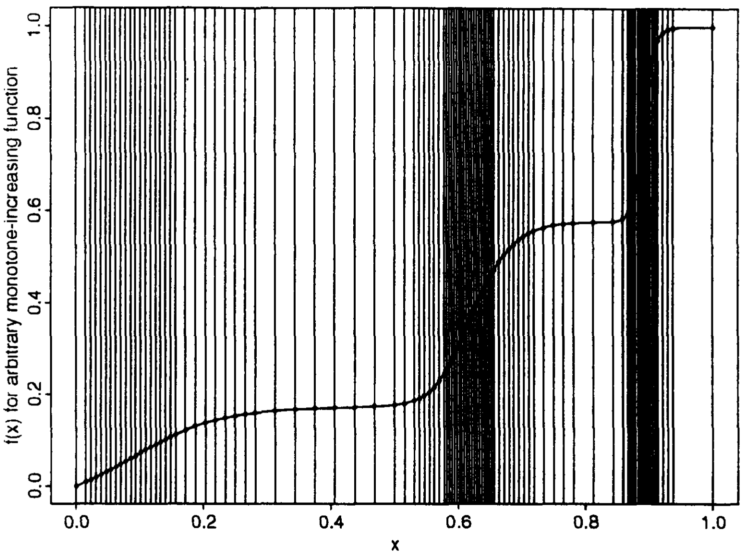

Figure 3-17: How the CLA chooses its examples. Vertical lines have been drawn to mark the x-coordinates of the points at which the algorithm asks for the value of the function.

Distribution of Points Selected

As has been mentioned earlier, the points selected by CLA depend upon the specific target function.Shown in fig. 3-5 is the performance of the algorithm for an arbitrarily constructed monotonically increasing function. Notice the manner in which it chooses its examples. Informally speaking, in regions where the function changes a lot (such regions can be considered to have high information density and consequently more "interesting"), CLA samples densely. In regions where the function doesn't change much (correspondingly low information density), it samples sparsely. As a matter of fact, the density of the points seems to follow the derivative of the target function as shown in fig. 3-18.

Consequently, we conjecture that

Conjecture 1 *The density of points sampled by the active learning algorithm is proportional to the derivative of the function at that point for differentiable functions.*

Remarks:

1. The CLA seems to sample functions according to its rate of change over the different regions. We have remarked earlier, that the best possible sampling strategy would be obtained by eq. 3.29 earlier. This corresponds to a teacher

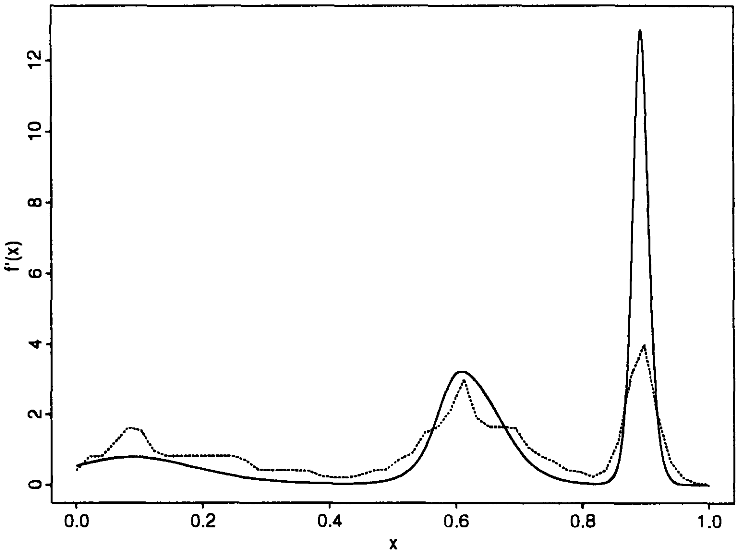

Figure 3-18: The dotted line shows the density of the samples along the x-axis when the target was the monotone-function of the previous example. The bold line is a plot of the derivative of the function. Notice the correlation between the two.

(who knows the target function and the learner) selecting points for the learner. How does the CLA sampling strategy differ from the best possible one? Does the sampling strategy converge to the best possible one as the data goes to infinity? In other words, does the CLA discover the best strategy? These are interesting questions. We do not know the answer.

2. We remarked earlier that another bound on the performance of the active strategy was that provided by the classical optimal recovery formulation of eq. 3.28. This, as we shall show in the next section, is equivalent to uniform sampling. We remind the reader that a crucial difference between uniform sampling and CLA lies in the fact that CLA is an adaptive strategy and for some functions might actually learn with very few examples. We will explore this difference soon.

Classical Optimal Recovery

For an L_1 error criterion, classical optimal recovery as given by eq. 3.28 yields a uniform sampling strategy. To see this, imagine that we chose to sample the function at points $x_i; i = 1, \ldots, n$. Pick a possible target function f and let $y_i = f(x_i)$ for each i. We then get the situation depicted in fig. 3-19. The n points divide the domain into

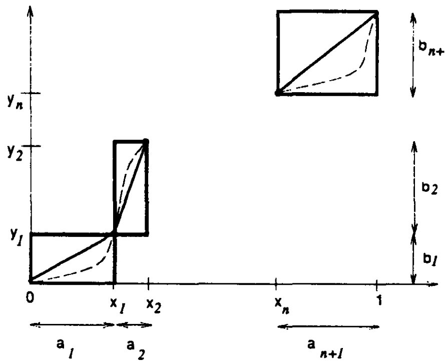

Figure 3-19: The situation when a function $f \in \mathcal{F}$ is picked, n sample points (the x's) are chosen and the corresponding y values are obtained. Each choice of sample points corresponds to a choice of the a's. Each choice of a function corresponds to a choice of the b's.

$n+1$ intervals. Let these intervals have length a_i each as shown. Further, if $[x_{i-1}, x_i]$ corresponds to the interval of length a_i, then let $y_i - y_{i-1} = b_i$. In other words we would get $n+1$ rectangles with sides a_i and b_i as shown in the figure.

It is clear that choosing a vector $\mathbf{b} = (b_1, \dots, b_{n+1})'$ with the constraint that $\sum_{i=1}^{n+1} b_i = M$ and $b_i \geq 0$ is equivalent to defining a set of y values (in other words, a data set) which can be generated by some function in the class $\mathcal{F}$. Specifically, the data values at the respective sample points would be given by $y_1 = b_1$, $y_2 = b_1 + b_2$ and so on. We can define $\mathcal{F}_{\mathbf{b}}$ to be the set of monotonic functions in $\mathcal{F}$ which are consistent with these data points. In fact, every $f \in \mathcal{F}$ would map onto some $\mathbf{b}$, and thus belong to some $\mathcal{F}_{\mathbf{b}}$. Consequently,

$$\mathcal{F} = \cup_{\{\mathbf{b}: b_i \geq 0, \sum b_i = M\}} \mathcal{F}_{\mathbf{b}}$$

Given a target function $f \in \mathcal{F}_{\mathbf{b}}$, and a choice of n points x_i, one can construct the data set $\mathcal{D} = \{(x_i, f(x_i))\}_{i=1\dots n}$ and the approximation scheme generates an approximating function $h(\mathcal{D})$. It should be clear that for an L_1 distance metric ($d(f,h) = \int_0^1 |f-h|dx$), the following is true:

$$\sup_{f \in \mathcal{F}_{\mathbf{b}}} d(f,h) = \frac{1}{2}\sum_{i=1}^{n+1} a_i b_i = \frac{1}{2}\mathbf{a}.\mathbf{b}$$

Thus, taking the supremum over the entire class of functions is equivalent to

$$\sup_{f \in \mathcal{F}} d(f, h(\mathcal{D})) = \sup_{\{\mathbf{b}: b_i \geq 0, \sum b_i = M\}} \frac{1}{2}\mathbf{a}.\mathbf{b}$$

The above is a straight forward linear programming problem and yields as its solution the result $\frac{1}{2}M \max\{a_i, i = 1, \ldots, (n+1)\}$.

Finally, every choice of n points $x_i, i = 1, \ldots, n$ results in a corresponding vector $\mathbf{a}$ where $a_i \geq 0$ and $\sum a_i = 1$. Thus minimizing the maximum error over all the choice of sample points (according to eq. 3.28) is equivalent to

$$\arg \min_{x_1, \ldots, x_n} \sup_{f \in \mathcal{F}} d(f, h(\mathcal{D} = \{(x_i, f(x_i))\}_{i=1 \ldots n}) = \arg \min_{\{\mathbf{a}: a_i \geq 0, \sum a_i = 1\}} \max\{a_i; i = 1 \ldots n+1\}$$

Clearly the solution of the above problem is $a_i = \frac{1}{n+1}$ for each i.

In other words, classical optimal recovery suggests that one should sample the function uniformly. Note that this is not an adaptive scheme. In the next section, we compare empirically the performance of three different schemes to sample. The passive, where one samples randomly, the non-sequential "optimal", where one samples uniformly, and the active which follows our sequentially optimal strategy.

Error Rates and Sample Complexities for some Arbitrary Functions: Some Simulations

In this section, we attempt to relate the number of examples drawn and error made by the learner for a variety of arbitrary monotone increasing functions. We begin with the following simulation:

Simulation A:

1. Pick an arbitrary monotone-increasing function.

2. Decide (N), the number of samples to be collected. There are three methods of collection of samples. The first is by randomly drawing N examples according to a uniform distribution on $[0, 1]$ (corresponding to the passive case). The second is by asking for function values on a uniform grid on $[0, 1]$ of grid spacing $1/N$. The third is the CLA.

3. The three learning algorithms differ only in their method of obtaining samples. Once the samples are obtained, all three algorithms attempt to approximate the target by the monotone function which is the linear interpolant of the samples.

4. This entire process is now repeated for various values of N for the same target function and then repeated again for different target functions.

Results: Let us first consider performance on the arbitrarily selected monotonic function of the earlier section. Shown in fig. 3-20 are performance for the three

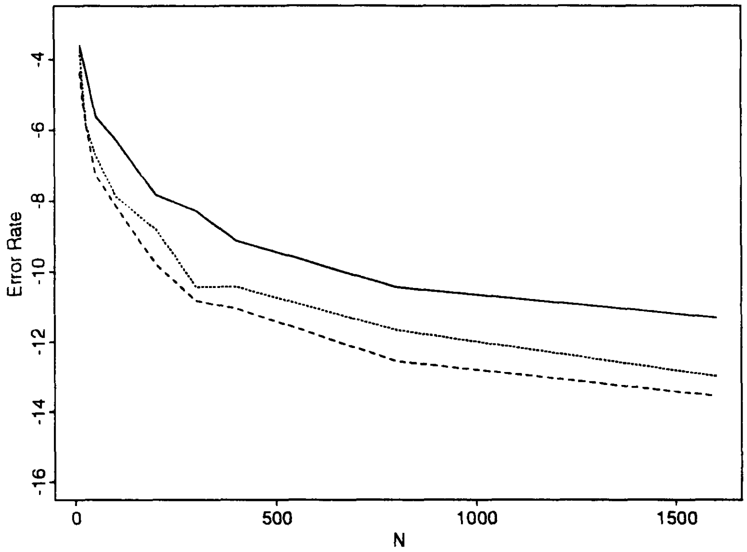

Figure 3-20: Error rates as a function of the number of examples for the arbitrary monotone function shown in a previous figure.

different algorithms. Notice that the active learning strategy (CLA) has the lowest error rate. On an average, the error rate of random sampling is 8 times the rate of CLA and uniform sampling is 1.5 times the rate of CLA.

Figure 3-21 shows four other monotonic functions on which we ran the same simulations comparing the three sampling strategies. The results of the simulations are shown in Fig. 3-22 and Table 3.2.3. Notice that the active strategy (CLA) far outperforms the passive strategy and clearly has the best error performance. The comparison between uniform sampling and active sampling is more interesting. For functions like function-2 (which is a smooth approximation of a step function), where most of the "information" is located in a small region of the domain, CLA outperforms the uniform learner by a large amount. Functions like function-3 which don't have any clearly identified region of greater information have the least difference between CLA and the uniform learner (as also between the passive and active learner). Finally on functions which lie in between these two extremes (like functions 4 and 5) we see decreased error-rates due to CLA which are in between the two extremes.

In conclusion, the active learner outperforms the passive learner. Further, it is even better than classical optimal recovery. The significant advantage of the active learner lies in its adaptive nature. Thus, for certain "easy" functions, it might converge very rapidly. For others, it might take as long as classical optimal recovery,

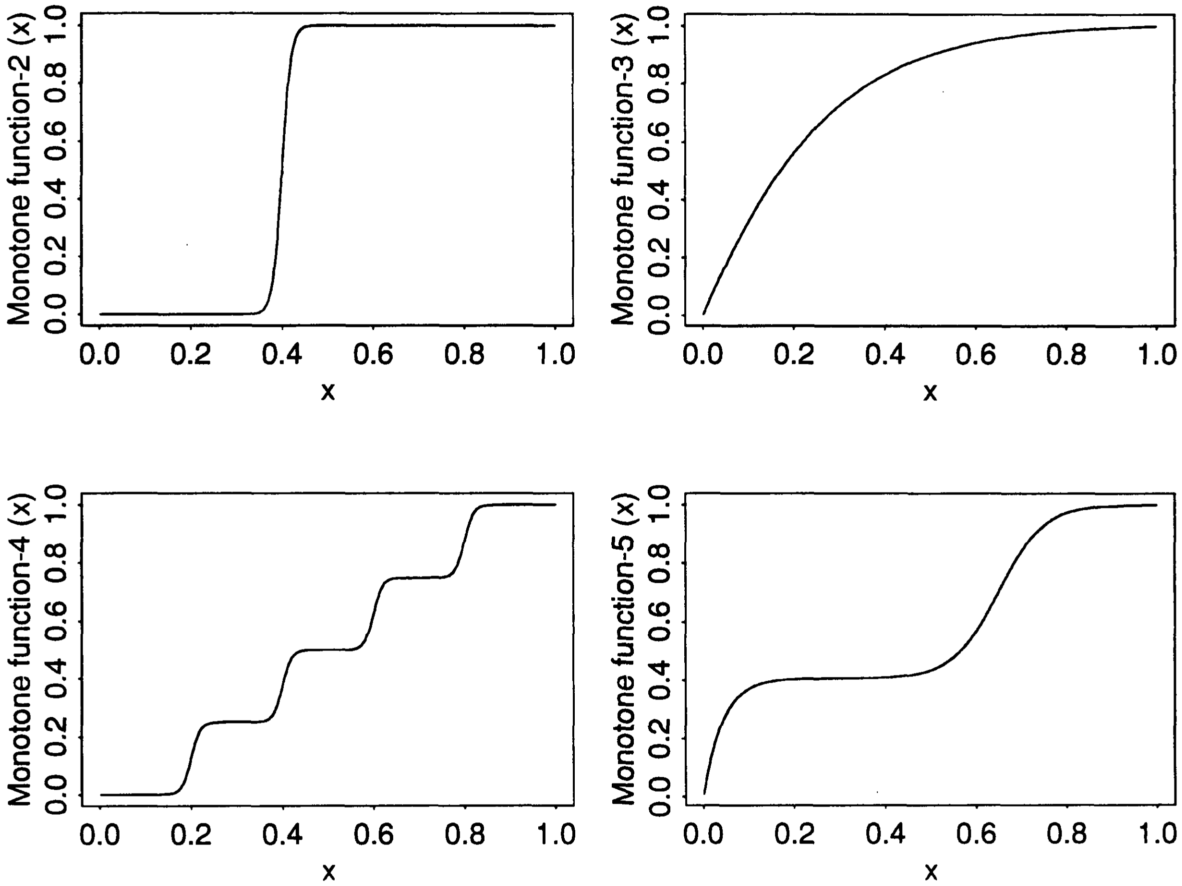

Figure 3-21: Four other monotonic functions on which simulations have been run comparing random, uniform, and active sampling strategies.

Function No.	Average Random/CLA	Average Uniform/CLA
1	7.23	1.66
2	61.37	10.91
3	6.67	1.10
4	8.07	1.62
5	6.62	1.56

Table 3.1: Shown in this table is the average error rate of the random sampling and the uniform sampling strategies when as a multiple of the error rates due to CLA. Thus for the function 3 for example, uniform error rates are on an average 1.1 times CLA error rates. The averages are taken over the different values of N (number of examples) for which the simulations have been done. Note that this is not a very meaningful average as the difference in the error rates between the various strategies grow with N (as can be seen from the curves)if there is a difference in the order of the sample complexity. However they have been provided just to give a feel for the numbers.

though never more.

3.3 Example 2: A Class of Functions with Bounded First Derivative

Here the class of functions we consider are from $[0,1]$ to R and of the form

$$\mathcal{F} = \{f|f(x) \text{ is differentiable and } |\frac{df}{dx}| \leq d\}$$

Notice a few things about this class. First, there is no direct bound on the values that functions in $\mathcal{F}$ can take. In other words, for every $M > 0$, there exists some function $f \in \mathcal{F}$ such that $f(x) > M$ for some $x \in [0,1]$. However, there is a bound on the first derivative, which means that a particular function belonging to $\mathcal{F}$ cannot itself change very sharply. Knowing the value of the function at any point, we can bound the value of the function at all other points. So for example, for every $f \in \mathcal{F}$, we see that $|f(x)| \leq dxf(0) \leq df(0)$.

We observe that this class too has infinite pseudo-dimension. We state this without proof.

Observation 3 *The class $\mathcal{F}$ has infinite pseudo-dimension in the sense of Pollard.*

As in the previous example we would like to investigate the possibility of devising active learning strategies for this class. We first provide a lower bound on the number of examples a learner (whether passive or active) would need in order to ϵ identify this

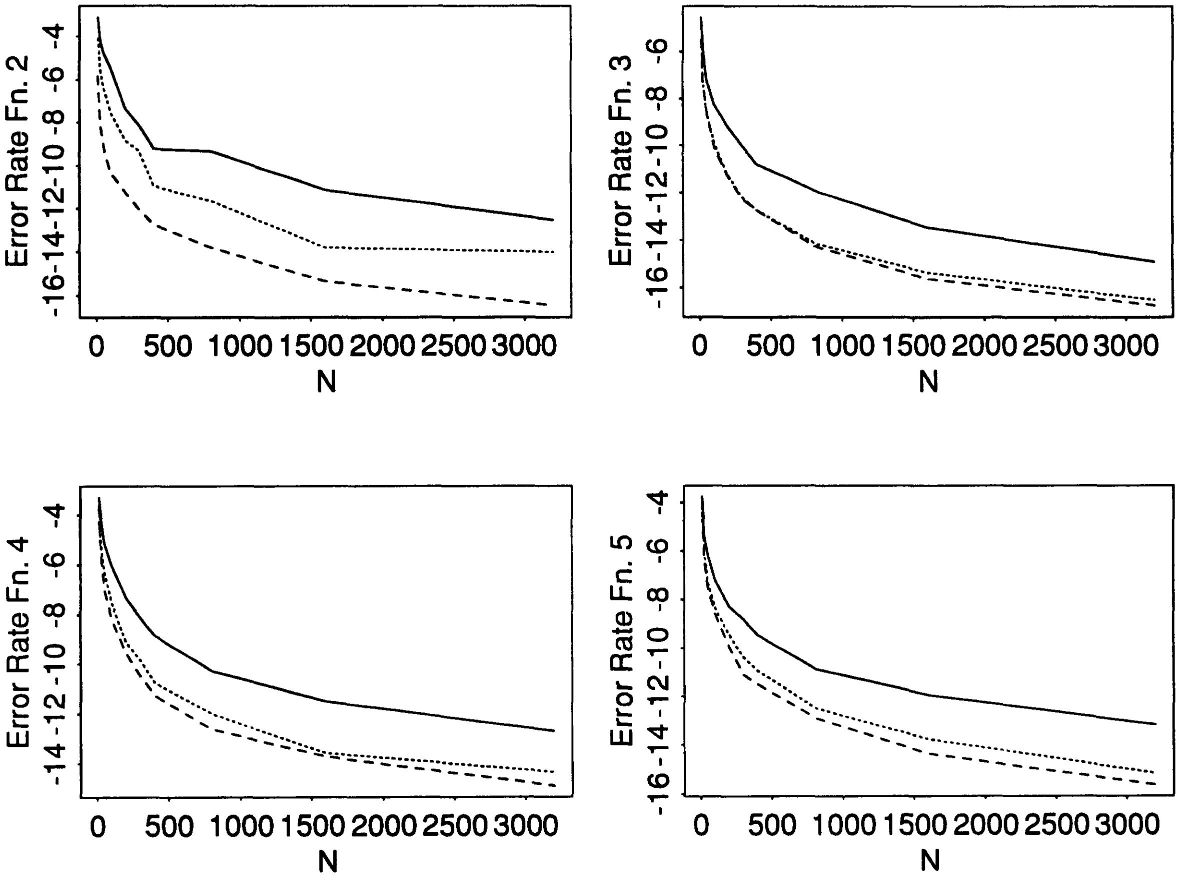

Figure 3-22: This figure plots the log of the error (L_1 error) against N the number of examples for each of the 4 monotonic functions shown in fig. 3-21. The solid line represents error rates for random sampling, the line with small dashes represents uniform sampling and the line with long dashes represents results for CLA. Notice how CLA beats random sampling by large amounts and does slightly better than uniform sampling.

class. We then derive in the next section, an optimal active learning strategy (that is, an instantiation of the Active Algorithm B earlier). We also provide an upper bound on the number of examples this active algorithm would take.

We also need to specify some other terms for this class of functions. The approximation scheme $\mathcal{H}$ is a first order spline as before, the domain $D = [0,1]$ is partitioned into intervals by the data $[x_i, x_{i+1}]$ (again as before) and the metric d_C is an L_1 metric given by $d_C(f_1, f_2) = \int_C |f_1(x) - f_2(x)|dx$. The results in this section can be extended to an L_p norm but we confine ourselves to an L_1 metric for simplicity of presentation.

3.3.1 Lower Bounds

Theorem 3.3.1 *Any learning algorithm (whether passive or active) has to draw at least $\Omega((d/\epsilon))$ examples (whether randomly or by choosing) in order to PAC learn the class $\mathcal{F}$.*

Proof Sketch: Let us assume that the learner collects m examples (passively by drawing according to some distribution, or actively by any other means). Now we show that an adversary can force the learner to make an error of atleast ϵ if it draws less than $\Omega((d/\epsilon))$ examples. This is how the adversary functions.

At each of the m points which are collected by the learner, the adversary claims the function has value 0. Thus the learner is reduced to coming up with a hypothesis that belongs to $\mathcal{F}$ and which it claims will be within an ϵ of the target function. Now we need to show that whatever the function hypothesized by the learner, the adversary can always come up with some other function, also belonging to $\mathcal{F}$, and agreeing with all the data points, which is more than an ϵ distance away from the learner's hypothesis. In this way, the learner will be forced to make an error greater than ϵ.

The m points drawn by the learner, divides the region $[0,1]$ into (at most) $m+1$ different intervals. Let the length of these intervals be $b_1, b_2, b_3, ..., b_{m+1}$. The "true" function, or in other words, the function the adversary will present, should have value 0 at the endpoints of each of the above intervals. We first state the following lemma.

Lemma 3.3.1 *There exists a function $f \in \mathcal{F}$ such that f interpolates the data and*

$$\int_{[0,1]} |f|dx > \frac{kd}{4(m+1)}$$

where k is a constant arbitrarily close to 1.

Proof: Consider fig. 3-23. The function f is indicated by the dark line. As is shown, f changes sign at each $x = x_i$. Without loss of generality, we consider an interval

$[x_i, x_{i+1}]$ of length b_i. Let the midpoint of this interval be $z = (x_i + x_{i+1})/2$. The function here has the values

$$f(x) = \begin{cases} d(x - x_i) & \text{for } x \in [x_i, z - \alpha] \\ -d(x - x_{i+1}) & \text{for } x \in [z + \alpha, x_{i+1}] \\ \frac{d(x-z)^2}{2\alpha} + \frac{d(b_i-\alpha)}{2} & \text{for } x \in [z - \alpha, z + \alpha] \end{cases}$$

Simple algebra shows that

$$\int_{x_i}^{x_{i+1}} |f| dx > d(\frac{b_i - \alpha}{2})^2 + \alpha d(\frac{b_i - \alpha}{2}) = d(b_i^2 - \alpha^2)/4$$

Clearly, α can be chosen small, so that

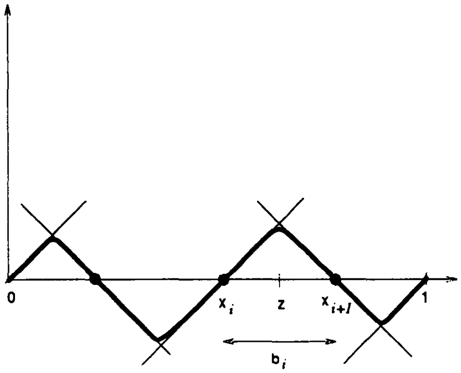

Figure 3-23: Construction of a function satisying Lemma 2.

$$\int_{x_i}^{x_{i+1}} |f| dx > \frac{kdb_i{}^2}{4}$$

where k is as close to 1 as we want. By combining the different pieces of the function we see that

$$\int_0^1 |f| dx > \frac{kd}{4} \sum_i^{m+1} b_i^2$$

Now we make use of the following lemma,

Lemma 3.3.2 *For a set of numbers $b_1, .., b_m$ such that $b_1 + b_2 + .. + b_m = 1$, the following inequality is true*

$$b_1^2 + b_2^2 + .. + b_m^2 \geq 1/m$$

Proof: By induction. □

Now it is easy to see how the adversary functions. Suppose the learner postulates that the true function is h. Let $\int_{[0,1]} |h|dx = \chi$. If $\chi > \epsilon$, the adversary claims that the true function was $f = 0$. In that case $\int_0^1 |h - f|dx = \chi > \epsilon$. If on the other hand, $\chi < \epsilon$, then the adversary claims that the true function was f (as above). In that case,

$$\int_0^1 |f - h|dx \geq \int_0^1 |f|dx - \int_0^1 |h|dx = \frac{kd}{4(m+1)} - \chi$$

Clearly, if m is less than $\frac{kd}{8\epsilon} - 1$, the learner is forced again to make an error greater than ϵ. Thus in either case, the learner is forced to make an error greater than or equal to ϵ if less than $\Omega(d/\epsilon)$ examples are collected (howsoever these examples are collected). □

The previous result holds for all learning algorithms. It is possible to show the following result for a passive learner.

Theorem 3.3.2 *A Passive learner must draw at least* $\max(\Omega((d/\epsilon), \sqrt{(d/\epsilon)}\ln(1/\delta)))$ *to learn this class.*

Proof Sketch: The d/ϵ term in the lower bound follows directly from the previous theorem. We show how the second term is obtained.

Consider the uniform distribution on $[0, 1]$ and a subclass of functions which have value 0 on the region $\text{A} = [0, 1 - \alpha]$ and belong to $\mathcal{F}$. Suppose the passive learner draws l examples uniformly at random. Then with probability $(1 - \alpha)^l$, all these examples will be drawn from region A. It only remains to show that for this event, and the subclass considered, whatever be the function hypothesized by the learner, an adversary can force it to make a large error.

It is easy to show (using the arguments of the earlier theorem) that there exists a function $f \in \mathcal{F}$ such that f is 0 on A and $\int_{1-\alpha}^1 |f|dx = \frac{1}{2}\alpha^2 d$. This is equal to 2ϵ if $\alpha = \sqrt{(4\epsilon/d)}$. Now let the learner's hypothesis be h. Let $\int_{1-\alpha}^1 |h|dx = \chi$. If χ is greater than ϵ, the adversary claims the target was $g = 0$. Otherwise, the adversary claims the target was $g = f$. In either case, $\int |g - h|dx > \epsilon$.

It is possible to show (by an identical argument to the proof of theorem 1), that unless $l \geq \frac{1}{4}\sqrt{(d/\epsilon)}\ln(1/\delta)$, all examples will be drawn from A with probability greater than δ and the learner will be forced to make an error greater than ϵ. Thus the second term appears indicating the dependence on δ in the lower bound.□

3.3.2 Active Learning Algorithms

We now derive in this section an algorithm which actively selects new examples on the basis of information gathered from previous examples. This illustrates how our formulation of section 3.1.1 can be used in this case to effectively obtain an optimal adaptive sampling strategy.

Derivation of an optimal sampling strategy

Fig. 3-24 shows an arbitrary data set containing information about some unknown target function. Since the target is known to have a first derivative bounded by d, it is clear that the target is constrained to lie within the parallelograms shown in the figure. The slopes of the lines making up the parallelogram are d and $-d$ appropriately. Thus, $\mathcal{F}_\mathcal{D}$ consists of all functions which lie within the parallelograms and interpolate the data set. We can now compute the uncertainty of the approximation scheme over any interval,C, (given by $e_C(\mathcal{H},\mathcal{D},\mathcal{F})$), for this case. Recall that the approximation scheme $\mathcal{H}$ is a first order spline, and the data $\mathcal{D}$ consists of (x,y) pairs. Fig. 3-25 shows the situation for a particular interval ($C_i = [x_i, x_{i+1}]$). Here i ranges from 0 to n. As in the previous example, we let $x_0 = 0$, and $x_{n+1} = 1$.

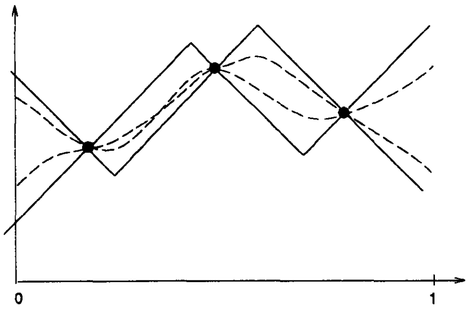

Figure 3-24: An arbitrary data set for the case of functions with a bounded derivative. The functions in $\mathcal{F}_\mathcal{D}$ are constrained to lie in the parallelograms as shown. The slopes of the lines making up the parallelogram are d and $-d$ appropriately.

The maximum error the approximation scheme $\mathcal{H}$ could have on this interval is given by (half the area of the parallelogram).

$$e_{C_i}(\mathcal{H},\mathcal{D},\mathcal{F}) = \sup_{f\in\mathcal{F}_\mathcal{D}} \int_{C_i} |h-f|dx = \frac{(d^2B_i^2 - A_i^2)}{4d}$$

where $A_i = |f(x_{i+1}) - f(x_i)|$ and $B_i = x_{i+1} - x_i$. Clearly, the maximum error the approximation scheme could have over the entire domain is given by

$$e_{D=[0,1]}(\mathcal{H},\mathcal{D},\mathcal{F}) = \sup_{f\in\mathcal{F}_\mathcal{D}} \sum_{j=0}^{n} \int_{C_j} |f-h|dx = \sum_{j=0}^{n} e_{C_j} \qquad (3.31)$$

The computation of e_C is crucial to the derivation of the active sampling strategy. Now imagine that we chose to sample at a point x in the interval C_i and received a value y (belonging to $\mathcal{F}_{\mathcal{D}}(x)$). This adds one more interval and divides C_i into two intervals C_{i1} and C_{i2} as shown in fig. 3-26.. We also obtain two correspondingly smaller parallelograms within which the target function is now constrained to lie.

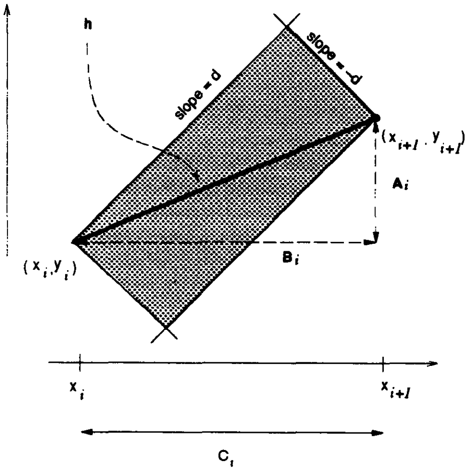

Figure 3-25: A zoomed version of the ith interval.

The addition of this new data point to the data set ($\mathcal{D}' = \mathcal{D} \cup (x, y)$) requires us to recompute the learner's hypothesis (denoted by h' in the fig. 3-26). Correspondingly, it also requires us to update e_C, i.e., we now need to compute $e_C(\mathcal{H}, \mathcal{D}', \mathcal{F})$. First we observe that the addition of the new data point does not affect the uncertainty measure on any interval other than the divided interval C_i. This is clear when we notice that the parallelograms (whose area denotes the uncertainty on each interval) for all the other intervals are unaffected by the new data point.

Thus,

$$e_{C_j}(\mathcal{H}, \mathcal{D}', \mathcal{F}) = e_{C_j}(\mathcal{H}, \mathcal{D}, \mathcal{F}) = \frac{1}{4d}(d^2 B_j^2 - A_j^2) \text{ for } j \neq i$$

For the ith interval C_i, the total uncertainty is now recomputed as (half the sum of the two parallelograms in fig. 3-26)

$$\begin{aligned} e_{C_i}(\mathcal{H}, \mathcal{D}', \mathcal{F}) &= \tfrac{1}{4d}\left((d^2u^2 - v^2) + (d^2(B_i - u)^2 - (A_i - v)^2)\right) \\ &= \tfrac{1}{4d}\left((d^2u^2 + d^2(B_i - u)^2) - (v^2 + (A - v)^2)\right) \end{aligned} \tag{3.32}$$

where $u = x - x_i$, $v = y - y_i$, and A_i and B_i are as before. Note that u ranges from 0 to B_i, for $x_i \leq x \leq x_{i+1}$. However, given a particular choice of x (this fixes a value of u), the possible values v can take are constrained by the geometry of the parallelogram. In particular, v can only lie within the parallelogram. For a particular x, we know that $\mathcal{F}_{\mathcal{D}}(x)$ represents the set of all possible y values we can receive. Since $v = y - y_i$, it is clear that $v \in \mathcal{F}_{\mathcal{D}}(x) - y_i$. Naturally, if $y < y_i$, we find that $v < 0$, and $A_i - v > A_i$. Similarly, if $y > y_{i+1}$, we find that $v > A_i$.

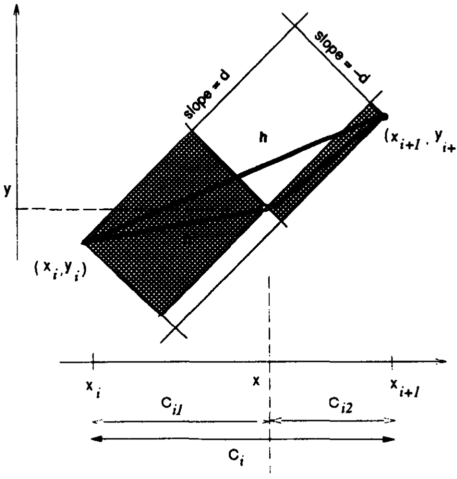

Figure 3-26: Subdivision of the ith interval when a new data point is obtained.

We now prove the following lemma:

Lemma 3.3.3 *The following two identities are valid for the appropriate mini-max problem.*

(1)$\frac{B}{2} = \arg\min_{u\in[0,B]} \sup_{v\in\{\mathcal{F}_{\mathcal{D}}(x)-y_i\}} ((d^2u^2 + d^2(B-u)^2) - (v^2 + (A-v)^2))$

(2) $\frac{1}{2}(d^2B^2 - A^2) = \min_{u\in[0,B]} \sup_{v\in\{\mathcal{F}_{\mathcal{D}}(x)-y_i\}} ((d^2u^2 + d^2(B-u)^2) - (v^2 + (A-v)^2))$

Proof: The expression on the right is a difference of two quadratic expressions and can be expressed as $q_1(u) - q_2(v)$. For a particular u, the expression is maximized when the quadratic $q_2(v) = (v^2 + (A-v)^2)$ is minimized. Observe that this quadratic is globally minimized at $v = A/2$. We need to perform this minimization over the set $v \in \mathcal{F}_{\mathcal{D}}(x) - y_i$ (this is the set of values which lie within the upper and lower boundaries of the parallelogram shown in fig. 3-27). There are three cases to consider.

Case I: $u \in [A/2d, B - A/2d]$

First, notice that for u in this range, it is easy to verify that the upper boundary of the parallelogram is greater than $A/2$ while the lower boundary is less than $A/2$. Thus we can find a value of v (viz. v = A/2) which globally minimizes this quadratic because $A/2 \in \mathcal{F}_{\mathcal{D}}(x) - y_i$. The expression thus reduces to $d^2u^2 + d^2(B-u)^2 - A^2/2$.

Over the interval for u considered in this case, it is minimized at $u = B/2$ resulting in the value

$$(d^2B^2 - A^2)/2$$

Case II: $u \in [0, A/2d]$

In this case, the upper boundary of the parallelogram (which is the maximum value v can take) is less than $A/2$ and hence the $q_2(v)$ is minimized when $v = du$. The total expression then reduces to

$$d^2u^2+d^2(B-u)^2-((du)^2+(A-du)^2) = d^2(B-u)^2-(A-du)^2 = (d^2B^2-A^2)-2ud(dB-A)$$

Since, $dB > A$, the above is minimized on this interval by choosing $u = A/2d$ resulting in the value

$$dB(dB - A)$$

Case III: By symmetry, this reduces to case II.

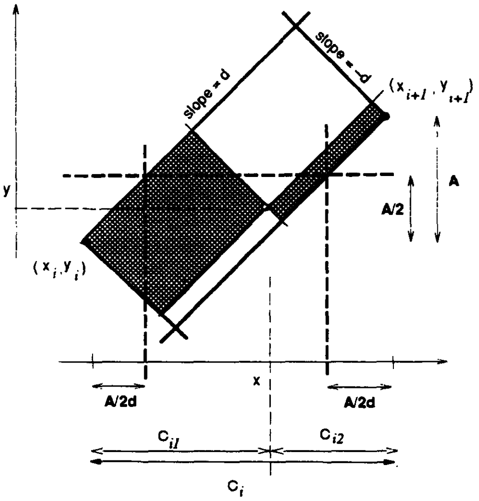

Figure 3-27: A figure to help the visualization of Lemma 4. For the x shown, the set $\mathcal{F}_\mathcal{D}$ is the set of all values which lie within the parallelogram corresponding to this x, i.e., on the vertical line drawn at x but within the parallelogram.

Since $(d^2B^2 - A^2)/2 \leq dB(dB - A)$ (this is easily seen by completing squares), it follows that $u = B/2$ is the global solution of the mini-max problem above. Further, we have shown that for this value of u, the sup term reduces to $(d^2B^2 - A^2)/2$ and the lemma is proved.□

Using the above lemma along with eq. 3.32, we see that

$$\min_{x\in C_i}\sup_{y\in\mathcal{F}_{\mathcal{D}}(x)} e_{C_i}(\mathcal{H},\mathcal{D}\cup(x,y),\mathcal{F}) = \frac{1}{8d}(d^2B_i^2 - A_i^2) = \frac{1}{2}e_{C_i}(\mathcal{H},\mathcal{D},\mathcal{F})$$

In other words, by sampling the midpoint of the interval C_i, we are guaranteed to reduce the uncertainty by 1/2. As in the case of monotonic functions now, we see that using eq. 3.31, we should sample the midpoint of the interval with largest uncertainty $e_{C_i}(\mathcal{H},\mathcal{D},\mathcal{F})$ to obtain the global solution in accordance with the principle of Algorithm B of section 3.1.

This allows us to formally state an active learning algorithm which is optimal in the sense implied in our formulation.

The Choose and Learn Algorithm - 2 (CLA-2)

1. **[Initial Step]** Ask for values of the function at points $x = 0$ and $x = 1$. At this stage, the domain $D = [0,1]$ is composed of one interval only, viz., $C_1 = [0,1]$. Compute $e_{C_1} = \frac{1}{4d}(d^2 - |f(1) - f(0)|^2)$ and $e_D = e_{C_1}$. If $e_D < \epsilon$, stop and output the linear interpolant of the samples as the hypothesis, otherwise query the midpoint of the interval to get a partition of the domain into two subintervals $[0,1/2)$ and $[1/2,1]$.

2. **[General Update and Stopping Rule]** In general, at the kth stage, suppose that our partition of the interval $[0,1]$ is $[x_0 = 0, x_1)$,$[x_1,x_2)$, $\ldots$, $[x_{k-1}, x_k = 1]$. We compute the uncertainty $e_{C_i} = \frac{1}{4d}\left(d^2(x_i - x_{i-1})^2 - |y_i - y_{i-1}|^2\right)$ for each $i = 1,\ldots,k$. The midpoint of the interval with maximum e_{C_i} is queried for the next sample. The total error $e_D = \sum_{i=1}^{k} e_{C_i}$ is computed at each stage and the process is terminated when $e_D \leq \epsilon$. Our hypothesis h at every stage is a linear interpolation of all the points sampled so far and our final hypothesis is obtained upon the termination of the whole process.

It is possible to show that the following upperbound exists on the number of examples CLA would take to learn the class of functions in consideration

Theorem 3.3.3 *The CLA-2 would PAC learn the class in at most* $\frac{d}{4\epsilon} + 1$ *examples.*

Proof Sketch: Following a strategy similar to the proof of Theorem 3, we show how a slight variant of CLA-2 would converge in at most $(d/4\epsilon + 1)$ examples. Imagine a grid of n points placed $1/(n-1)$ apart on the domain $D = [0,1]$ where the kth point is $k/(n-1)$ (for k going from 0 to $n-1$). The variant of the CLA-2 operates by confining its queries to points on this grid. Thus at the kth stage, instead of querying the midpoint of the interval with maximum uncertainty, it will query the gridpoint closest to this midpoint. Suppose it uses up all the gridpoints in this fashion, then there will be $n-1$ intervals and by our arguments above, we have seen that the maximum error on each interval is bounded by

$$\frac{1}{4d}(d^2(\frac{1}{n-1})^2 - |y_i - y_{i-1}|^2) \leq \frac{1}{4d}d^2(\frac{1}{n-1})^2$$

Since there are $n-1$ such intervals, the total error it could make is bounded by

$$(n-1)\frac{1}{4d}d^2(\frac{1}{n-1})^2 = \frac{1}{4d}(\frac{1}{n-1})$$

It is easy to show that for $n > d/4\epsilon + 1$, this maximum error is less than ϵ. Thus the learner need not collect any more than $d/4\epsilon + 1$ examples to learn the target function to within an ϵ accuracy. Note that the learner will have identified the target to ϵ accuracy with probability 1 (always) by following the strategy outlined in this variant of CLA-2. □

We now have both an upper and lower bound for PAC-learning the class (under a uniform distribution) with queries. Notice that here as well, the sample complexity of active learning does not depend upon the confidence parameter δ. Thus for δ arbitrarily small, the difference in sample complexities between passive and active learning becomes arbitrarily large with active learning requiring much fewer examples.

3.3.3 Some Simulations

We now provide some simulations conducted on arbitrary functions of the class of functions with bounded derivative (the class $\mathcal{F}$). Fig. 3-28 shows 4 arbitrary selected functions which were chosen to be the target function for the approximation scheme considered. In particular, we are interested in observing how the active strategy samples the target function for each case. Further, we are interested in comparing the active and passive techniques with respect to error rates for the same number of examples drawn. In this case, we have been unable to derive an analytical solution to the classical optimal recovery problem. Hence, we do not compare it as an alternative sampling strategy in our simulations.

Distribution of points selected

The active algorithm CLA-2 selects points adaptively on the basis of previous examples received. Thus the distribution of the sample points in the domain D of the function depends inherently upon the arbitrary target function. Consider for example, the distribution of points when the target function is chosen to be Function-1 of

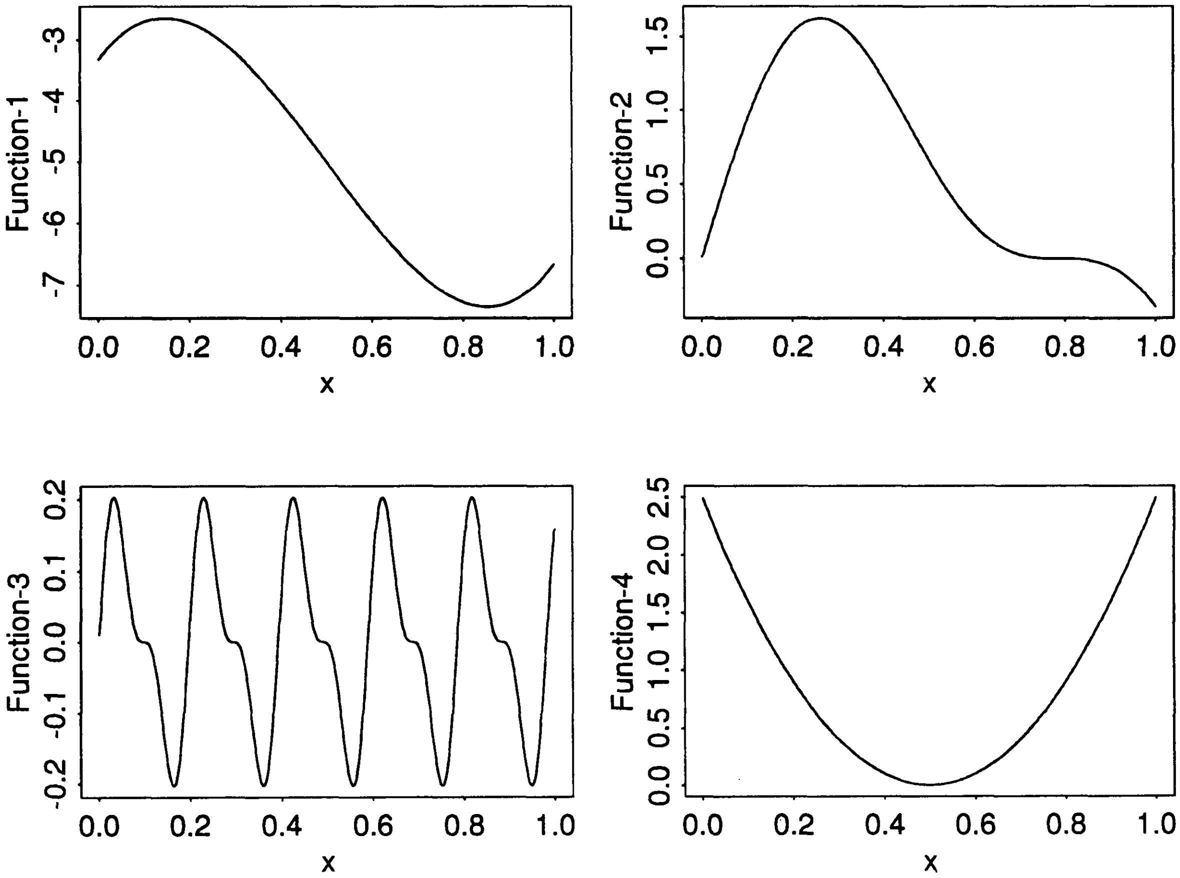

Figure 3-28: Four functions with bounded derivative considered in the simulations. The uniform bound on the derivative was chosen to be $d = 10$.

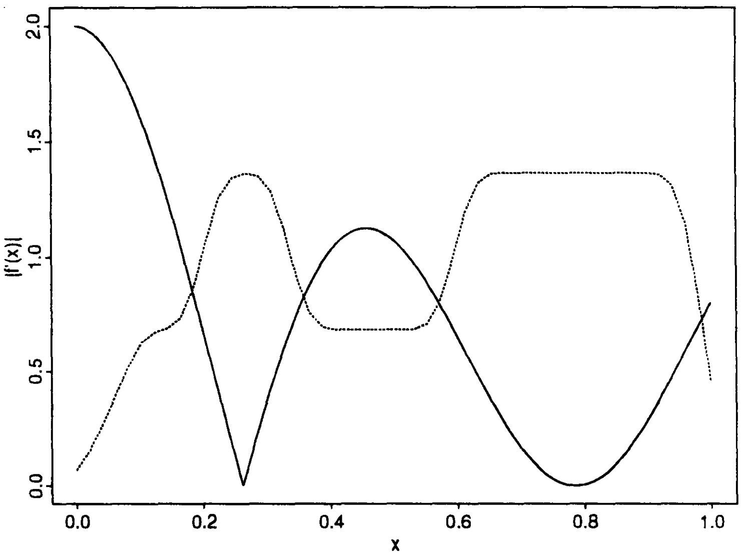

Figure 3-30: How CLA-2 chooses to sample its points. The solid line is a plot of $|f'(x)|$ where f is Function-1 of our simulation set. The dotted line shows the density of sample points (queried by CLA-2) on the domain.

Error Rates:

In an attempt to relate the number of examples drawn and the error made by the learner, we performed the following simulation.

Simulation B:

1. Pick an arbitrary function from class $\mathcal{F}$.

2. Decide N, the number of samples to be collected. There are two methods of collection of samples. The first (*passive*) is by randomly drawing N examples according to a uniform distribution on $[0,1]$. The second (*active*) is the CLA-2.

3. The two learning algorithms differ only in their method of obtaining samples. Once the samples are obtained, both algorithms attempt to approximate the target by the linear interpolant of the samples (first order splines).

4. This entire process is now repeated for various values of N for the same target function and then repeated again for the four different target functions of fig. 3-28

The results are shown in fig. 3-31. Notice how the active learner outperforms the

the set shown in fig. 3-28.

Notice (as shown in fig. 3-29) that the algorithm chooses to sample densely in places where the target is flat, and less densely where the function has a steep slope. As our mathematical analysis of the earlier section showed, this is well founded. Roughly speaking, if the function has the same value at x_i and x_{i+1}, then it could have a variety of values (wiggle a lot) within. However, if, $f(x_{i+1})$ is much greater (or less) than $f(x_i)$, then, in view of the bound, d, on how fast it can change, it would have had to increase (or decrease) steadily over the interval. In the second case, the rate of change of the function over the interval is high, there is less uncertainty in the values of the function within the interval, and consequently fewer samples are needed in between.

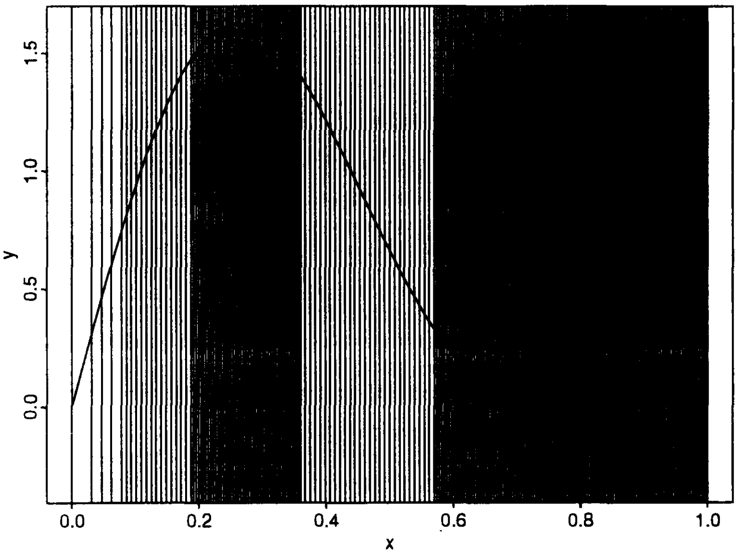

Figure 3-29: How CLA-2 chooses to sample its points. Vertical lines have been drawn at the x values where the CLA queried the oracle for the corresponding function value.

In example 1, for the case of monotone functions, we saw that the density of sample points was proportional to the first derivative of the target function. By contrast, in this example, the optimal strategy chooses to sample points in a way which is inversely proportional to the magnitude of the first derivative of the target function. Fig. 3-30 exemplifies this.

passive learner. For the same number of examples, the active scheme having chosen its examples optimally by our algorithm makes less error.

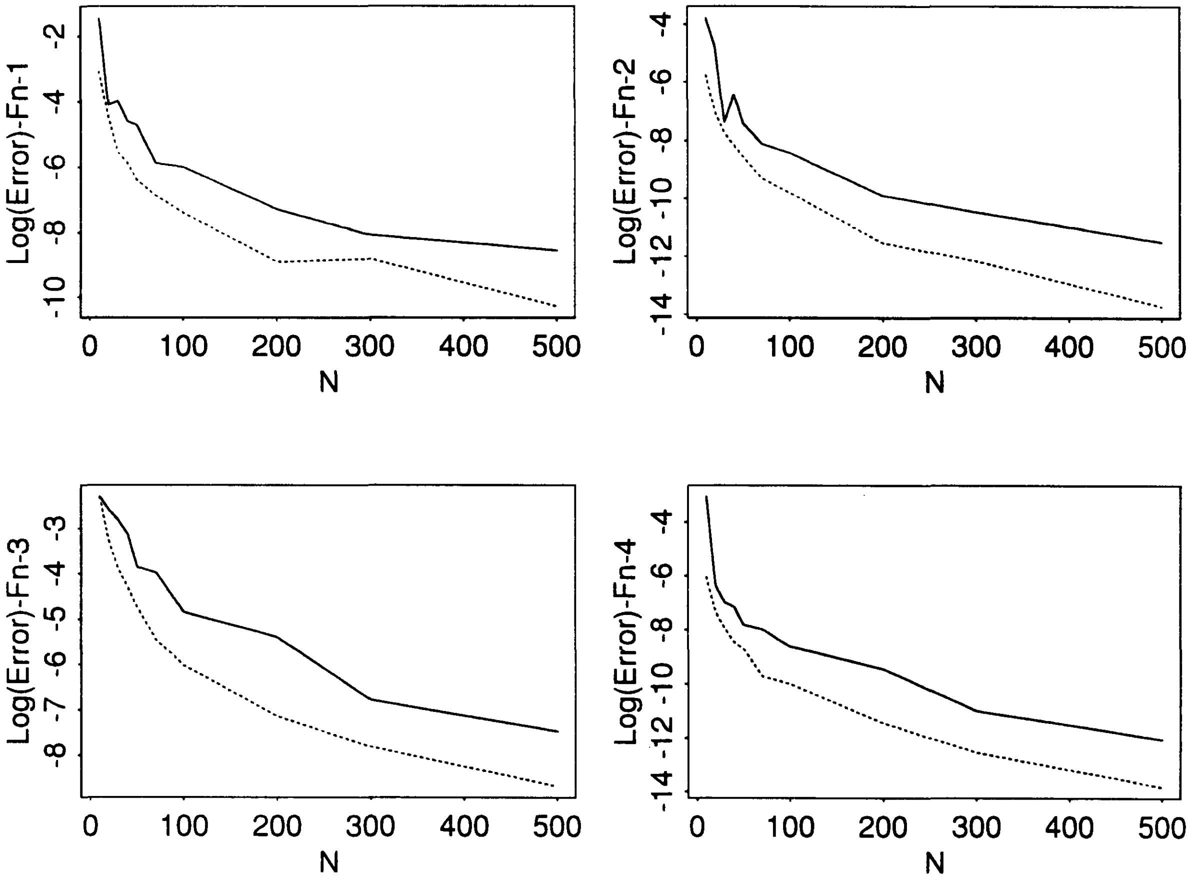

Figure 3-31: Results of Simulation B. Notice how the sampling strategy of the active learner causes better approximation (lower rates) for the same number of examples.

We have obtained in theorem 6, an upper bound on the performance of the active learner. However, as we have already remarked earlier, the number of examples the active algorithm takes before stopping (i.e., outputting an ϵ-good approximation) varies and depends upon the nature of the target function. "Simple" functions are learned quickly, "difficult" functions are learned slowly. As a point of interest, we have shown in fig. 3-32, how the actual number of examples drawn varies with ϵ. In order to learn a target function to ϵ-accuracy, CLA-2 needs at most $n_{\max}(\epsilon) = d/4\epsilon+1$ examples. However, for a particular target function, f, let the number of examples it actually requires be $n_f(\epsilon)$. We plot $\frac{n_f(\epsilon)}{n_{\max}(\epsilon)}$ as a function of ϵ. Notice, first, that this ratio is always much less than 1. In other words, the active learner stops before the worst case upper bound with a guaranteed ϵ-good hypothesis. This is the significant advantage of an adaptive sampling scheme. Recall that for uniform sampling (or classical optimal recovery even) we would have no choice but to ask for $d/4\epsilon$ examples to be sure of having an ϵ-good hypothesis. Further, notice that that as ϵ gets smaller, the ratio gets smaller. This suggests that for these functions, the sample complexity of the active learner is of a different order (smaller) than the worst case bound. Of course, there always exists some function in $\mathcal{F}$ which would force the active learner to perform at its worst case sample complexity level.

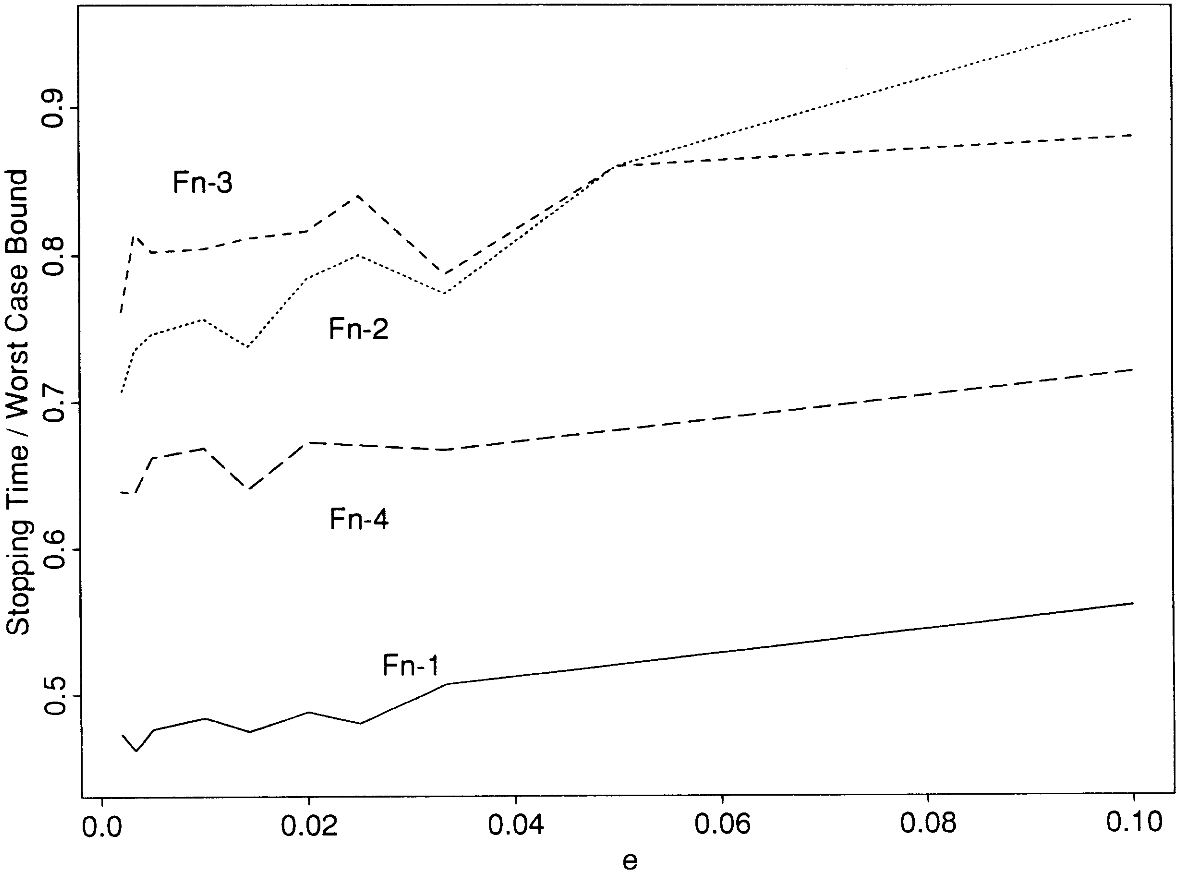

Figure 3-32: Variation with epsilons.

3.4 Conclusions, Extensions, and Open Problems

This part of the chapter focused on the possibility of devising *active* strategies to collect data for the problem of approximating real-valued function classes. We were able to derive a sequential version of optimal recovery. This sequential version, by virtue of using partial information about the target function is superior to classical optimal recovery. This provided us with a general formulation of an adaptive sampling strategy, which we then demonstrated on two example cases. Theoretical and empirical bounds on the sample complexity of passive and active learning for these cases suggest the superiority of the active scheme as far as the number of examples needed is concerned. It is worthwhile to observe that the same general framework gave rise to completely different sampling schemes in the two examples we considered. In one, the learner sampled densely in regions of high change. In the other, the learner did the precise reverse. This should lead us to further appreciate the fact that active sampling strategies are very task-dependent.

Using the same general formulation, we were also able to devise active strategies (again with superior sample complexity gain) for the following concept classes. 1) For the class of indicator functions $\{1_{[a,b]} : 0 < a < b < 1\}$ on the interval $[0, 1]$, the sample complexity is reduced from $1/\epsilon \ln(1/\delta)$ for passive learning to $\ln(1/\epsilon)$ by adding membership queries. 2) For the class of half-spaces on a regular n-simplex, the sample complexity is reduced from $n/\epsilon \ln(1/\delta)$ to $n^2 \ln(s/\epsilon)$ by adding membership queries. Note that similar gains have been obtained for this class by Eisenberg (1992) using a different framework.

There are several directions for further research. First, one could consider the possibility of adding noise to our formulation of the problem. Noisy versions of optimal recovery exist and this might not be conceptually a very difficult problem. Although the general formulation (at least in the noise-free case) is complete, it might not be possible to compute the uncertainty bounds e_C for a variety of function classes. Without this, one could not actually use this paradigm to obtain a specific algorithm. A natural direction to pursue would be to investigate other classes (especially in more dimensions than 1) and other distance metrics to obtain further specific results. We observed that the active learning algorithm lay between classical optimal recovery and the optimal teacher. It would be interesting to compare the exact differences in a more principled way. In particular, an interesting open question is whether the sampling strategy of the active learner converges to that of the optimal teacher as more and more information becomes available. It would not be unreasonable to expect this, though precise results are lacking. In general, on the theme of better characterizing the conditions under which active learning would vastly outperform passive learning for function approximation, much work remains to be done. While active learning might require fewer examples to learn the target function, its computational burden is significantly larger. It is necessary to explore the information/computation trade-off with active learning schemes. Finally, we should note, that we have adopted in this part, a model of learning motivated by PAC but with a crucial difference. The distance metric, d, is not necessarily related to the distribution according to which data is drawn (in the passive case). This prevents us from using traditional uniform convergence (Vapnik, 1982) type arguments to prove learnability. The problem of learning under a different metric is an interesting one and merits further investigation in its own right.

Part II: Epsilon Focusing: A Strategy for Active Learning

In Part I, we discussed a principled strategy by means of which an active learner could choose its own examples, thereby potentially reducing the informational complexity of learning real-valued functions. The formalization adopted ideas from optimal recovery, and active learning reduced to a sequential version of the optimal recovery problem. In this part of the chapter, we discuss another possible scheme for choosing examples.

Recall that according to the PAC criterion for learning, we need to learn the target function to ϵ accuracy (according to some distance metric d on the space of functions, $\mathcal{F}$), with confidence greater than $1 - \delta$. Sometimes, knowledge that the function lies within some ϵ-ball (in function space) might directly translate (due to locality properties) into knowledge about the regions of the domain X over which the target function values are uncertain. The learner can then zoom (*epsilon-focus*) in on this region of uncertainty, and sample there. As a motivating real, world example, one could imagine that in a pattern classification task, the knowledge that the learner is within ϵ of the optimal discriminant boundary, might inform the learner about which regions of the feature space are worth sampling to a greater degree. Intuitively, one might think that regions close to the decision boundary are such worthwhile regions.

We formally illustrate this idea with a simple example in the next section. In all the cases we consider, the concept class (class of indicator functions) have bounded VC dimension. Consequently, they are learnable, and upper and lower bounds on the sample complexity of passive learning exist for these function classes. Roughly speaking, instead of learning to (ϵ, δ) accuracy at one shot by collecting the requisite number of examples, the learner attempts to obtain a loose estimate of the target.

Making use of locality properties, then, the learner obtains a loose estimate of the regions of the domain to sample more closely. On the basis of these fresh samples, the learner tightens its estimate of the target, thereby reducing the region of uncertainty. It then freshly samples this new, reduced, region of uncertainty and carries on in this fashion. The learner can arbitrarily reduce the sample complexity of learning by this scheme.

After our motivating example, we provide some generalizations, and finally end with some open questions.

3.5 A Simple Example

Suppose we want to PAC-learn (with (ϵ, δ) accuracy) the following class of indicator functions from $[0, 1]$ to $\{0, 1\}$.

$$\mathcal{F} = \{1_{[a,1]} : 0 \leq a1\}$$

Further suppose the distribution P on $[0, 1]$ according to which data is drawn is known and is uniform. It is known that a *passive* learner would take atleast $\Omega((1/\epsilon)\ln(1/\delta))$ examples to do so. We suggest the following k-step strategy which seeks examples from successively smaller well-focused regions of the domain to learn this class in $\Omega((k/\epsilon^{2k})\ln(k/\delta))$ examples.

The ϵ-focusing Algorithm (1)

The learning occurs over k (k can be arbitrarily chosen) stages.

1. Draw enough examples to learn the target with $\epsilon^{1/k}$ accuracy with δ/k confidence. Obtain hypothesis $1_{[}\hat{a_1}, 1]$.

2. Now ask for examples drawn uniformly at random from the region $[\hat{a_1} - \epsilon^{1/k}, \hat{a_1} + \epsilon^{1/k}]$ and try to learn the target function with $\epsilon^{1/k}/2$ accuracy with δ/k confidence (with respect to this new distribution over the smaller region). Obtain hypothesis $1_{[\hat{a_2},1]}$.

3. Repeat like step 2, i.e., ask for enough examples drawn uniformly at random from the region $[\hat{a_2} - \epsilon^{2/k}, \hat{a_2} + \epsilon^{2/k}]$ in order to learn the target function to $\epsilon^{1/k}/2$ accuracy with δ/k confidence. Obtain hypothesis $1_{[\hat{a_3},1]}$. In general at the jth step, ask for examples drawn uniformly at random from the region $[\hat{a_{j-1}} - \epsilon^{(j-1)/k}, \hat{a_{j-1}} + \epsilon^{(j-1)/k}]$ to learn the target to within $\epsilon^{1/k}/2$ accuracy with δ/k confidence. Obtain hypothesis $1_{[\hat{a_j},1]}$.

4. Stop with hypothesis $1_{[\hat{a_k},1]}$.

Proof of Correctness: Let the target be $1_{[a_t,1]}$. At the end of the first step, the target is within $\epsilon^{1/k}$ of the hypothesis with probability greater than $1 - \delta/k$. This means that with high probability $|a_t - \hat{a_1}| \leq \epsilon^{1/k}$ or in other words $\hat{a_1} - \epsilon^{1/k} \leq a_t \leq \hat{a_1} + \epsilon^{1/k}$.

We now draw examples only from the region $[\hat{a_1} - \epsilon^{1/k}, \hat{a_1} + \epsilon^{1/k}]$. Let this distribution be P_2. By a theorem of Vapnik and Chervonenkis, we need to draw $4/\epsilon^{2/k} \ln(k/\delta)$ examples to learn the target to within $\epsilon^{1/k}/2$ with δ/k confidence (for an arbitrary distribution) at this stage. This means that

$$d_{P_2}(1_{[a_t,1]}, 1_{[\hat{a_2},1]}) = 1/(2\epsilon^{1/k})|a_t - \hat{a_2}| \leq \epsilon^{1/k}/2$$

In other words,

$$|a_t - \hat{a_2}| \leq \epsilon^{2/k}$$

Thus after two steps, the above inequality is true. We now draw examples only from the region $[a_2 - \epsilon^{2/k}, a_2 + \epsilon^{2/k}]$.

In general, at the jth step, if we draw $4/\epsilon^{2/k} \ln(k/\delta)$ examples, we would have learnt the target to $\epsilon^{1/k}/2$ accuracy with δ/k confidence. The distribution (P_j) according to which examples are drawn at this stage is uniform over $[\hat{a_{j-1}} - \epsilon^{(j-1)/k}, \hat{a_{j-1}} + \epsilon^{(j-1)/k}]$. Thus,

$$d_{P_j}(1_{[a_t,1]}, 1_{[\hat{a_j},1]}) = 1/(2\epsilon^{(j-1)/k})|a_t - \hat{a_j}| \leq \epsilon^{1/k}/2.$$

So we have,

$$|a_t - \hat{a_j}| \leq \epsilon^{j/k}.$$

This happens with probability greater than $1 - \delta/k$. Thus with high probability, from the $(j-1)$th stage to the jth stage, we have "focused" more closely onto a_t. If this is true at every stage, we would eventually have after k steps ensured that

$$|\hat{a_k} - a_t| \leq \epsilon$$

which would mean that we have learnt the target to within an ϵ width.

If we fail at any stage, the eventual hypothesis a_k is not necessarily within an ϵ width of the target. The probability of failing at each stage is less than than δ/k so the probability of failing in at least one stage is less than $k.\delta/k = \delta$. Thus the probability of failing is less than δ or in other words with greater than $1-\delta$ probability, we would have learnt the target to within an ϵ width which was our goal.

The total number of examples drawn at each stage is $4/\epsilon^{2/k} \ln(k/\delta)$ and since there are k stages in all, the total number of examples in the whole process is

$4k/(\epsilon^{2/k})\ln(k/\delta)$.□

3.6 Generalizations

This general strategy can be extended to several other scenarios. We introduce the notion of localized function classes. These classes which have a *local focusing* property can be learned faster by the method of ϵ-focusing. We mention some concrete results obtained by using this scheme for n-dimensional cases, and for the case of noisy examples. No proofs or formal arguments are provided for these extensions. We hope, though, that the reader will appreciate the spirit of this idea.

3.6.1 Localized Function Classes

The previous sections showed how to use the ϵ-focusing strategy to obtain superior sample complexity results for some simple concept classes. It is of interest to characterize general conditions on function classes for which the ϵ-focusing strategy would yield such a superior performance. It is noteworthy that the previous function class had the property that knowledge of the distance between any two functions f and g in $\mathcal{F}$ (in the d_P metric) allowed us to focus in on a region of interest in the domain $X = [0,1]$ where f and g differ. We formalize this notion to derive a general bound on sample complexity for the ϵ-focusing strategy.

Let $\mathcal{F}$ be a concept class (i.e. class of indicator functions) on some compact domain X. Let P be the uniform distribution on this domain, i.e., the distribution which corresponds to the normalized Lebesgue measure on it. We define the usual $L_1(\mu)$ distance metric on the space functions by

$$d_\mu(f,g) = \int_X |f-g|d\mu$$

(where μ is a probability measure on the set X.)

We define the *local focusing* property of such an arbitrarily defined concept class as follows:

Definition 3.6.1 *For a given f belonging to some concept class $\mathcal{F}$ on X, and for any given $\epsilon > 0$, its ϵ-region of interest, $\mathcal{R}_\epsilon(f)$ is given by*

$$\{x \in X | f(x) \neq g(x) \textit{ for some } g \in \mathcal{F} \textit{ such that } d_P(f,g) \leq \epsilon\}$$

Definition 3.6.2 *The concept class $\mathcal{F}$ is said to be locally focused with focusing bound*

g (g is a real valued function taking values on $[0,1]$) if for every $\epsilon > 0$,

$$\sup_{f \in \mathcal{F}} Volume(\mathcal{R}_\epsilon(f)) \leq g(\epsilon)$$

Here, $Volume(s)$ for any set $s \subset X$, is simply the volume[17] of that set. We assume that $Volume(X) = 1$.

Clearly, locally focused classes are those with bounded ϵ regions of interest into which we can focus in the iterative manner of Algorithm 1.

3.6.2 The General ϵ-focusing strategy;

The general algorithm to learn such ϵ-focused classes is as follows:
Algorithm 2

1. Begin with the entire class $\mathcal{F}$, draw examples according to the uniform distribution P on X, (call this P_1) and attempt to learn the target ($f_t \in \mathcal{F}$) to $\epsilon^{1/k}$ with probability at least $1 - \delta/k$. Obtain hypothesis $\hat{f}_1$. Also obtain the reduced set of candidate target functions (version space),

   $$\mathcal{F}_1 = \{f \in \mathcal{F} | d_{P_1}(f, \hat{f}_1) \leq \epsilon^{1/k}\}$$

   Finally, also obtain the ϵ-region of interest:

   $$R_1 = \mathcal{R}_{\epsilon^{1/k}}(\hat{f}_1).$$

2. Draw examples according to a uniform distribution on R_1 (call this distribution P_2) and learn the target to $\epsilon^{2/k}/g(\epsilon^{1/k})$ (according to P_2) with probability greater than $1 - \delta/k$. Now obtain hypothesis $\hat{f}_2 \in \mathcal{F}_1$, the reduced version space:

   $$\mathcal{F}_2 = \{f \in \mathcal{F}_1 | d_{P_2}(f.\hat{f}_2) \leq \frac{\epsilon^{2/k}}{g(\epsilon^{1/k})}\}.$$

   and $R_2 = \mathcal{R}_{\epsilon^{2/k}}(\hat{f}_2)$.

3. Repeat step 2. In general, at the jth step, learn the target to $\frac{\epsilon^{j/k}}{g(\epsilon^{(j-1)/k})}$ (according to distribution P_j), and obtain $\hat{f}_j, \mathcal{F}_j$, and R_j in the obvious way.

[17] From a more formal perspective, one should really replace $Volume(s)$ by the measure on the set s, i.e., $P(s)$. Clearly, $P(X) = 1$. In our case, we assume that $Volume(X) = 1$. Since P is a uniform distribution, i.e., any point in this set is as likely as any other point, it follows that $P(s)$ is simply $Volume(s)$. We will continue to use this notation, but the reader will easily see that P can be used in general, and in fact, need not even be uniform.

4. Stop at the kth step and output hypothesis $\hat{f}_k$.

Proof of Learnability:

Recall that our eventual goal is to learn the unknown target f_t within ϵ accuracy (according to the distance metric d_P) with probability greater than $1 - \delta$.

Consider the first step. The target has been learned to $\epsilon^{1/k}$ accuracy with high confidence. The learner's hypothesis is $\hat{f}_1$. Clearly, with high probability (greater that $1 - \delta$), the target lies within in an $\epsilon^{1/k}$ ball around $\hat{f}_1$ (this is denoted by $\mathcal{F}_1$). According to our definition, all functions in $\mathcal{F}_1$ agree on the region outside of R_1. So we only need to sample the region R_1 which is what we do in the second step.

In the second step, we learn the target to $\epsilon^{2/k}/g(\epsilon^{1/k})$. This is according to a distribution P_2 (uniform on the region R_1). Again, the target, is within an $\epsilon^{2/k}/g(\epsilon^{1/k})$ ball of the hypothesis at this stage ($\hat{f}_2$). Thus,

$$d_{P_2}(\hat{f}_2, f_t) = \frac{Volume(\{x \in R_1 | \hat{f}_2(x) \neq f_t(x)\})}{Volume(R_1)} \leq \epsilon^{2/k}/g(\epsilon^{1/k})$$

But, $Volume(R_1) = g(\epsilon^{1/k})$. Therefore,

$$Volume(\{x \in R_1 | \hat{f}_2(x) \neq f_t(x)\}) \leq \epsilon^{2/k}$$

Clearly, then,

$$d_P(\hat{f}_2, f_t) = Volume(X \setminus R_1)(0) + Volume(\{x \in R_1 | \hat{f}_2(x) \neq f_t(x)\}) \leq \epsilon^{2/k}$$

Thus, after the second step, we see that the target f_t is within $\epsilon^{2/k}$ accuracy (with respect to our original distribution P). By our definition of the local focusing property, we know that $f_t \in \mathcal{F}_2$, and the points on which f_t and $\hat{f}_2$ disagree must lie within R_2.

In general, before the jth step, the points on which the target and the $(j-1)$th hypothesis disagree must lie within R_{j-1}. Since, we sample according to a uniform distribution on this (P_j), and attempt to learn the target to an accuracy of $\epsilon^{j/k}/g(\epsilon^{(j-1)/k})$, by a similar argument,

$$d_{P_j}(\hat{f}_2, f_t) = \frac{Volume(\{x \in R_{j-1} | \hat{f}_j(x) \neq f_t(x)\})}{Volume(R_{j-1})} \leq \epsilon^{j/k}/g(\epsilon^{(j-1)/k})$$

But, $Volume(R_{j-1}) = g(\epsilon^{(j-1)/k})$. Therefore,

$$Volume(\{x \in R_{j-1} | \hat{f}_j(x) \neq f_t(x)\}) \leq \epsilon^{j/k}$$

and,

$$d_P(\hat{f}_j, f_t) = Volume(X \setminus R_{j-1})(0) + Volume(\{x \in R_{j-1} | \hat{f}_j(x) \neq f_t(x)\}) \leq \epsilon^{j/k}$$

Thus, after the jth step, the learner has learned the target to $\epsilon^{j/k}$ accuracy. Further, according to our definition of the local focusing property, the points on which the learner and target disagree must lie within the set $R_j = \mathcal{R}_{\epsilon^{j/k}}(\hat{f}_j)$.

Clearly, after the kth step, the learner will have learned the target to ϵ accuracy. The only way, in which the learner could have made a mistake, is if it made a mistake on any one of the steps. The probability of making a mistake in each step is δ/k. The probability of making a mistake in any one is bounded by δ. Thus, the learner would have identified the target to ϵ accuracy with confidence greater than $1 - \delta$.

Sample Complexity: By the standard Vapnik Chervonenkis theorem, we see that at the jth stage, the learner will have to draw at most $O(\frac{g^2(\epsilon^{(j-1)/k})}{\epsilon^{2j/k}} \ln(k/\delta))$ examples to satisfy the learnability requirement of that stage. The total number of examples the learner needs would be

$$O(\sum_{j=1}^{k} \frac{g^2(\epsilon^{(j-1)/k})}{\epsilon^{2j/k}} \ln(k/\delta))$$

3.6.3 Generalizations and Open Problems

Now we are in a position to re-evaluate our simple example from this general perspective. It is easy to see that

1. **Opening Example:** For an arbitrary $f_a = 1_{[a,1]}$, we see that

   $$\mathcal{R}_\epsilon(f_a) = [a - \epsilon, a + \epsilon]$$

   Clearly, $g(\epsilon) = 2\epsilon$. The sample complexity is $O((k/\epsilon^{2/k}) \ln(k/\delta))$.

2. **Box Functions:** Consider the following class of indicator functions on $[0, 1]$.

   $$\mathcal{F} = \{1_{[a, b]} : 0 \leq a \leq b \leq 1\}$$

   For an arbitrary $f_{a,b} = 1_{[a,b]}$, we see that

   $$\mathcal{R}_\epsilon(f_{a,b}) = [a - \epsilon, a + \epsilon] \cup [b - \epsilon, b + \epsilon]$$

   Clearly, $g(\epsilon) = 4\epsilon$. The sample complexity $O((k/\epsilon^{2/k}) \ln(k/\delta))$ follows.

Some other generalizations should be noted. We do not attempt to provide any formal arguments.

1. Extensions to n-dimensions: It is possible to extend the ϵ focusing strategy of our opening example to an n-dimensional situation. A concrete example includes the PAC learning of a concept class of hyperplanes dividing an n-simplex into two regions. Essentially, the hyperplane cuts the simplex at its edges. Consequently, along each edge, the points on one side of the cut are labelled 0, while the points on the other side are labelled 1. Thus, if one confines oneself to finding the intersection of the hyperplane with the simplex edge, the problem reduces to a single dimensional case exactly like our opening example. If n such edge-intersection problems are solved, then the total n-dimensional problem can be solved.

In view of the fact that we have an effective ϵ-focusing strategy for box functions, we can even address concept classes represented by multilayer perceptrons with two hidden layers. In such a case, there are at most two hyperplanes intersecting each edge. The single-dimensional problem associated with each edge is like a box function.

2. Handling misclassification noise: The ϵ-focusing strategy in this part has been developed for a noise-free case. Extensions to cover a situation with a bound on the misclassification noise (the label of the example can be flipped with probability at most η) can easily be considered as well.

Finally, some natural questions arise at this stage. First, what kinds of concept classes have the locally focusing property? Second, given the existence of the locally focusing property, how easy is it to compute the ϵ-region of interest $\mathcal{R}_\epsilon$ for such concept classes. Further research on these questions is awaited.

Chapter 4

Language Learning Problems in the Principles and Parameters Framework

Abstract

This chapter considers a learning problem in which the hypothesis class is a class of parameterized grammars. After a brief introduction to the "principles and parameters" framework of modern linguistic theory, we consider a specific learning problem previously analyzed in a seminal work by Gibson and Wexler (1994). With our informational-complexity point of view developed in this thesis, we reanalyze their learning problem. This puts particular emphasis on the sample complexity of learning, in contrast to previous research in the inductive inference, or Gold frameworks (see Osherson and Weinstein, 1986). We show how to formally characterize this problem in particular, and a class of learning problems in finite parameter spaces in general, as a Markov structure. Important new language learning results follow directly: we explicitly compute sample complexity bounds under different distributional assumptions, learning regimes, and grammatical parameterizations. Briefly, we may view this as a precise way to model the "poverty of stimulus" children face in language acquisition. Our reanalysis alters several conclusions made by Gibson and Wexler. We therefore consider this chapter as a useful application of learning-theoretic notions to natural languages, and their acquisition. Finally, we describe several directions for further research.

In Chapters 2 and 3, we considered the problem of learning target functions (belonging to certain classes) from examples. Particular emphasis was given to the *sample complexity* of learning such functions, and we have seen how it depends upon the complexity of the hypothesis classes concerned. The classes of functions we have investigated, have arguably, very little cognitive relevance. However, the investigations have helped us to develop a point of view crucial to the analysis of learning systems—a point of view which allows us to appreciate the inherent tension between the approximation error, and the estimation error, in learning from examples. In particular we have seen how the hypothesis classes used by the learner must be large to reduce the approximation error, and small to reduce the estimation error. In the rest of the thesis (Chapters 4 and 5), we remedy our cognitive irrelevance by considering some classes of functions which linguists and cognitive scientists believe the brain must compute. As we shall soon see, there is a learning-theoretic argument at the heart of the modern approach to linguistics—hence our choice of linguistic structures for analysis. The origin of the research presented in this chapter lies in the paper "Triggers" (Gibson and Wexler, 1994; henceforth GW) which marks a seminal attempt to formally investigate language learning within the "principles and parameters" framework (Chomsky, 1981). The results presented in this chapter emerged out of a reanalysis of "Triggers" using more sophisticated mathematical techniques, than had previously been used in this context. One can, thus, regard this as a demonstration, of how our information-theoretic point of view, and the arguments and tools of current learning theory, can help us to sharpen certain important questions, and lead to insightful analysis of relevant linguistic theories.

In the next section, we provide a brief account of the learning-theoretic considerations inherent in the modern approach to linguistics. We then give a brief account of the principles and parameters framework, and the issues involved in learning within this framework. This sets the stage for our investigations, and we use as a starting point the Triggering Learning Algorithm (TLA) working on a three-parameter syntactic subsystem first analyzed by Gibson and Wexler. The rest of the chapter analyzes the TLA from the perspective of learnability and sample complexity. Issues pertaining to parameter learning in general, and the TLA in particular, are discussed at appropriate points. Finally, we suggest various directions for further research—this chapter marks only the opening of our research on this theme. Very little work has been done on the formal, computational, aspects of parameter setting, and we attempt here to pose questions which we think are of importance in the field.

4.1 Language Learning and The Poverty of Stimulus

The inherent tension between having large hypothesis classes, for greater expressive power, and small ones, for better learnability, is beautifully instantiated in the human language system. Humans develop a mature knowledge of language that is both rich and subtle, on exposure to fairly limited number (the so called "poverty of stimulus") of example sentences spoken by parents and guardians in childhood. Languages are infinite sets of sentences[18]. Yet on exposure to a finite number of them (during the

[18] There are an infinite number of sentences in the English language. You haven't heard all of them, yet you can judge the grammaticality of sentences you have not heard before. In the view of many linguists, you have internalized a grammar–a set of rules, a theory, or schema, by means of which you are able to generalize to unseen sentences (examples).

language acquisition phase in childhood) children correctly generalize to the infinite set. Further, they generalize in exactly the same way: too striking a coincidence to be attributed to chance. This motivated Chomsky (1965) to argue that children must operate with constrained hypotheses about language—constraints which restrict the sorts of generalizations that they can make. These constrained hypothesis classes which children operate with, in the language context, are classes of grammars. Children choose one particular grammar[19] from this class, on the basis of the examples they have seen. Thus, a child born in a Spanish speaking environment would choose the grammar which appropriately describes the data it has seen (Spanish sentences), and, similarly, a child born in a Chinese speaking environment chooses a different grammar, and so on. Of course, children might make mistakes, and they do. These mistakes are often resolved as more data becomes available to the child. Sometimes (when this happens, is undoubtedly, of great interest), these mistakes might never be resolved—a possibility which we explore in the next chapter.

Thus, we see, that if we were totally unconstrained in the kinds of hypotheses we could make, then, on the basis of a finite data set, we would all generalize in wildly different ways, implying, thereby, that we would never be able to learn languages. Yet, we learn languages, apparently with effortless ease as children. This realization is crucial to linguistics. Humans, thus, are predisposed to choose certain generalizations over others, they are predisposed to choose hypotheses belonging to a constrained class of grammars—this predisposition is the essence of the innatist view of language; the universal constraints on the class of grammars belong to universal grammar. Furthermore, such a class of grammars must be large enough to capture the richness of language, yet small enough to be learned— exemplifying the tension discussed previously. The thrust thus shifted to finding the right constraints incorporated in such a class of grammars, in other words, finding the class of grammars of the right complexity. Notice, here, the similarity in spirit to the problem of finding a regularization network of the right complexity. Consequently, we see that an analysis

[19] It should be pointed out that there are various components of a language. There is its syntax, that concerns itself with syntactic units like verbs, noun phrases, etc. and their appropriate combinations. Further, there is its phonology that deals with its sound structure, its morphology that deals with word structure, and finally, the vocabulary or "words" which are the building blocks out of which sentences are ultimately composed. Acquisition of a language involves the acquisition of all of this. We have been using the term grammar in a loose sort of way—it is a system of rules and principles which govern the production of acceptable sentences of the language. The grammar too could be broken into its syntactic parts, its phonological parts and so on. Some readers, recalling vivid memories of stuffy English school teachers, might have a natural resistance to the idea of rigid rules of grammaticality. For such people, we note, that while there is undoubtedly greater flexibility in word order than such teachers would suggest, it is a fact, that no one speaks "word salad"—with absolutely no attention to word order combinations at all.

of the complexity of language learning coupled with a computational view of the language acquisition device is crucial to the theoretical underpinnings of modern linguistics (see Wexler and Culicover (1980) for an excellent formal exposition of this idea).

4.2 Constrained Grammars–Principles and Parameters

Having recognized the need for constraints on the class of grammars (this can be regarded as an attempt to build a hypothesis class with finite learnability dimension[20]) researchers have investigated several possible ways of incorporating such constraints in the classes of grammars to describe the natural languages of the world. Examples of this range from linguistically motivated grammars such as Head-driven Phrase Structure Grammars (HPSG), Lexical-Functional grammars, Optimality theory for phonological systems, to bigrams, trigrams and connectionist schemes suggested from an engineering consideration of the design of spoken language system. Note that every such grammar suggests a very specific model for human language, with its own constraints and its own complexity. Model-free, unconstrained, tabula rasa learning schemes correspond to hypothesis classes with infinite dimension, and these can never be learned in finite time. An important program of research consists of computing the sample complexity of learning each of these diverse classes of grammars.

In this chapter, we conduct our investigations within the purview of the principles and parameters framework (Chomsky, 1981). Such a framework attempts to capture the "universal" principles common to all the natural languages of the world, (part of our biological endowment as human beings possessed of the unique language faculty) and the parameters of variation across languages of the world. Roughly speaking, there are a finite number of principles governing the production of human languages. These abstract principles, can take one of several (finite) specific forms—this specific form manifests itself as a rule, peculiar to a particular language (or classes of languages). The specific forms that such an abstract principle can take is governed by setting an associated parameter to one of several values. In typical versions of theories constructed within such a framework, one ends up with a parameterized

[20] In previous chapters, we have utilized the notion of VC-dimension, and pseudo-dimension to characterize the complexity of learning real-valued function classes. It is not immediately clear, what complexity measure should be used for characterizing classes of grammars–the development of a suitable measure, in tune with the demands of the language acquisition process, is an open question.

class of grammars. The parameters are boolean valued–setting them to one set of values, defines the grammar of German (say), setting them to another set of values, defines the grammar, perhaps, of Chinese. Specific examples of theories within such a framework could include Government and Binding, Head-driven Phrase Structure Grammar, Optimality Theory, varieties of lexical-functional grammars and so forth. The idea is best illustrated in the form of examples. We provide, now, two examples, drawn from syntax, and phonology, respectively.

4.2.1 Example: A 3-parameter System from Syntax

Two X-bar parameters: A classic example of a parametric grammar for syntax comes from X-bar theory (Chomsky, 1981; Haegeman, 1991). This describes a parameterized phrase structure grammar, which defines the production rules for phrases, and ultimately sentences in the language. The general format for phrase structure is summarized by the following parameterized production rules:

$$XP \rightarrow SpecX'(p_1 = 0) \text{ or } X'Spec(p_1 = 1)$$
$$X' \rightarrow CompX'(p_2 = 0) \text{ or } X'Comp(p_2 = 1)$$
$$X' \rightarrow X$$

XP refers to an X-phrase, where X, or the "head", is a lexical category like N (Noun), V (Verb), A (Adjective), P (Preposition), and so on. Thus, one could generate NP, or Noun Phrases, VP, or Verb Phrases, and other phrases in this fashion. *Spec* refers to *specifier*, in other words, that part of the phrase that "specifies" it, roughly like *the old* in *the old book*. *Comp* refers to the *complement*, roughly a phrase's arguments, like *an ice-cream* in the Verb Phrase *ate an ice-cream*, or *with envy* in the Adjective Phrase *green with envy*. Both *Spec* and *Comp* can themselves be phrases with their own specifiers and complements. Furthermore, in a particular phrase, the spec-position, or the comp-position might be blank (in these cases, $Spec \rightarrow \emptyset$, or $Comp \rightarrow \emptyset$ respectively). Applying these rules recursively, one can thus generate embedded phrases of arbitrary length in the language. Further, these rules are parameterized. Languages can be spec-first ($p_1 = 0$) or spec-final ($p_1 = 1$). Similarly, they can be comp-first, or comp-final. For example, the parameter settings of English are (spec-first,comp-final). Shown in fig. 4-33 is an embedded phrase which demonstrates the use of the X-bar production rules (with the English parameter settings) to generate an arbitrary English phrase.

In contrast, the parameter settings of Bengali are (spec-first,comp-first). The

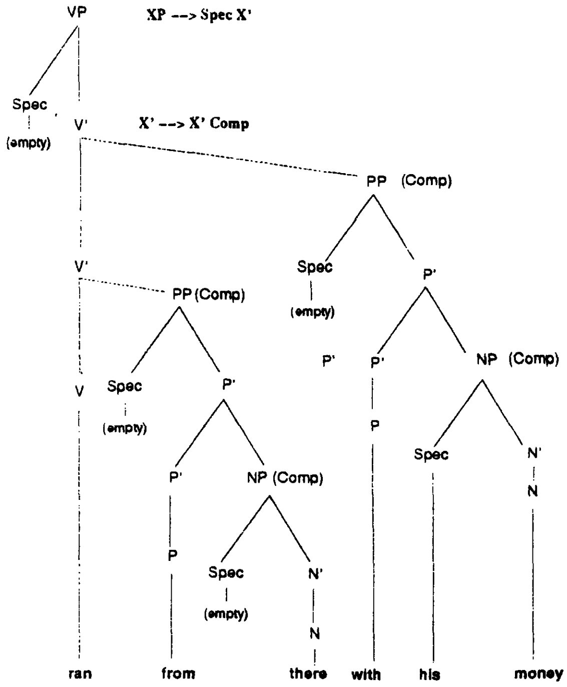

Figure 4-33: Analysis of an English sentence. The parameter settings for English are spec-first, and comp-final.

translation of the same sentence is provided in fig. 4-34. Notice, how a difference in the comp-parameter setting causes a difference in word orders. It is claimed that as far as basic, underlying word order is concerned, X-bar theory covers all the possibilities for natural languages[21]. Languages of the world simply differ in their parameter settings.

One transformational parameter (V2): The two parameters described above define generative rules to obtain basic word-order combinations permitted in the world's languages. As mentioned before, there are many other aspects which govern the formation of sentences. For example, there are transformational rules which determine the production of surface word order from the underlying (base) word-order structure obtained from the production rules above. One such parameterized transformational rule that governs the movement of words within a sentence is associated with the *V2* parameter. It is observed that in German and Dutch declarative sentences, the relative order of verbs and their complements seem to vary depending upon whether the clause in which they appear is a root clause or subordinate clause. Consider, the

[21] There are a variety of other formalisms developed to take care of finer details of sentence structure. This has to do with case theory, movement, government, binding and so on. See Haegeman (1991).

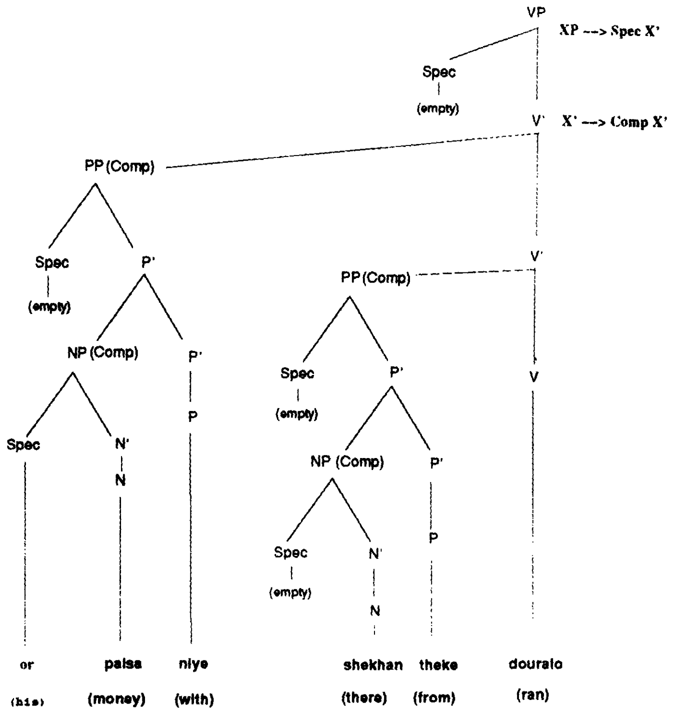

Figure 4-34: Analysis of the Bengali translation of the English sentence of the earlier figure. The parameter settings for Bengali are spec-first, and comp-first.

following German sentences:

(1)...dass (that) Karl das (the) Buch (book) kauft (buys).

...that Karl buys the book.

(2)...Karl kauft das Buch.

...Karl buys the book.

This seems to present a complication in that from these sentences it is not clear whether German is comp-first (as example 1 seems to suggest) or comp-final (as example 2 seems to suggest). It is believed (Haegeman, 1991) that the underlying word-order form is comp-first (like Bengali, and unlike English, in this respect); however, the $V2$ parameter is set for German ($p_3 = 1$). This implies that finite verbs must appear in the exact second position in root declarative clauses ($p_3 = 0$ would mean that this need not be the case). This is a specific application of a transformational rule Move-α. For details and analysis, see (Haegeman, 1991).

Each of these three parameters can take one of two values. There are, thus, 8 possible grammars, and correspondingly 8 languages by extension, generated in this fashion. At this stage, the languages are defined over a vocabulary of syntactic categories, like N, V etc. Applying the three parameterized rules, one would obtain different ways of combining these syntactic categories to obtain sentences. Appendix A is a list of the set of unembedded (degree-0) sentences obtained for each of the languages, L_1 through L_8 in this parametric system. The vocabulary has been modified so that sentences are now defined over more abstract units than syntactic categories.

4.2.2 Example: Parameterized Metrical Stress in Phonology

The previous example dealt with a parameterized family for syntax. As we mentioned before, syntax is only one component of language. Here we consider an example from phonology; in particular, our example deals with metrical stress which describes the possible ways in which words in a language can be stressed.

Consider the English word, "candidate". This is a three syllable word, composed of the three syllables, /can/,/di/,and, /date/. A native speaker of American English typically pronounces this word by stressing the first syllable of this word. Similarly, such a native speaker would also stress the first syllable of the tri-syllabic word, "/al/-/pha/-/bet/" so that it almost rhymes with "candidate". In contrast, a French speaker would stress the final syllable of both these words—a contrast which is perceived as a "French" accent by the English ear.

For simplicity, assume that stress has two levels, i.e., each syllable in each word can be either stressed, or unstressed[22]. Thus, an n-syllable long word could have, in principle, as many as 2^n different possible ways of being stressed. For a particular language, however, only one of these ways is phonologically well-formed. Other stress patterns sound accented, or awkward. Words could potentially be of arbitrary length[23]. Thus one could write phonological grammars—a functional mapping from these words to their correct stress pattern. Clearly, this is another example of a functional mapping the brain must compute. Further, different languages correspond to different such functions,i.e., they correspond to different phonological grammars. Within the principles and parameters framework, these grammars are parameterized as well.

Let us consider a simplified version of two principles associated with 3 boolean valued parameters which play a role in the Halle and Idsardi metrical stress system. These principles describe how a multisyllable word can be broken into its constituents (recall how sentences were composed of constituent phrases in syntax) before stress assignment takes place. This is done by a bracketing schema which places brackets at different points in the word, thereby marking (bracketing) off different sections as constituents. A constituent is then defined as a syllable sequence between consecutive brackets. In particular, a constituent must be bounded by a right bracket on its right edge, or, a left bracket on its left edge (both these conditions need not be satisfied simultaneously). Further, it cannot have any brackets in the middle. Finally, note that not all syllables of the word need be part of a constituent. A sequence of syllables might not be bracketed by either an appropriate left, or right bracket—such a sequence, cannot have a stress-bearing head, and might be regarded as an extra-metrical sequence.

1) the **edge** parameters: there are two such parameters.

a) put a left ($p_1 = 0$) or right ($p_1 = 1$) bracket

b) put the above mentioned bracket exactly one syllable *after* the left ($p_2 = 0$) edge or *before* the right ($p_2 = 1$) edge of the word.

2) the **head** parameter: each constituent (made up of one or more syllables) has a

[22] While we have not provided a formal definition of either stress, or syllable, it is hoped, that at some level, the concepts are intuitive to the reader. It should, however, be pointed out that linguists differ on their characterization of both these objects. For example, how many levels can stress have? Typically, (Halle and Idsardi, 1991) three levels are assumed. Similarly, syllables are classified into heavy and light syllables. We have discounted such niceties for ease of presentation.

[23] One shouldn't be misled by the fact that that a particular language has only a finite number of words. When presented with a foreign word, or a "non-sense" word one hasn't heard before, one can still attempt to pronounce it. Thus, the system of stress assignment rules in our native language probably dictates the manner in which we choose to pronounce it. Speakers of different languages would accent these non-sense words differently.

"head". This is the stress bearing syllable of the constituent, and is in some sense, the primary, or most important syllable of that constituent (recall how syntactic constituents, the phrases, had a lexical head). This phonological head could be the *leftmost* ($p_3 = 0$), or, the *rightmost* ($p_3 = 1$) syllable in the constituent.

Suppose, the parameters are set to the following set of values: [$p_1 = 0,\ p_2 = 0,\ p_3 = 0$]. Fig. 4-35 shows how some multisyllable words would have stress assigned to them. In this case, any n-syllable word would have stress in exactly the second position (if such a position exists) and no other. In contrast, if [$p_1 = 0,\ p_2 = 0,\ p_3 = 1$], the corresponding language would stress the final syllable of all multi-syllable words. Monosyllabic words are unstressed in both languages.

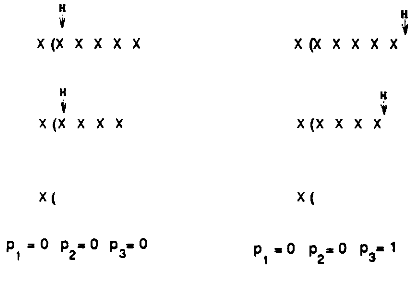

Figure 4-35: Depiction of stress pattern assignment to words of different syllable length under the parameterized bracketing scheme described in the text.

These 3 parameters represent a very small (almost trivial) component of stress pattern assignment. There are many more parameters which describe in more complete fashion, metrical stress assignment. At this level of analysis, for example, the language Koya has $p_3 = 0$, while Turkish has $p_3 = 1$; see Kenstowicz (1992) for more details. The point of this example was to provide a flavor or how the problem of stress-assignment can be described formally by a parametric family of functions. The analysis of parametric spaces developed in this chapter can be equally well applied to such stress systems.

4.3 Learning in the Principles and Parameters Framework

Language acquisition in the principles and parameters framework reduces to the setting of the parameters corresponding to the "target" language. A child is born in an arbitrary linguistic environment. It receives examples in the form of sentences it hears in its linguistic environment. On the basis of example sentences it hears, it presumably learns to set the parameters appropriately. Thus, referring to our 3-parameter system for syntax, if the child is born in a German speaking environment, and hears German sentences, it should learn to set the V2 parameter, and the spec-parameter to spec-first. Similarly, a child hearing English sentences, should learn to set the comp-parameter to comp-final. In principle, the child is thus solving a parameter estimation problem—an unusual class of parameter estimation problems, no doubt, but in spirit, little different from the parameter estimation problem associated with the regularization networks of Chapter 2. One can thus ask a number of questions about such problems. What sort of data does the child need in order to set the target parameters? Is such data readily available to the child? How often is such data made available to the child? What sort of algorithms does the child use in order to set the parameters? How efficient are these algorithms? How much data does the child need? Will the child always converge to the target "in the limit" ??

Language acquisition, in the context of parameterized linguistic theories, thus, gives rise to a class of learning problems associated with finite parameter spaces. Furthermore, as emphasized particularly by Wexler in a series of works (Hamburger and Wexler, 1975; Culicover and Wexler, 1980; and Gibson and Wexler, 1994), the finite character of these hypothesis spaces does *not* solve the language acquisition problem. As Chomsky noted in *Aspects of the Theory of Syntax* (1965), the key point is how the space of possible grammars- even if finite–is "scattered" with respect to the primary language input data. It is logically possible for just two grammars (or languages) to be so near each other that they are not separable by psychologically realistic input data. This was the thrust of Wexler and Hamburger, and Wexler and Culicover's earlier work on the learnability of transformational grammars from simple data (with at most 2 embeddings). More recently, a significant analysis of specific parameterized theories has come from Gibson and Wexler (1994). They propose the Triggering Learning Algorithm—a simple, psychologically plausible algorithm which children might conceivably use to set parameters in finite parameter spaces. Investigating the performance of the TLA on the 3-parameter syntax subsystem shown in the example yields the surprising result, that the TLA cannot achieve the target parameter setting for every possible target grammar in the system. Specifically, there are certain target parameter settings, for which the TLA could get stuck in *local maxima* from which it would never be able to leave, and consequently, learnability would never result.

We are interested, both in the *learnability*, and the *sample complexity* of the finite hypothesis classes suggested by the principles and parameters theory. An investigation of this sort requires us to define the important dimensions of the learning problem—the issues which need to be systematically addressed. The following figure provides a schematic representation of the space of possibilities which need to be explored in order to completely understand and evaluate a parameterized linguistic theory from a learning perspective. The important dimensions are as follows:

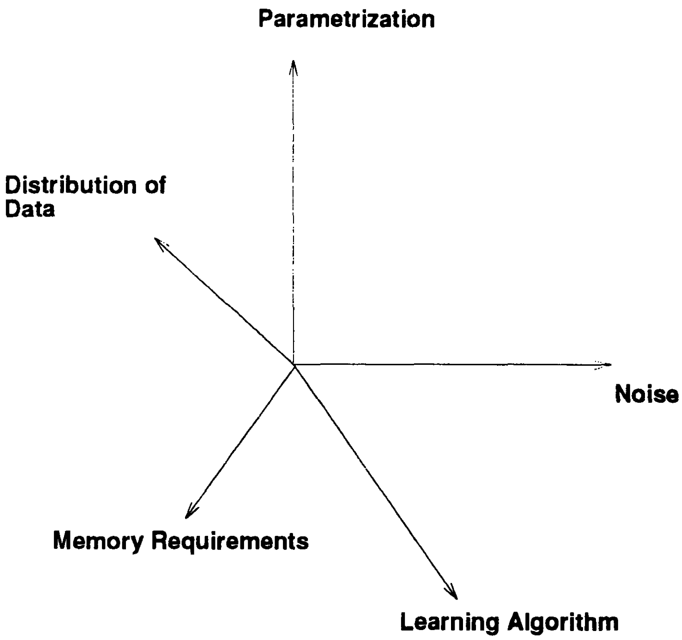

Figure 4-36: The space of possible learning problems associated with parameterized linguistic theories. Each axis represents an important dimension along which specific learning problems might differ. Each point in this space specifies a particular learning problem. The entire space represents a class of learning problems which are interesting.

1) the *parameterization* of the language space itself: a particular linguistic theory would give rise to a particular choice of universal principles, and associated parameters. Thus, one could vary along this dimension of analysis, the parameterization hypothesis classes which need to be investigated. The parametric system for metrical stress (Example 2) is due to Halle and Idsardi. A variant, investigated by Dresher and Kaye (1990), can equally well be subjected to analysis.

2)the *distribution* of the input data: once a parametric system is decided upon, one must, then, decide the distribution according to which data (i.e., sentences generated by some target grammar belonging to the parameterized family of grammars) is presented to the learner. Clearly, not all sentences occur with equal likelihood. Some are more likely than others. How does this affect learnability? How does this affect sample complexity? One could, of course, attempt to come up with distribution-independent bounds on the sample complexity. This, as we shall soon see, is not possible.

3) the presence, and nature, of *noise*, or extraneous examples: in practice, children are exposed to noise (sentences, which are inconsistent with the target grammar) due to the presence of foreign, or idiosyncratic speakers, disfluencies in speech, or a variety of other reasons. How does one model noise? How does it affect sample complexity or learnability or both?

4) the type of learning *algorithm* involved: a learning algorithm is an effective procedure mapping data to hypotheses (parameter values). Given that the brain has to solve this mapping problem, it then becomes of interest, to study the space of algorithms which can solve it. How many of them converge to the target? What is their sample complexity? Are they psychologically plausible?

5) the use of *memory*: this is not really an independent dimension, in the sense, that it is related to the kind of algorithms used. The TLA, and variants, as we shall soon see, are memoryless algorithms. These can be modeled by a Markov chain.

This is the space which needs to be explored. By making a specific choice along each of the five dimensions discussed (corresponding to a single point in the 5-dimensional space of fig. 4-36, we arrive at a specific learning problem. Varying the choices along each dimension (thereby traversing the entire space of fig. 4-36) gives rise to the class of learning problems associated with parameterized linguistic theories. For our analysis, we choose as a concrete starting point the Gibson and Wexler Triggering Learning Algorithm (TLA) working on the 3-parameter syntactic subsystem in the example shown. In our space of language learning problems, this corresponds to (1) a 3-way parameterization, using mostly X-bar theory; (2) a uniform sentence distribution over unembedded (degree-0) sentences; (3) no noise; (4) a local gradient ascent search algorithm; and (5) memoryless (online) learning. Following our analysis of this learning system, we consider variations in learning algorithms, sentence distribution, noise, and language/grammar parameterizations.

4.4 Formal Analysis of the Triggering Learning Algorithm

Let us start with the TLA. We first show that this algorithm and others like it is completely modeled by a Markov chain. We explore the basic computational consequences of this fundamental result, including some surprising results about sample complexity and convergence time, the dominance of random walk over gradient ascent, and the applicability of these results to actual child language acquisition, and possibly language change.

Background. Following Gold (1967) the basic framework is that of *identification in the limit.* We assume some familiarity with Gold's assumptions. The learner receives an (infinite) sequence of (positive) example sentences from some target language. After each, the learner either (i) stays in the same state; or (ii) moves to a new state (change its parameter settings). If after some finite number of examples the learner converges to the correct target language and never changes its guess, then it has correctly identified the target language in the limit; otherwise, it fails.

In the GW model (and others) the learner obeys two additional fundamental constraints: (1) the *single-value constraint*—the learner can change only 1 parameter value each step; and (2) the *greediness constraint*—if the learner is given a positive example it cannot recognize and changes one parameter value, finding that it can accept the example, then the learner retains that new value. The TLA can then be precisely stated as follows. See Gibson and Wexler (1994) for further details.

- [Initialize] Step 1. Start at some random point in the (finite) space of possible parameter settings, specifying a single hypothesized grammar with its resulting extension as a language;

- [Process input sentence] Step 2. Receive a positive example sentence s_i at time t_i (examples drawn from the language of a single target grammar, $L(G_t)$), from a uniform distribution on the degree-0 sentences of the language (we shall be able to relax this distributional constraint later on);

- [Learnability on error detection] Step 3. If the current grammar parses (generates) s_i, then go to Step 2; otherwise, continue.

- [Single-step gradient-ascent] Select a single parameter at random, uniformly with probability $1/n$, to flip from its current setting, and change it (0 mapped to 1, 1 to 0) *iff that change allows the current sentence to be analyzed*; otherwise go to Step 2;

Of course, this algorithm never halts in the usual sense. GW aim to show under what conditions this algorithm converges "in the limit"—that is, after some number, n, of steps, where n is unknown, the correct target parameter settings will be selected and never be changed. They investigate the behavior of the TLA on the linguistically natural 3-parameter syntactic subsystem of example 1. Note that a *grammar* in this space is simply a particular n-length array of 0's and 1's; hence there are 2^n possible grammars (languages). Gibson and Wexler's surprising result is that the simple 3-parameter space they consider is unlearnable in the sense that positive-only examples can lead to *local maxima*– incorrect hypotheses from which a learner can never escape. More broadly, they show that learnability in such spaces is still an interesting problem, in that there is a substantive learning theory concerning feasibility, convergence time, and the like, that must be addressed beyond traditional linguistic theory and that might even choose between otherwise adequate linguistic theories.

Triggers: Various researchers (Lightfoot, 1991; Clark and Roberts, 1993; Gibson and Wexler, 1994; Frank and Kapur, 1992) have explored the notion of *triggers* as a way to model parameter space language learning. Intuitively, triggers are supposed to represent evidence which allows the child to set the parameter for the target language. Concretely, Gibson and Wexler define triggers to be sentences from the target which allow a parameter to be correctly set. Thus, *global* triggers for a particular parameter are sentences from the target language which force the learner to set that parameter correctly (irrespective of the learner's current hypothesis about the target parameter settings). On the other hand, *local* triggers for a particular parameter depend upon the learner's hypothesis. Given values for all parameters but one (the parameter in question), local triggers are sentences which force the learner to correctly set the value of that parameter.

Gibson and Wexler suggest that the existence of local triggers for every (hypothesis,target) pair in the space suffices for TLA learnability to hold. As we shall see later, one important corollary of our stochastic formulation shows that this condition does *not* suffice. In other words, even if a *triggered* path exists from the learner's hypothesis language to the target, the learner might, with high probability, not take this path, resulting in non-learnability. A further consequence is that many of Gibson and Wexler's proposed cures for nonlearnability in their example system, such as "maturational" ordering imposed on parameter settings, simply do not apply. On the other hand, this result reinforces Gibson and Wexler's basic point that seemingly simple parameter-based language learning models can be quite subtle—so subtle that even a superficially complete computer simulation can fail to uncover learnability problems.

4.4.1 The Markov formulation

From the standpoint of learning theory, GW leave open several questions that can be addressed by a more precise formalization of this model in terms of Markov chains (a possible formalization suggested but left unpursued in footnote 9 of GW).

Consider a parameterized grammar (language) family with n parameters. We can picture the hypothesis space, of size 2^n, as a set of points, each corresponding to one particular array of parameter settings (languages, grammars). Call each point a *hypothesis state* or simply *state* of this space. As is conventional, we define these languages over some alphabet Σ as a subset of Σ^*. One of them is the target language (grammar). We arbitrarily place the (single) target grammar at the center of this space. Since by the TLA the learner is restricted to moving at most 1 binary value in a single step, the theoretically possible transitions between states can be drawn as (directed) lines connecting parameter arrays (hypotheses) that differ by at most 1 binary digit (a 0 or a 1 in some corresponding position in their arrays). Recall that this is the so-called *Hamming distance*.

We may further place *weights* on the transitions from state i to state j. These correspond to the probabilities that the learner will move from hypothesis state i to state j. In fact, as we shall show below, given a distribution over $L(G_t)$, we can further carry out the calculation of the actual transition probabilities themselves. Thus, we can picture the TLA learning space as a directed, labeled graph V with 2^n vertices.[24] More precisely, we can make the following remarks about the TLA system GW describe.

Remark. The TLA system is *memoryless*, that is, given a sequence s of sentences up to time t_i, the selection of hypothesis $h(t_{i+1})$ depends only on sentence $s(t_i)$, and not (directly) on previous sentences, i.e.,

$$p\{h(t_{i+1}) = h|h(t), s(t), t \leq t_i\} = P\{h(t_{i+1}) = h|h(t_i), s(t_i)\}$$

In other words, the TLA system is a classical *discrete stochastic process*, in particular, a discrete *Markov process* or Markov chain. We can now use the theory of Markov chains to describe TLA parameter spaces (Isaacson and Masden, 1976). For example, as is well known, we can convert the graphical representation of an n-dimensional Markov chain M to an $n \times n$ matrix T, where each matrix entry (i, j) represents the transition probability from state i to state j. A single step of the Markov process is computed via the matrix multiplication $T \times T$; n steps is given by T^n. A "1" entry in any cell (i, j) means that the system will converge with probability 1 to state j, given that it starts in state i.

As mentioned, not all these transitions will be possible in general. For example, by the single value hypothesis, the system can only move 1 Hamming bit at a time. Also, by assumption, only differences in surface strings can force the learner from one hypothesis state to another. For instance, if state i corresponds to a grammar that

[24] GW construct an identical transition diagram in the description of their computer program for calculating local maxima. However, this diagram is not explicitly presented as a Markov structure; it does not include transition probabilities. Of course, topologically both structures must be identical.

generates a language that is a proper subset of another grammar hypothesis j, there can never be a transition from j to i, and there must be one from i to j. Further, by assumption and the TLA, it is clear that once we reach the target grammar there is nothing that can move the learner from this state, since all remaining positive evidence will not cause the learner to change its hypothesis. Thus, there must be a loop from the target state to itself and no exit arcs. In the Markov chain literature, this is known as an *Absorbing State* (AS). Obviously, a state that only leads to an AS will also drive the learner to that AS. Finally, if a state corresponds to a grammar that generates some sentences of the target there is always a loop from that state to itself, that has some nonzero probability.

Example.

Consider the 3-parameter syntax subsystem of Example 1. Its binary parameters are: (1) Spec(ifier) first (0) or last (1); (2) Comp(lement) first (0) or last (1); and Verb Second (V2) does not exist (0) or does exist (1). As discussed in the example, the 3 parameters give rise to 8 distinct grammars. Further, these grammars generate different combinations of syntactic categories.

Rather than considering categories of the form Noun, Adjective, and so on, one could use more abstract constituents to define the vocabulary of the language. One possible approach is to allow the usage of phrases as possible "words" in the language. This is what GW choose to do. The net result is that the grammars are now defined over a vocabulary, $\Sigma = \{S, V, O, O1, O2, Adv, Aux\}$, corresponding to Subject, Verb, Object, Direct Object, Indirect Object, Adverb, and Auxiliary verb. See Haegeman (1991) for an account of such a transformation. Sentences in $\Sigma*$ now correspond to concatenations of these basic "words"–which are really phrases.

For instance, parameter setting (5) corresponds to the array [0 1 0]= Specifier first, Comp last, and −V2, which works out to the possible basic English surface phrase order of Subject–Verb–Object (SVO). As shown in GW's figure (3), the other possible arrangements of surface strings corresponding to this parameter setting include SV (as in *John runs*); SV O1 O2 (two objects, as in *give John an ice-cream*); S Aux V (as in *John is running*; S Aux V O; S Aux V O1 O2; Adv S V (where Adv is an Adverb, like *quickly*; Adv S V O; Adv S V O1 O2; Adv S Aux V; Adv S Aux V O; and Adv S Aux V O1 O2. Shown in appendix A of this chapter are all the possible degree-0 (unembedded) sentences generated by the 8 possible grammars of this parametric system.

Suppose SOV (setting #5=[0 1 0]) is the target grammar (language). With the GW 3-parameter system, there are $2^3 = 8$ possible hypotheses, so we can draw this as an 8-point Markov configuration space, as shown in fig. 4-37. The shaded rings represent increasing Hamming distances from the target. Each labeled circle is a Markov state, a possible array of parameter settings or grammar, hence extensionally specifies a possible target language. Each state is exactly 1 binary digit away from its possible transition neighbors. Each directed arc between the points is a possible (nonzero) transition from state i to state j; we shall show how to compute this immediately below. We assume that the target grammar, a double circle, lies at the center. This corresponds to the (English) SOV language. Surrounding the bulls-eye target are the 3 other parameter arrays that differ from [0 1 0] by one binary digit each: we picture these as a ring 1 Hamming bit away from the target: [0, 1, 1], corresponding to GW's parameter setting #6 in their figure 3 (Spec-first, Comp-final, +V2, basic order SVO+V2); [0 0 0], corresponding to GW's setting #7 (Spec-first, Comp-first, −V2), basic order SOV; and [1 1 0], GW's setting #1 (Spec-final, Comp-final, −$V2$], basic order VOS.

Around this inner ring lie 3 parameter setting hypotheses, all 2 binary digits away from the target: [0 0 1], [1 0 0], and [1 1 1] (grammars #2, 3, and 8 in GW figure 3). Note that by the Single Value hypothesis, the learner can only move one grey ring towards or away from the target at any one step. Finally, one more ring out, three binary digits different from the target, is the hypothesis [1 0 1], corresponding to target grammar 4.

Using this picture, we can also now readily interpret some of the terminological notions in GW's article. A **local trigger** is simply a datum that would allow the learner to move along an *ingoing* link in the figure. This is because an ingoing link is associated with sentences which allow the learner to move 1 bit closer to the target in parameter space, and consequently, set one parameter correctly. For example, the link from grammar state 3 to grammar state 7 *does* correspond to a local trigger, as does the link from 4 to 2; however, the link from grammar 3 to 4 is *not* a local trigger. Also, because of the Single Value and Greediness constraints, the learner can only either (i) stay in its current state; (ii) move 1 step inwards (a local trigger); or (iii) move 1 step outwards (note that this also happens given data from the target, just as in Case (ii)). These are the only allowed moves; one cannot move to another state within the same ring.

One can also describe the learnability properties of this space more formally. In this Markov chain, certain states have no outgoing arcs; these are among the *Absorbing States* (AS) because once the system has made a transition into one of these states, it can never exit. More generally, let us define the set of **closed states** CS to be any proper subset of states in the Markov chain such that there is no arc from any of the states in CS to any other state in the Markov chain.

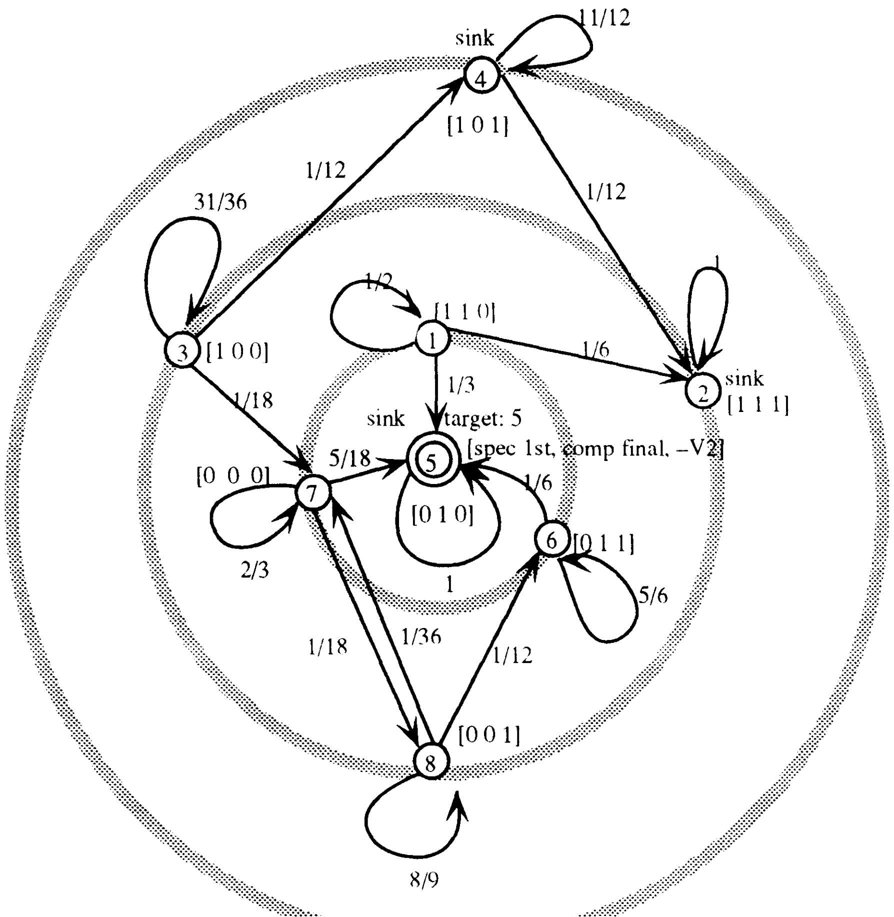

Figure 4-37: The 8 parameter settings in the GW example, shown as a Markov structure. Directed arrows between circles (states, parameter settings, grammars) represent possible nonzero (possible learner) transitions. The target grammar (in this case, number 5, setting [0 1 0]), lies at dead center. Around it are the three settings that differ from the target by exactly one binary digit; surrounding those are the 3 hypotheses two binary digits away from the target; the third ring out contains the single hypothesis that differs from the target by 3 binary digits. Note that the learner can either cycle or step in or out one ring (binary digit) at a time, according to the single-step learning hypothesis; but some transitions are not possible because there is no data to drive the learner from one state to the other under the TLA. Numbers on the arcs denote transition probabilities between grammar states; these values are not computed by the original GW algorithm.

Note that in the systems under discussion the target state is always an Absorbing State (once the learner is at the target grammar, it can never exit), so the Markov chains we will consider always have at least one AS. In the example 3-parameter system, state 2 is also an Absorbing State. Given this formulation, one can immediately give a very simple criterion for the learnability of such parameter spaces operated upon by the TLA[25].

Theorem 4.4.1 *Given a Markov chain C corresponding to a parameter space, a target parameter setting, and a GW TLA learner that attempts to learn the target parameters,* $\exists$ exactly *1 AS (corresponding to the target grammar) and no other CS iff target parameters can be correctly set by the TLA in the limit (with probability 1).*

Proof. $\Leftarrow$. By assumption, C is learnable. Now assume for sake of contradiction that there is more than one CS. Pick the CS which is not the target state. If the learner starts in this state, it can never reach the target AS, by the definition of a closed state. This contradicts the assumption that the space was learnable.

$\Rightarrow$. Assume that there exists exactly 1 AS in the Markov chain M and no other closed states CS. There are two cases. Case (i): at some time the learner reaches the target state. Then, by definition, the learner has converged and the system is learnable. Case (ii): there is no time at which the learner reaches the target state. Then the learner must move among a set of nontarget states. But this by definition forms a closed set of states distinct from the target, a contradiction. ▮

It is also of interest to be able to compute the set of inital states from which the TLA learner is guaranteed to converge to the target state. The following corollary describes these states.

Corollary 4.4.1 *Given a Markov chain C corresponding to a GW TLA learner, the set of learnable initial states is exactly the set of states that are connected to the target and unconnected to the nontarget closed states of the Markov chain.*

It is easy to see from inspection of the figure that there are exactly 2 absorbing states in this Markov chain, that is, states that have no exit arcs. One AS is the target grammar (by definition). The other AS is state 2. Correspondingly, by our theorem above, the target is not learnable by the TLA. This is correctly noted by Gibson and Wexler. In an attempt to obtain a list of initial states from which the learner is unable to reach the target, Gibson and Wexler, list only states 2, and 4.

[25] Any memoryless algorithm operating on this finite parameter space can be modeled as a first-order Markov chain. See appendix B of this chapter. The theorem is true for all such algorithms, not just the TLA

State 2, as we have seen is an additional AS, clearly the learner will not reach the target from here. State 4 is unconnected to the target by any path in the chain, clearly, the learner cannot reach the target from here as well. They compute the list of problematic initial states as those, from which the learner can never reach the target, in other words, those states which are **unconnected** to the target. They have implicitly assumed that if a triggered path to the target exists, it will be taken with probability one. This need not be the case. We will soon see that there are additional problematic states, from which the learner cannot reach the target with probability one. Gibson and Wexler omit these states in their analysis.

4.5 Derivation of Transition Probabilities for the Markov TLA Structure

We have argued in the previous section, that the TLA working on finite parameter spaces reduces to a Markov chain. This argument cannot be complete without the precise computation of the transition probabilities from state to state. We do this now.

Consider, a parametric family with n boolean valued parameters. These define, 2^n grammars (and by extension, languages), as we have discussed. Let the target language L_t consist of the strings (sentences) $s_1, s_2, ...$, i.e.,

$$L_t = \{s_1, s_2, s_3, ...\} \subseteq \Sigma *$$

Let there be a probability distribution P on these strings[26], according to which they are drawn and presented to the learner. Suppose the learner is in a state s corresponding to the language L_s. Consider some other state k corresponding to the language L_k. What is the probability that the TLA will update its hypothesis from L_s to L_k after receiving the next example sentence? First, observe that due to the single valued constraint, if k and s differ by more than one parameter setting, then the probability of this transition is zero. As a matter of fact, the TLA will move from s to k only if the following two conditions are met, viz., 1)the next sentence it receives (say ω which occurs with probability $P(\omega)$) is analyzable by the parameter

[26] This is equivalent to assuming a noise-free situation, in the the sense that no sentence outside of the target language can occur. However, one could choose malicious distributions so that all strings from the target are not presented to the learner. If one wishes to include noise, one only need consider a distribution P on $\Sigma *$ rather than on the strings of L_t. Everything else in the derivation remains identical. This would yield a Markov chain corresponding to the TLA operating in the presence of noise. We study this situation in greater detail in the next chapter.

settings corresponding to k and not by the parameter setting corresponding to s, and 2)the TLA has a choice of n parameters to flip on not being able to analyze ω and it picks the one which would move it to state k.

Event 1 occurs with probability $\sum_{\omega \in L_k \setminus L_s} P(\omega)$ while event 2 occurs with probability $1/n$ since the parameter to flip is chosen uniformly at random out of the n possible choices. Thus the co-occurrence of both these events yields the following expression for the total probability of transition from s to k after one step:

$$P[s \rightarrow k] = \sum_{s_j \notin L_s, s_j \in L_k} (1/n) P(s_j)$$

Since the total probability over all the arcs out of s (including the self loop) must be 1, we obtain the probability of remaining in state s after one step as

$$P[s \rightarrow s] = 1 - \sum_{k \text{ is a neighboring state of } s} P[s \rightarrow k]$$

Finally, given any parameter space with n parameters, we have 2^n languages. Fixing one of them as the target language L_t we obtain the following procedure for constructing the corresponding Markov chain. Note that this will yield a Markov chain with the same topology (in the absence of noise) as the GW procedure in their paper. However, there is the significant difference of adding a probability measure on the language family.

- (Assign distribution) First fix a probability measure P on the strings of the target language L_t.

- (Enumerate states) Assign a state to each language i.e., each L_i.

- (Take set differences.) Now for any two states i and k, if they are more than 1 Hamming distance apart, then the transition $P[i \rightarrow k] = 0$. If they are 1 Hamming distance apart then $P[i \rightarrow k] = \frac{1}{n} P(L_k \setminus L_i$.

This model captures the dynamics of the TLA completely. We have indicated, in a previous footnote, how to extend the model to cover noise. In general, a class of memoryless algorithms can me modeled by a Markov chain. Appendix B of this chapter shows how to do this.

Example (continued).

Consider again the 3-parameter system in the previous figure with target language 5 (spec-first, comp-final, -V2; English). We can calculate the set differences between the languages (this is easily done for unembedded sentences using the data from Appendix A). Thereafter, assuming a distribution on the sentences of the target (uniform on degree-0 sentences), one could simply follow the procedure prescribed above, and obtain the transition probabilities which annotate the Markov chain of fig. 4-37.

For example, since the set difference between states 1 and 5 gives all of the target language, there is a (high) transition probability from state 1 to state 5. Similarly, since states 7 and 8 share some target language strings in common, such as S V, and do not share others, such as Adv S and S V O, the learner can move from state 7 to 8 and back again.

Many additional properties of the triggering learning system now become evident once the mathematical formalization has been given. It is easy to imagine other alternatives to the TLA that will avoid the local maxima problem. For example, as it stands, the learner only changes a parameter setting if that change allows the learner to analyze the sentence it could not analyze before. If we relax this condition so that in this situation the learner picks a parameter at random to change, then the problem with local maxima disappears, because there can be only 1 Absorbing State, namely the target grammar. All other states have exit arcs. Thus, by our main theorem, such a system *is* learnable.

Or consider, for example, the possibility of noise—that is, occasionally the learner gets strings that are not in the target language. GW state (fn. 4, p. 5) that this is not a problem; the learner need only pay attention to frequent data; *how* is the learner to "pay attention" to frequent data? Unless some kind of memory or frequency-counting device is added, the learner cannot know whether the example it receives is noise or not. If this is the case, then there is always some finite probability, however small, of escaping a local maximum. It appears that the identification in the limit framework as given is simply incompatible with the notion of noise, unless a memory window of some kind is added.

We may now proceed to ask the following questions about the TLA more precisely:

1. Does it converge?

2. How fast does it converge? How does this vary with distributional assumptions on the input examples?

3. Since our derivation is general, we can now compute the dynamics for other "natural" parameter systems, like the 10-parameter system for the acquisition of stress in languages developed by Dresher and Kaye (1990). What results do they yield?

4. Variants of TLA would correspond to other Markov structures. Do they converge? If so, how fast?

5. How does the convergence time scale up with the number of parameters?

6. What is the computational complexity of learning parameterized language families?

7. What happens if we move from on-line to batch learning? Can we get PAC-style bounds (Valiant, 1984)?

8. What does it mean to have non-stationary (nonergodic) Markov structures? How does this relate to assumptions about parameter ordering and maturation?

To explore these and other possible variations systematically, let us return to the 5-way classification scheme for learning models introduced at the beginning of this chapter. Recall that we have chosen a particular point in the 5-dimensional space for preliminary analysis. This, among other things, corresponds to an assumption of no noise, and a uniform probability distribution on the unembedded sentences of the target. We have shown how to model this particular learning problem by a Markov chain. This allows us to characterize learnability by our theorem earlier. We will soon see how to characterize the sample complexity of such a learning system.

In the next section, we discuss how to characterize the sample complexity of a learning system modeled as a Markov chain. Our eventual goal, however, is to explore more completely the space of fig. 4-36. We consider variations to our first learning problem along several dimensions. In particular, we discuss in turn, the effect on learnability and sample complexity of distributional assumptions on the data (question 2 above), and some variations in the learning algorithm (question 4). In the next chapter, we will consider the effect of noise, and how that can potentially bring about diachronic syntax change, as well as some alternate parameterizations (question 3).

4.6 Characterizing Convergence Times for the Markov Chain Model

The Markov chain formulation gives us some distinct advantages in theoretically characterizing the language acquisition problem. First, we have already seen how given a Markov Chain one could investigate whether or not it has exactly one absorbing state corresponding to the target grammar. This is equivalent to the question of whether any local maxima exist. One could also look at other issues (like stationarity or ergodicity assumptions) that might potentially affect convergence. Later we will consider several variants to TLA and analyze them formally within the Markov framework. We will also see that these variants do not suffer from the local maxima problem associated with GW's TLA.

Perhaps the significant advantage of the Markov chain formulation is that it allows us to also analyze convergence times. Given the transition matrix of a Markov chain, the problem of how long it takes to converge has been well studied. This question is of crucial importance in learnability. Following GW, we believe that it is not enough to show that the learning problem is *consistent* i.e., that the learner will converge to the target in the limit. We also need to show, as GW point out, that the learning problem is *feasible*, i.e., the learner will converge in "reasonable" time. This is particularly true in the case of finite parameter spaces where consistency might not be as much of a problem as feasibility. The Markov formulation allows us to attack the feasibility question. It also allows us to clarify the assumptions about the behavior of data and learner inherent in such an attack. We begin by considering a few ways in which one could formulate the question of convergence times.

4.6.1 Some Transition Matrices and Their Convergence Curves

Let us begin by following the procedure detailed in the previous section to actually obtain a few transition matrices. Consider the example which we looked at informally in the previous section. Here the target grammar was grammar 5 (according to our numbering of the languages in Appendix A). For simplicity, let us first assume a uniform distribution on the degree-0 strings in L_5, i.e., the probability the learner sees a particular string s_j in L_5 is 1/12 because there are 12 (degree-0) strings in L_5. We can now compute the transition matrix as the following, where 0's occupy matrix entries if not otherwise specified:

$$
\begin{array}{c|cccccccc}
 & L_1 & L_2 & L_3 & L_4 & L_5 & L_6 & L_7 & L_8 \\
\hline
L_1 & \frac{1}{2} & \frac{1}{6} & & & \frac{1}{3} & & & \\
L_2 & & 1 & & & & & & \\
L_3 & & & \frac{3}{4} & \frac{1}{12} & & & \frac{1}{6} & \\
L_4 & & \frac{1}{12} & & \frac{11}{12} & & & & \\
L_5 & & & & & 1 & & & \\
L_6 & & & & & \frac{1}{6} & \frac{5}{6} & & \\
L_7 & & & & & \frac{5}{18} & & \frac{2}{3} & \frac{1}{18} \\
L_8 & & & & & & \frac{1}{12} & \frac{1}{36} & \frac{8}{9}
\end{array}
$$

Notice that both 2 and 5 correspond to absorbing states; thus this chain suffers from the local maxima problem. Note also (following the previous figure as well) that state 4 only exits to either itself or to state 2, hence is also a local maximum. For a given transition matrix T, it is possible to compute

$$T_{\infty} = \lim_{m \to \infty} T^{m}.$$

If T is the transition probability matrix of a chain, then t_{ij}, i.e. the element of T in the ith row and jth column is the probability that the learner moves from state i to state j in one step. It is a well-known fact that if one considers the corresponding i,j element of T^{m} then this is the probability that the learner moves from state i to state j in m steps. Correspondingly, the i,jth element of T_{∞} is the probability of going from initial state i to state j "in the limit" as the number of examples goes to infinity. For learnability to hold irrespective of which state the learner starts in, the probability that the learner reaches state 5 should tend to 1 as m goes to infinity. This means that column 5 of T_{∞} should consist of 1's, and the matrix should contain 0's everywhere else. Actually we find that T^{m} converges to the following matrix as m goes to infinity:

$$
\begin{array}{c|cccccccc}
 & L_1 & L_2 & L_3 & L_4 & L_5 & L_6 & L_7 & L_8 \\
\hline
L_1 & & \frac{1}{3} & & & \frac{2}{3} & & & \\
L_2 & & 1 & & & & & & \\
L_3 & & \frac{1}{3} & & & \frac{2}{3} & & & \\
L_4 & & 1 & & & & & & \\
L_5 & & & & & 1 & & & \\
L_6 & & & & & 1 & & & \\
L_7 & & & & & 1 & & & \\
L_8 & & & & & 1 & & &
\end{array}
$$

Examining this matrix we see that if the learner starts out in states 2 or 4, it will certainly end up in state 2 in the limit. These two states correspond to local maxima grammars in the GW framework. If the learner starts in either of these two states, it will never reach the target. From the matrix we also see that if the learner starts in states 5 through 8, it will certainly converge in the limit to the target grammar.

The situation regarding states 1 and 3 is more interesting, and not covered in Gibson and Wexler (1994). If the learner starts in either of these states, it will reach the target grammar with probability 2/3 and reach state 2, the other absorbing state with probability 1/3. Thus we see that local maxima (states unconnected to the target) are *not* the only problem for learnability. As a consequence of our stochastic formulation, we see that there are initial hypotheses from which triggered paths exist to the target, however the learner will not take these paths with probability one. In our case, because of the uniform distribution assumption, we see that the path to the target will only be taken with probability 2/3. By making the distribution more favorable, this probability can be made larger, but it can never be made one.

This analysis, motivated as it was by our information-theoretic perspective, considerably increases the number of problematic initial states from that presented in Gibson and Wexler. While the broader implications of this is not clear, it certainly renders moot some of the linguistic[27] implications of GW's analysis.

Obviously one can examine other details of this particular system. However, let us now look at a case where there is no local maxima problem. This is the case when the target languages have verb-second (V2) movement in GW's 3-parameter case.

[27] For example, GW rely on "connectedness" to obtain their list of local maxima. From this (incorrect) list, noticing that all local maxima were +Verb Second (+V2), they argued for ordered parameter acquisition or "maturation". In other words, they claimed that the V2 parameter was more crucial, and had to be set earlier in the child's language acquisition process. Our analysis shows that this is incorrect, an example of how computational analysis can aid the search for adequate linguistic theories.

Consider the transition matrix (shown below) obtained when the target language is L_1. Again we assume a uniform distribution on strings of the target.

$$\begin{array}{c|cccccccc}
 & L_1 & L_2 & L_3 & L_4 & L_5 & L_6 & L_7 & L_8 \\
\hline
L_1 & 1 & & & & & & & \\
L_2 & \frac{1}{6} & \frac{5}{6} & & & & & & \\
L_3 & \frac{5}{18} & & \frac{2}{3} & \frac{1}{18} & & & & \\
L_4 & & \frac{3}{36} & \frac{1}{36} & \frac{8}{9} & & & & \\
L_5 & \frac{1}{3} & & & & \frac{23}{36} & \frac{1}{36} & & \\
L_6 & & \frac{5}{36} & & & & \frac{31}{36} & & \\
L_7 & & & \frac{1}{18} & & & & \frac{11}{12} & \frac{1}{36} \\
L_8 & & & & \frac{1}{18} & & & & \frac{17}{18}
\end{array}$$

Here we find that T^m does indeed converge to a matrix with 1's in the first column and 0's elsewhere. Consider the first column of T^m. It is of the form:

$$(p_1(m), p_2(m), p_3(m), p_4(m), p_5(m), p_6(m), p_7(m), p_8(m))'$$

Here $p_i(m)$ denotes the probability of being in state 1 at the end of m examples in the case where the learner started in state i. Naturally we want

$$\lim_{m \to \infty} p_i(m) = 1$$

and for this example this is indeed the case. Fig. ?? shows a plot of the following quantity as a function of m, the number of examples.

$$p(m) = \min_i \{p_i(m)\}$$

The quantity $p(m)$ is easy to interpret. Thus $p(m) = 0.95$ means that for every initial state of the learner the probability that it is in the target state after m examples is at least 0.95. Further there is one initial state (the worst initial state with respect to the target, which in our example is L_8) for which this probability is exactly 0.95. We find on looking at the curve that the learner converges with high probability within 100 to 200 (degree-0) example sentences, a psychologically plausible number. (One can now of course proceed to examine actual transcripts of child input to calculate convergence times for "actual" distributions of examples, and we are currently engaged in this effort.)

Now that we have made a first attempt to quantify the convergence time, several other questions can be raised. How does convergence time depend upon the distribution of the data? How does it compare with other kinds of Markov structures with

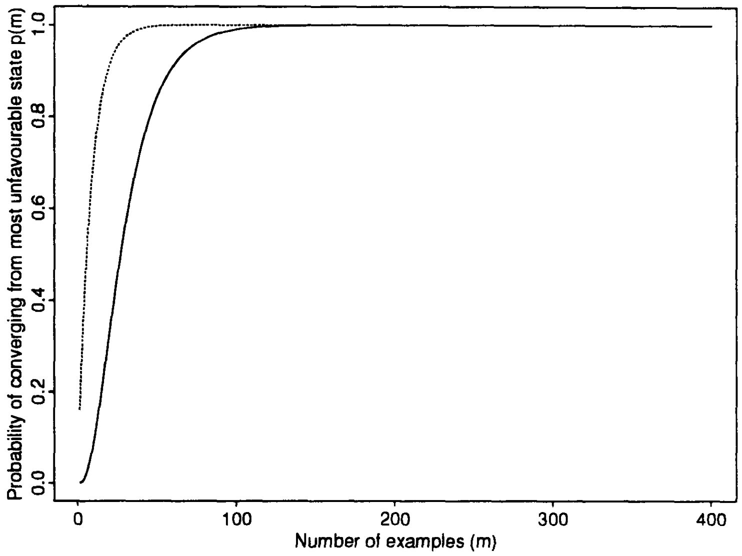

Figure 4-38: Convergence as function of number of examples. The horizontal axis denotes the number of examples received and the vertical axis represents the probability of converging to the target state. The data from the target is assumed to be distributed uniformly over degree-0 sentences. The solid line represents TLA convergence times and the dotted line is a random walk learning algorithm (RWA). Note that random walk actually converges *faster* than the TLA in this case.

the same number of states? How will the convergence time be affected if the number of states increases, i.e the number of parameters increases? How does it depend upon the way in which the parameters relate to the surface strings? Are there other ways to characterize convergence times? We now proceed to answer some of these questions.

4.6.2 Absorption Times

In the previous section, we computed the transition matrix for a fixed (in principle, this could be arbitrary) distribution and showed the rate of convergence in a certain way. In particular, we plotted $p(m)$, (the probability of converging from the most unfavorable initial state) against m (the number of samples). However, this is not the only way to characterize convergence times. Given an initial state, the time taken to reach the absorption state (known as the absorption time) is a random variable. One can compute the mean and variance of this random variable. For the case when the target language is L_1, we have seen that the transition matrix has the form:

$$T = \begin{pmatrix} 1 & 0 \\ R & Q \end{pmatrix}$$

Here Q is a 7-dimensional square matrix. The mean absorption times from states 2 through 8 is given by the vector (see Isaacson and Madsen (1976))

$$\mu = (I - Q)^{-1}\mathbf{1}$$

where $\mathbf{1}$ is a 7-dimensional column vector of ones. The vector of second moments is given by

$$\mu' = (I - Q)^{-1}(2\mu - \mathbf{1}).$$

Using this result, we can now compute the mean and standard deviation of the absorption time from the most unfavorable initial state of the learner. (We note that the second moment is fairly skewed in such cases and so is not symmetric about the mean, as may be seen from the previous curves.) The four learning scenarios considered are the TLA with uniform, and increasingly malicious distributions (discussed later), and the random walk (also discussed later).

Learning scenario	Mean abs. time	St. Dev. of abs. time
TLA (uniform)	34.8	22.3
TLA ($a = 0.99$)	45000	33000
TLA ($a = 0.9999$)	4.5×10^6	3.3×10^6
RW	9.6	10.1

4.6.3 Eigenvalue Rates of Convergence

In classical Markov chain theory, there are also well-known convergence theorems derived from a consideration of the eigenvalues of the transition matrix. We state without proof a convergence result for transition matrices stated in terms of its eigenvalues.

Theorem 4.6.1 *Let T be an $n \times n$ transition matrix with n linearly independent left eigenvectors $\mathbf{x}_1, \ldots \mathbf{x}_2$ corresponding to eigenvalues $\lambda_1, \ldots, \lambda_n$. Let $\mathbf{x}_0$ (an n-dimensional vector) represent the starting probability of being in each state of the chain and π be the limiting probability of being in each state. Then after k transitions, the probability of being in each state $\mathbf{x}_0 T^k$ can be described by*

$$\| \mathbf{x}_0 T^k - \pi \| = \| \sum_{i=1}^{n} \lambda_i^k \mathbf{x}_0 \mathbf{y}_i \mathbf{x}_i \| \le \max_{2 \le i \le n} |\lambda_i|^k \sum_{i=2}^{n} \| \mathbf{x}_0 \mathbf{y}_i \mathbf{x}_i \|$$

where the $\mathbf{y}_i$'s are the right eigenvectors of T.

This theorem thus bounds the rate of convergence to the limiting distribution π (in cases where there is only one absorption state, π will have a 1 corresponding to that state and 0 everywhere else). Using this result we can now bound the rates of convergence (in terms of number, k, of samples) by:

Learning scenario	Rate of Convergence
TLA (uniform)	$O(0.94^k)$
TLA($a = 0.99$)	$O((1 - 10^{-4})^k)$
TLA($a = 0.9999$)	$O((1 - 10^{-6})^k)$
RW	$O(0.89^k)$

This theorem also helps us to see the connection between the number of examples and the number of parameters since a chain with n states (corresponding to an $n \times n$ transition matrix) represents a language family with $\log_2(n)$ parameters.

4.7 Exploring Other Points

We have developed, by now, a complete set of tools to characterize learnability and sample complexity of memoryless algorithms working on finite parameter spaces. We applied these tools to a specific learning problem which corresponded to a point in our 5-dimensional space previously investigated by Gibson and Wexler. We also provided an account of how our new analysis revised some of their conclusions and had possible applications to linguistic theory. Here we now explore some other points in the space. In the next section, we consider varying the learning algorithm, while keeping other assumptions about the learning problem identical to that before. Later, we vary the distribution of the data.

4.7.1 Changing the Algorithm

As one example of the power of this approach, we can compare the convergence time of TLA to other algorithms. TLA observes the single value and greediness constraints. We consider the following three simple variants by dropping either or both of the Single Value and Greediness constraints:

Random walk with neither greediness nor single value constraints: We have already seen this example before. The learner is in a particular state. Upon receiving a new sentence, it remains in that state if the sentence is analyzable. If not, the learner moves uniformly at random to any of the other states and stays there waiting for the next sentence. This is done without regard to whether the new state allows the sentence to be analyzed.

Random walk with no greediness but with single value constraint: The learner remains in its original state if the new sentence is analyzable. Otherwise, the learner chooses one of the parameters uniformly at random and flips it thereby moving to an adjacent state in the Markov structure. Again this is done without regard to whether the new state allows the sentence to be analyzed. However since only one parameter is changed at a time, the learner can only move to neighboring states at any given time.

Random walk with no single value constraint but with greediness: The learner remains in its original state if the new sentence is analyzable. Otherwise the learner moves uniformly at random to any of the other states and stays there iff the

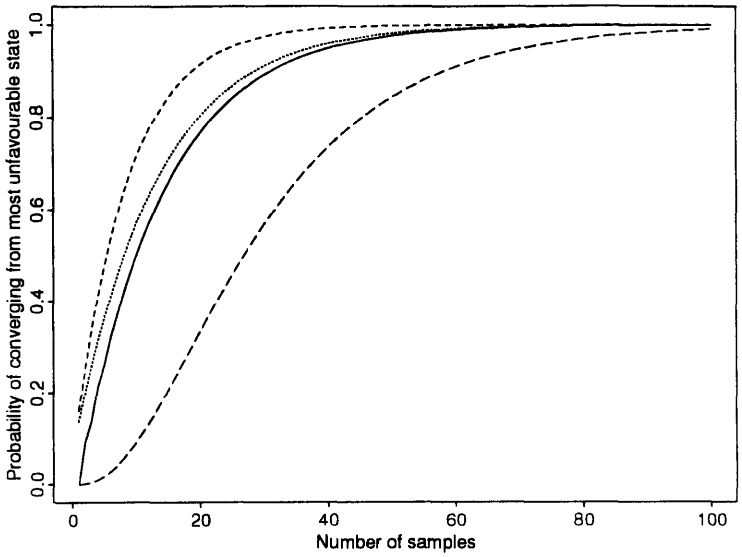

Figure 4-39: Convergence rates for different learning algorithms when L_1 is the target language. The curve with the slowest rate (large dashes) represents the TLA. The curve with the fastest rate (small dashes) is the Random Walk (RWA) with no greediness or single value constraints. Random walks with exactly one of the greediness and single value constraints have performances in between these two and are very close to each other.

sentence can be analyzed. If the sentence cannot be analyzed in the new state the learner remains in its original state.

Fig. 4-39 shows the convergence times for these three algorithms when L_1 is the target language. Interestingly, all three perform better than the TLA for this task (learning the language L_1). More generally, it is found that the variants converge faster than the TLA for every target language. Further, they do not suffer from local maxima problems. In other words, the class of languages is not learnable by the TLA, but is by its variants. This is another striking consequence of our analysis. The TLA seems to be the "most preferred algorithm" by psychologists. The failure of the TLA to learn the 3-parameter space was used to argue for maturational theories, alternate parameterizations, and parameter ordering.

In view of the fact that the failure of the TLA can be corrected by fairly simple alterations[28], one should examine the conceptual support (from psychologists) for the TLA more closely before drawing any serious linguistic implications. This remains

[28] Note that we have barely scraped the tip of the iceberg as far as exploring the space of possible algorithms is concerned.

yet another example of how the computational perspective can allow us to rethink cognitive assumptions. Of course, it may be that the TLA has empirical support, in the sense of independent evidence that children do use this procedure (given by the pattern of their errors, etc.), but this evidence is lacking, as far as we know.

4.7.2 Distributional Assumptions

In an earlier section we assumed that the data was uniformly distributed. We computed the transition matrix for a particular target language and showed that convergence times were of the order of 100-200 samples. In this section we show that the convergence times depend crucially upon the distribution. In particular we can choose a distribution that will make the convergence time as large as we want. Thus the distribution-free convergence time for the 3-parameter system is infinite.

As before, we consider the situation where the target language is L_1. There are no local maxima problems for this choice. We begin by letting the distribution be parameterized by the variables a, b, c, d where

$$\begin{aligned}
a &= P(A = \{\text{Adv V S}\}) \\
b &= P(B = \{\text{Adv V O S, Adv Aux V S}\}) \\
c &= P(C = \{\text{Adv V O1 O2 S, Adv Aux V O S,} \\
& \quad \text{Adv Aux V O1 O2 S}\}) \\
d &= P(D = \{\text{V S}\})
\end{aligned}$$

Thus each of the sets A, B, C and D contain different degree-0 sentences of L_1. Clearly the probability of the set $L_1 \backslash \{A \cup B \cup C \cup D\}$ is $1-(a+b+c+d)$. The elements of each defined subset of L_1 are equally likely with respect to each other. Setting positive values for a, b, c, d such that $a+b+c+d < 1$ now defines a unique probability for each degree(0) sentence in L_1. For example, the probability of (Adv V O S) is $b/2$, the probability of (Adv Aux V O S) is $c/3$, that of (V O S) is $(1-(a+b+c+d))/6$ and so on.

We can now obtain the transition matrix corresponding to this distribution. This is shown in Table 4.2.

Compare this matrix with that obtained with a uniform distribution on the sentences of L_1 in the earlier section. This matrix has non-zero elements (transition probabilities) exactly where the earlier matrix had non-zero elements. However, the value of each transition probability now depends upon a, b, c, and d. In particular if we choose $a = 1/12, b = 2/12, c = 3/12, d = 1/12$ (this is equivalent to assuming a uniform distribution) we obtain the appropriate transition matrix as before. Looking

	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8
L_1	1							
L_2	$\frac{1-a-b-c}{3}$	$\frac{2+a+b+c}{3}$						
L_3	$\frac{1-a-d}{3}$		$\frac{2+a+d-b}{3}$	$\frac{b}{3}$				
L_4		$\frac{c}{3}$	$\frac{d}{3}$	$\frac{3-c-d}{3}$				
L_5	$\frac{1}{3}$				$\frac{2-a}{3}$	$\frac{a}{3}$		
L_6		$\frac{b+c}{3}$				$\frac{3-b-c}{3}$		
L_7			$\frac{a+d}{3}$				$\frac{3-2a-d}{3}$	$\frac{a}{3}$
L_8				$\frac{b}{3}$				$\frac{3-b}{3}$

Table 4.2: Transition matrix corresponding to a parameterized choice for the distribution on the target strings. In this case the target is L_1 and the distribution is parameterized according to Section 4.7.2

more closely at the general transition matrix, we see that the transition probability from state 2 to state 1 is $(1-(a+b+c))/3$. Clearly if we make a arbitrarily close to 1, then this transition probability is arbitrarily close to 0 so that the number of samples needed to converge can be made arbitrarily large. Thus choosing large values for a and small values for b will result in large convergence times.

This means that the sample complexity cannot be bounded in a distribution-free sense, because by choosing a highly unfavorable distribution the sample complexity can be made as high as possible. For example, we now give the convergence curves calculated for different choices of a, b, c, d. We see that for a uniform distribution the convergence occurs within 200 samples. By choosing a distribution with $a = 0.9999$ and $b = c = d = 0.000001$, the convergence time can be pushed up to as much as 50 million samples. (Of course, this distribution is presumably not psychologically realistic.) For $a = 0.99, b = c = d = 0.0001$, the sample complexity is on the order of $100,000$ positive examples.

4.7.3 Natural Distributions–CHILDES CORPUS

It is of interest to examine the fidelity of the model using real language distributions, namely, the CHILDES database. We have carried out preliminary direct experiments using the CHILDES caretaker English input to "Nina" and German input to "Katrin"; these consist of 43,612 and 632 sentences each, respectively. We note, following well-known results by psycholinguists, that both corpuses contain a much higher percentage of aux-inversion and wh-questions than "ordinary" text (e.g., the LOB): 25,890 questions, and 11, 775 wh-questions; 201 and 99 in the German corpus; but only 2,506 questions or 3.7% out of 53,495 LOB sentences.

To test convergence, an implemented system using a newer version of deMarcken's

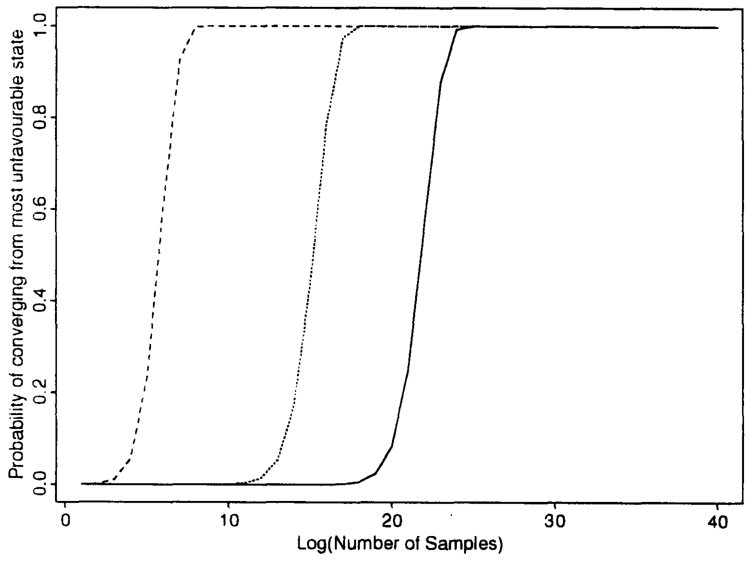

Figure 4-40: Rates of convergence for TLA with L_1 as the target language for different distributions. The y-axis plots the probability of converging to the target after m samples and the x-axis is on a log scale, i.e., it shows $\log(m)$ as m varies. The solid line denotes the choice of an "unfavorable" distribution characterized by $a = 0.9999; b = c = d = 0.000001$. The dotted line denotes the choice of $a = 0.99; b = c = d = 0.0001$ and the dashed line is the convergence curve for a uniform distribution, the same curve as plotted in the earlier figure.

partial parser (see deMarcken, 1990) analyzed each degree-0 or degree-1 sentence as falling into one of the input patterns SVO, S Aux V, etc., as appropriate for the target language. Sentences not parsable into these patterns were discarded (presumably "too complex" in some sense following a tradition established by many other researchers; see Wexler and Culicover (1980) for details). Some examples of caretaker inputs follow:

> this is a book ? what do you see in the book ?
>
> how many rabbits ?
>
> what is the rabbit doing ? (...)
>
> is he hopping ? oh . and what is he playing with ?
>
> red mir doch nicht alles nach !
>
> ja , die schwätzen auch immer alles nach (...)

When run through the TLA, we discover that convergence falls roughly along the TLA convergence time displayed in figure 1–roughly 100 examples to asymptote. Thus, the feasibility of the basic model is confirmed by actual caretaker input, at least in this simple case, for both English and German. We are continuing to explore this model with other languages and distributional assumptions. However, there is one very important new complication that must be taken into account: we have found that one must (obviously) add patterns to cover the predominance of auxiliary inversions and wh-questions. However, that largely begs the question of whether the language is verb-second or not. Thus, as far as we can tell, we have not yet arrived at a satisfactory parameter-setting account for V2 acquisition.

4.8 Batch Learning Upper and Lower Bounds: An Aside

So far we have discussed a memoryless learner moving from state to state in parameter space and hopefully converging to the correct target in finite time. As we saw this was well-modeled by our Markov formulation. In this section however we step back and consider upper and lower bounds for learning finite language families if the learner was allowed to remember all the strings encountered and optimize over them. Needless to say this might not be a psychologically plausible assumption, but it can shed light on the information-theoretic complexity of the learning problem.

Consider a situation where there are n languages $L_1, L_2, \ldots L_n$ over an alphabet Σ. Each language can be represented as a subset of Σ^* i.e.

$$L_i = \{\omega_{i1}, \omega_{i2}, \ldots\}; \omega_j \in \Sigma^*$$

The learner is provided with positive data (strings that belong to the language) drawn according to distribution P on the strings of a particular target language. The learner is to identify the target. It is quite possible that the learner receives strings that are in more than one language. In such a case the learner will not be able to uniquely identify the target. However, as more and more data becomes available, the probability of having received only ambigious strings becomes smaller and smaller and eventually the learner will be able to identify the target uniquely. An interesting question to ask then is how many samples does the learner need to see so that with high confidence it is able to identify the target, i.e. the probability that after seeing that many samples, the learner is still ambigious about the target is less than δ. The following theorem provides a lower bound.

Theorem 4.8.1 *The learner needs to draw at least* $M = \max_{j \neq t} \frac{1}{\ln(1/p_j)} \ln(1/\delta)$ *samples (where* $p_j = P(L_t \cap L_j)$*) in order to be able to identify the target with confidence greater than* $1 - \delta$.

Proof. Suppose the learner draws m (less than M) samples. Let $k = \arg\max_{j \neq t} p_j$. This means 1) $M = \frac{1}{\ln(1/p_k)} \ln(1/\delta)$ and 2) that with probability p_k the learner receives a string which is in both L_k and L_t. Hence it will be unable to discriminate between the target and the kth language. After drawing m samples, the probability that all of them belong to the set $L_t \cap L_k$ is $(p_k)^m$. In such a case even after seeing m samples, the learner will be in an ambiguous state. Now $(p_k)^m > (p_k)^M$ since $m < M$ and $p_k < 1$. Finally since $M \ln(1/p_k) = \ln((1/p_k)^M) = \ln(1/\delta)$, we see that $(p_k)^m > \delta$. Thus the probability of being ambiguous after m examples is greater than δ which means that the confidence of being able to identify the target is less than $1 - \delta$. ∎

This simple result allows us to assess the number of samples we need to draw in order to be confident of correctly identifying the target. Note that if the distribution of the data is very unfavorable, that is, the probability of receiving ambiguous strings is quite high, then the number of samples needed can actually be quite large. While the previous theorem provides the number of samples *necessary* to identify the target, the following theorem provides an upper bound for the number of samples that are *sufficient* to guarantee identification with high confidence.

Theorem 4.8.2 *If the learner draws more than* $M = \frac{1}{\ln(1/(1-b_t))} \ln(1/\delta)$ *samples, then it will identify the target with confidence greater than* $1-\delta$. *(Here* $b_t = P(L_t \backslash \cup_{j \neq t} L_j)$*).*

Proof. Consider the set $L = L_t \setminus \cup_{j \neq t} L_j$. Any element of this set is present in the target language L_t but not in any other language. Consequently upon receiving such a string, the learner will be able to instantly identify the target. After $m > M$ samples, the probability that the learner has not received any member of this set is $(1 - P(L))^m = (1 - b_t)^m < (1 - b_t)^M = \delta$. Hence the probability of seeing some member of L in those m samples is greater than $1 - \delta$. But seeing such a member enables the learner to identify the target so the probability that the learner is able to identify the target is greater than $1 - \delta$ if it draws more than M samples. ∎

To summarize, this section provides a simple upper and lower bound on the sample complexity of exact identification of the target language from positive data. The δ parameter that measures the confidence of the learner of being able to identify the target is *suggestive* of a PAC [123] formulation. However there is a crucial difference. In the PAC formulation, one is interested in an ϵ-approximation to the target language with at least $1 - \delta$ confidence. In our case, this is not so. Since we are not allowed to approximate the target, the sample complexity shoots up with choice of unfavorable distributions. There are some interesting directions one could follow within this batch learning framework. One could try to get true PAC-style distribution-free bounds for various kinds of language families. Alternatively one could use the exact identification results here for linguistically plausible language families with "reasonable" probability distributions on the data. It might be an interesting exercise to recompute the bounds for cases where the learner receives both positive and negative data. Finally the bounds obtained here could be sharpened further. We intend to look into some of these questions in the future.

4.9 Conclusions, Open Questions, and Future Directions

The problem of learning parameterized families of grammars has several different dimensions as we have emphasized earlier. One needs to investigate the learnability for a variety of algorithms, distributional assumptions, parameterizations, and so on. In this chapter, we have emphasized that it is not enough to merely check for learnability in the limit (as previous research within an inductive inference Gold framework has tended to do; see, for example, Osherson and Weinstein, 1986); one also needs to quantify the sample complexity of the learning problem, i.e., how many examples does the learning algorithm need to see in order to be able to identify the target grammar with high confidence. To illustrate the importance of this, we re-analyzed a particular learning problem previously studied by Gibson and Wexler.

Our reanalysis, shows that on finite parameter spaces, the Triggering Learning Algorithm in particular, and memoryless algorithms in general, can be completely modeled by a Markov process. This Markov model then allows us to check for learnability in a very simple fashion, rather than the more complicated procedures previously used in the linguistics community. Further, it also allows us to characterize the sample complexity of learning with such algorithms. On studying the performance of the TLA on the specific 3-parameter subspace from this perspective, we found several new results. First, the existence of new problematic initial hypotheses was discovered—leading to revisions of certain aspects of maturation and parameter ordering suggested by Gibson and Wexler. Second, we showed that the existence of local triggers (in other words, a triggered path from the initial hypothesis to the target) is not sufficient to guarantee learnability. Third, we found that the TLA was suboptimal; for example the random walk algorithm on this space had no local maxima and converged faster.

This analysis on a simple, previously studied, example demonstrates the usefulness of our perspective. It should be reiterated that any finite parameterization, and a class of memoryless algorithms can be studied by this approach. There are several important questions which need to be pursued further. For example, one could turn to other natural parametric systems suggested (the example of metrical phonology given in this chapter, a variant studied by Dresher and Kaye (1990), a parameterization chosen by Clark and Roberts (1993)) and so on. One could then establish the complexity of learning these other parametric schemes, possibly with useful results again.

Another crucial direction relates to the learning algorithm used. What happens when the learner is allowed the use of memory? An interesting investigation of this issue has been done by Kapur (1992). However some questions remain unresolved. For example, is it true that any algorithm with a finite memory size (n examples, say) can be modeled as a finite order Markov chain (presumably, the order would be related to n, in some sense)? Is this a useful way to characterize such algorithms? A complete characterization of human language requires us to describe the linguistic knowledge (equivalent to parameterization), and the algorithm children use to acquire this knowledge. Insights about the kinds of algorithms available, and their psychological feasibility, could often direct the search for the right kind of linguistic knowledge.

It is also of interest to study the relationship between the expressive power of the parameterized family of grammars and the number of parameters. One needs to reiterate, here, the importance of our point of view in this thesis. Recall how in Chapter 2, we investigated regularization networks from an approximation and estimation point of view. Grammars, are no different from regularization networks in this sense. Thus, one could pose the following general problem. Assume a class of grammars G as the concept class, and a parameterized class of grammars H_n as the hypothesis class. Now, for a target grammar $g \in G$, how many example sentences need to be drawn, and how large must the number of parameters, n, be, so that the learner's hypothesis will be close to the target with high confidence?

Yet another issue has to do with the "smoothness" relation between the parameter settings and the resulting surface strings. In principles-and-parameters theory, it has often been suggested that a small parameter change could lead to a large deductive change in the grammar, hence a large change in the surface language generated. In all the examples considered so far there is a smooth relation between surface sentences and parameters, in that switching from a V2 to a non-V2 system, for instance, leads us to a Markov state that is not too far away from the previous one. If this is not so, it is not so clear that the TLA will work as before. In fact, the whole question of how to formulate the notion of "smoothness" in a language–grammar framework is unclear. We know in the case of continuous functions, as discussed in Chapter 3, that if the learner is allowed to choose examples (which can be simulated by selective attention), then such an "active" learner can approximate such functions much more quickly than a "passive" learner, like the one presented in GW. Is there an analog to this in the discrete, digital domain of language? Further, how can one approximate a language? Here too mathematics may play a helpful role. Recall that there is an analog to a functional analysis of languages—namely, the algebraic approach advanced by Chomsky and Schutzenberger (1963). In this model, a language is described by an (infinite) polynomial generating function, where the coefficients on the polynomial term x gives the number of ways of deriving the string x. A (weak, string) approximation to a language can then be defined in terms of an approximation to the generating function. If this method can be deployed, then one might be able to carry over the results of functional analysis and approximation for active vs. passive learners into the "digital" domain of language. If this is possible, we would then have a very powerful set of previously underutilized mathematical tools to analyze language learnability.

Appendix

4-A Unembedded Sentences For Parametric Grammars

The following table provides the unembedded (degree-0) sentences from each of the 8 grammars (languages) obtained by setting the 3 parameters of example 1 to different values. The languages are referred to as L_1 through L_8.

Language	Spec	Comp	V2	Degree-0 unembedded sentences
L_1	1	1	0	"v s" "v o s" "v o1 o2 s" "AUX v s" "AUX v o s" "AUX v o1 o2 s" "ADV v s" "ADV v o s" "ADV v o1 o2 s" "ADV AUX v s" "ADV AUX v o s" "ADV AUX v o1 o2 s"
L_2	1	1	1	"s v" "s v o" "o v s" "s v o1 o2" "o1 v o2 s" "o2 v o1 s" "s AUX v" "s AUX v o" "o AUX v s" "s AUX v o1 o2" "o1 AUX v o2 s" "o2 AUX v o1 s" "ADV v s" "ADV v o s" "ADV v o1 o2 s" "ADV AUX v s" "ADV AUX v o s" "ADV AUX v o1 o2 s"
L_3	1	0	0	"v s" "o v s" "o2 o1 v s" "v AUX s" "o v AUX s" "o2 o1 v AUX s" "ADV v s" "ADV o v s" "ADV o2 o1 v s" "ADV v AUX s" "ADV o v AUX s" "ADV o2 o1 v AUX s"
L_4	1	0	1	"s v" "o v s" "s v o" "s v o2 o1" "o1 v o2 s" "o2 v o1 s" "s AUX v" "s AUX o v" "o AUX v s" "s AUX o2 o1 v" "o1 AUX o2 v s" "o2 AUX o1 v s" "ADV v s" "ADV v o s" "ADV v o2 o1 s" "ADV AUX v s" "ADV AUX o v s" "ADV AUX o2 o1 v s"
L_5 (English, French)	0	1	0	"s v" "s v o" "s v o1 o2" "s AUX v" "s AUX v o" "s AUX v o1 o2" "ADV s v" "ADV s v o" "ADV s v o1 o2" "ADV s AUX v" "ADV s AUX v o" "ADV s AUX v o1 o2"
L_6	0	1	1	"s v" "s v o" "o v s" "s v o1 o2" "o1 v s o2" "o2 v s o1" "s AUX v" "s AUX v o" "o AUX s v" "s AUX v o1 o2" "o1 AUX s v o2" "o2 AUX s v o1" "ADV v s" "ADV v s o" "ADV v s o1 o2" "ADV AUX s v" "ADV AUX s v o" "ADV AUX s v o1 o2"
L_7 (Bengali, Hindi)	0	0	0	"s v" "s o v" "s o2 o1 v" "s v AUX" "s o2 o1 v AUX" "ADV s v" "ADV s o v" "ADV s o2 o1 v" "ADV s v AUX" "ADV s o v AUX" "ADV s o2 o1 v AUX"
L_8 (German, Dutch)	0	0	1	"s v" "s v o" "o v s" "s v o2 o1" "o1 v s o2" "o2 v s o1" "s AUX v" "s AUX o v" "o AUX s v" "o1 AUX s o2 v" "o2 AUX s o1 v" "ADV v s" "ADV v s o" "ADV v s o2 o1" "ADV AUX s v" "ADV AUX s o v" "s AUX o2 o1 v" "s o v AUX" "ADV AUX s o2 o1 v"

4-B Memoryless Algorithms and Markov Chains

Memoryless algorithms can be regarded as those which have no recollection of previous data, or previous inferences made about the target function. At any point in time, the only information upon which such an algorithm acts is the current data, and the current hypothesis (state). A memoryless algorithm can then be regarded as an effective procedure mapping this information to a new hypothesis. In general. given a particular hypothesis state (h in $\mathcal{H}$, the hypothesis space), and a new datum (sentence, s in $\Sigma*$), such a memoryless algorithm will map onto a new hypothesis ($g \in \mathcal{H}$). Ofcourse, g could be the same as h or it could be different depending upon the specifics of the algorithm and the datum. If one includes the possibility of randomization, then the mapping need not be deterministic. In other words, given a state h, and sentence s, the algorithm maps onto a distribution $P_{\mathcal{H}}$ over the hypothesis space, according to which the new state is selected. Clearly,

$$\sum_{h \in \mathcal{H}} P_{\mathcal{H}}(h) = 1$$

Let $\mathcal{P}$ be the set of all possible probability distributions over the (finite) hypothesis space. For any $P_{\mathcal{H}} \in \mathcal{P}$, thus, $P_{\mathcal{H}}[h]$ is the probability measure on the hypothesis (state) h.

A memoryless algorithm can then be regarded as a computable function (f) from $(\mathcal{H}, \Sigma*)$ to $\mathcal{P}$ as follows:

$$f : (\mathcal{H}. \Sigma*) \longrightarrow \mathcal{P}$$

Thus, for any $h \in \mathcal{H}$, and $s \in \Sigma*$. the quantity $f(h, s)$ is a distribution over the hypothesis space according to which the learner would pick the next hypothesis. Consequently, a learner following such an algorithm, would update its hypothesis with each new sentence, and move from state to state in our finite parameter space of hypotheses. Suppose, at a point in time, the learner is in a state h_1. What is the probability that it will move to state h_2 after the next example? It will do so only if the following two conditions are met. First, it receives a sentence (example), s, for which $f(h_1, s)$ has a non-zero probability measure on the state h_2. Let this probability measure be $f(h_1, s)[h_2]$. Second, given the probability over the hypothesis space according to which it chooses the next hypothesis, the learner actually ends up choosing h_2 as the next hypothesis.

Given a distribution P on $\Sigma*$, according to which sentences are drawn, and presented to the learner. the transition probability from h_1 to h_2 is now given by:

$$Prob[h_1 \rightarrow h_2] = \sum_{\{s|f(h_1,s)[h_2]>0\}} f(h_1,s)[h_2]P(s)$$

Having obtained the transition probabilities, it is clear that the memoryless algorithm is a Markov chain.

Chapter 5

The Logical Problem of Language Change

Abstract

In this chapter, we consider the problem of language change. Linguists have to explain not only how languages are learned (a problem we investigated in the previous chapter), but also how and why they have evolved in certain trajectories. While the language learning problem has concentrated on the behavior of the individual child, and how it acquires a particular grammar (from a class of grammars $\mathcal{G}$), we consider, in this chapter, a population of such child learners, and investigate the *emergent*, global, population characteristics of the linguistic community over several generations. We argue that language change is the logical consequence of specific assumptions about grammatical theories, and learning paradigms. In particular, we are able to transform the parameterized theories, and memoryless algorithms of the previous chapter into grammatical dynamical systems, whose evolution depicts the evolving linguistic composition of the population. We investigate the linguistic, and computational consequences of this fact. From a more programmatic perspective, we lay a possible logical framework for the scientific study of historical linguistics, and introduce thereby, a formal *diachronic* criterion for adequacy of linguistic theories.

5.1 Introduction

As is well known, languages change over time. Language scientists have long been occupied with describing language changes in phonology, syntax, and semantics. There have been many descriptive and a few explanatory accounts of language change, including some explicit computational models. Many authors appeal naturally to the analogy between language change and another familiar model of change, namely, biological evolution. There is also a notion that language systems are adaptive (dynamical) ones. For instance, Lightfoot (1991, chapter 7, pages 163–65ff.) talks about language change in this way:

> Some general properties of language change are shared by other dynamic systems in the natural world...

Indeed, entire books have been devoted to the description of language change using the terminology of population biology: genetic drift, clines, etc. (UCLA book on language diversity in space and time).

However, these analogies have rarely been pursued beyond casual and descriptive accounts.[29] In this paper we would like to formalize these linguists' intuitive notions in a specific way as a concrete computational model, and investigate the consequences of this formalization. In particular, we show that a model of language change emerges as a logical consequence of language learnability, a point made by Lightfoot (1991). We shall see that Lightfoot's intuition that languages could behave just as though they were dynamical systems is essentially correct, and we can provide concrete examples of both "gradual" and "sudden" syntactic changes occuring over time periods of many generations to just a single generation.[30]

Not surprisingly, many other interesting points emerge from the formalization, some programmatic in nature:

- We provide a general procedure for deriving a dynamical systems model from grammatical theories and learning paradigms.

- *Learnability* is a well-known criterion for testing the adequacy of grammatical theories. With our new model, we can now give an *evolutionary* criterion. By this we mean that by comparing the evolutionary trajectories of derived dynamical linguistic systems to historically observed trajectories, one can determine the adequacy of linguistic theories or learning algorithms.

- We explicitly derive dynamical systems corresponding to parameterized linguistic theories (e.g. Head First/Final parameter in HPSG or GB grammars) and memoryless language learning algorithms (e.g. gradient ascent in parameter space).

- Concretely, we illustrate the use of dynamical systems as a research tool by considering the loss of Verb Second position in Old French as compared to Modern French. We demonstrate that, when mathematically modeled by our system, one grammatical parameterization in the literature does not seem to permit this historical change, while another does. We are also able to more accurately model the time course of language change. In particular, in contrast to Kroch (1989) and others, who mimic population biology models by imposing

[29] Some notable exceptions are Kroch (1990), Clark and Roberts (1993).

[30] Lightfoot 1991 refers to these sudden changes, acting over 1 generation, as "catastrophic" but in fact this term usually has a different sense in the dynamical systems literature.

an S-shaped logistic change by *assumption*, we show that the time course of language change need not be S-shaped. Rather, language-change envelopes are ***derivable*** from more fundamental properties of dynamical systems; sometimes they are S-shaped, but they can also have a nonmonotonic shape, or even non-smooth, "catastrophic" properties.

- We formally examine the "diachronic envelopes" possible under varying conditions of alternative language distributions, language acquisition algorithms, parameterizations, input noise, and sentence distributions—that is, what language changes are possible by varying these dimensions. This involves the simulation of these dynamical systems under different initial conditions, and characterizations of the resulting evolutionary trajectories, phase-space plots, issues of stability, and the like.
- The formal diachronic model as a dynamical system provides a novel possible source for explaining several linguistic changes including (a) the evolution of modern Greek phonology from proto-Indo-European (b) Bickerton's (199x) creole hypothesis (concerning the striking fact that all creoles, irrespective of linguistic origin, have exactly the same grammar) as the condensation point of a dynamical system (though we have not tested these possibilities explicitly).

The Acquisition-Based Model of Language Change: The Logical Problem of Language Change

How does the combination of a grammatical theory and learning algorithm lead to a model of language change? We first note that, just as with language acquisition, there is a seeming paradox in language change: it is generally assumed that children acquired their caretaker (target) grammars without error. However, if this were always true, at first glance grammatical changes within a population could seemingly never occur, since generation after generation, the children would have successfully acquired the grammar of their parents.

Of course, Lightfoot and others have pointed out the obvious solution to this paradox: the possibility of slight misconvergence to target grammars could, over time (generations), drive language change, much as speciation occurs in the population biology sense. We pursue this point in detail below. Similarly, just as in the biological case, some of the most commonly observed changes in languages seem to occur as the result of the effects of surrounding populations, whose features infiltrate the original language.

We begin our treatment of this subject by arguing that the problem of language acquisition at the individual level leads logically to the problem of language change at the group (or population) level. Consider a population speaking a particular language[31]. This is the target language—children are exposed to primary linguistic data from this source (language); typically in the form of sentences uttered by caretakers (adults). The logical problem of language acquisition is how children acquire this target language from the primary linguistic data—in other words to come up with an adequate learning theory. Such a learning algorithm is simply a mapping from primary linguistic data to the class of grammars. For example, in a typical inductive inference model (as we saw in the previous chapter), given a stream of sentences (primary linguistic data), the algorithm would simply update its grammatical hypothesis with each new sentence according to some preprogrammed procedure. An important criterion for learnability (as we saw in the previous chapter) is to require that the algorithm converge to the target as the data goes to infinity.

Now, suppose that the primary linguistic data presented to the child is altered (due, perhaps, to presence of foreign speakers, contact with another population, disfluencies etc.). In other words, the sentences presented to the learner (child) are no longer consistent with a single target grammar. In the face of this input, the learning algorithm might no longer converge to the target grammar. Indeed, it might converge to some other grammar (g_2); or it might converge to g_2 with some probability, g_3 with some other probability, and so on. In either case, children attempting to solve the acquisition problem by means of the learning algorithm, would have internalized grammars different from the parental (target) grammar. Consequently, in one generation, the linguistic composition of the population would have changed[32]. Furthermore, this change is driven by 1) the primary linguistic data (composed in this case of sentences from the original target language, and sentences from the foreign speakers) 2) the language acquisition device: which acting upon the primary evidence, causes the acquisition of a different grammar by the children. Finally, the change is limited by the hypothesis space of possible grammars; after all, the children can never converge to a grammar which lies outside this space of grammars.

In short, on this view, language change is a logical consequence of specific assumptions about

1. the *hypothesis space* of grammars—in a parametric theory, like the ones we ex-

[31] In our framework of analysis, this implies that all the adult members of this population have internalized the same grammar (corresponding to the language they speak).

[32] Sociological factors affecting language change, affect language acquisition in exactly the same way, yet are abstracted away from the formalization of the logical problem of language acquisition. In this same sense, we similarly abstract away such causes.

amine in this thesis, this corresponds to a particular choice of parameterization

2. the *language acquisition device*—in other words, the learning algorithm the child uses to develop hypotheses on the basis of data

3. the *primary linguistic data*—the sentences which are presented to the children of any one generation

If we specify 1) through 3) for a particular generation, we should, in principle, be able to compute the linguistic composition for the next generation. In this manner, we can compute the evolving linguistic composition of the population from generation to generation; we arrive at a dynamical system. We can be a bit more precise about this. First, let us recall our framework for language learning. Then we will show how to derive a dynamical system from this framework.

The Language Learning Framework:

Denote by $\mathcal{G}$, a family of possible (target) grammars. Each grammar $g \in \mathcal{G}$ defines a language $L(g) \subseteq \Sigma*$ over some alphabet Σ in the usual fashion. Let there be a distribution P on $\Sigma*$ according to which sentences are drawn and presented to the learner. Note that if there is well defined target, g_t, and only positive examples from this target are presented to the learner, then P will have all its measure on $L(g_t)$, and zero measure on sentences outside of this. Suppose n examples are drawn in this fashion, one can then let $\mathcal{D}_n = (\Sigma*)^n$ be the set of all n-example data sets the learner might potentially be presented with. A learning algorithm $\mathcal{A}$ can then be regarded as a mapping from $\mathcal{D}_n$ to $\mathcal{G}$. Thus, acting upon a particular presentation sequence $d_n \in \mathcal{D}_n$, the learner posits a hypothesis $\mathcal{A}(d_n) = h_n \in \mathcal{G}$. Allowing for the possibility of randomization, the learner could, in general, posit $h_i \in \mathcal{G}$ with probability p_i for such a presentation sequence d_n. The standard (stochastic version) learnability criterion (after Gold, 1967) can then be stated as follows:

For every target grammar, $g_t \in \mathcal{G}$, with positive-only examples presented according to P as above, the learner must converge to the target with probability 1, i.e.,

$$Prob[\mathcal{A}(d_n) = g_t] \longrightarrow_{n\to\infty} 1$$

In the previous chapter, we concerned ourselves with this learnability issue for memoryless algorithms in finite parameter spaces.

From Language Learning to Population Dynamics:

The framework for language learning has learners (children) attempting to infer grammars on the basis of linguistic data (sentences). At any point in time, n, (i.e., after hearing n examples) the child learner has a current hypothesis, h, with probability $p_n(h)$. What happens when there is a population of child learners? Since an arbitrary child learner, has a probability $p_n(h)$ of developing hypothesis h (for every $h \in \mathcal{G}$), it follows that a fraction $p_n(h)$ of the population of children would have internalized the grammar h after n examples. We therefore have a current state of the population after n examples. This state of the population (of children) might well be different from the state of the parental population. Pretend for a moment that after n examples, maturation occurs, i.e., the child retains for the rest of its life, the grammatical hypothesis after n examples, then we would have arrived at the state of the mature population for the next generation[33]. This new generation now produces sentences for the following generation of children according to the distribution of grammars in the population. The same process repeats itself and the linguistic composition of the population evolves from generation to generation.

Formalizing the Argument Further:

This formulation leads naturally to a discrete-time dynamical systems model for language change. In order to define such a dynamical system formally, one needs to specify

1. the *state space*, $\mathcal{S}$— a set of states the system can be in. At any given point in time, t, the system is in exactly one state $s \in \mathcal{S}$;

2. an *update rule* defining, the manner in which the state of the system changes from one time to the next. Typically, this involves the specification of a function, f, which maps s_t, (the state at time t) to s_{t+1} (the state at time $t+1$).[34]

For example, a typical linear dynamical system might consist of state variables $\mathbf{x}$ (where $\mathbf{x}$ is a k-dimensional state vector) and a system of differential equations $\mathbf{x}' = Ax$ (A is a matrix operator) which characterize the evolution of the states with time. RC circuits are a simple example of linear dynamical systems. The state

[33] Maturation is a reasonable hypothesis. After all, it seems even more unreasonable to imagine that children are forever wandering around in hypothesis space. After a certain point, and there is evidence from developmental psychology to suggest that this is the case, the child matures and retains its current grammatical hypothesis for the rest of its life.

[34] In general, this mapping could be fairly complicated. For example, it could depend on previous states, future states etc. For reference, see Strogatz (1993).

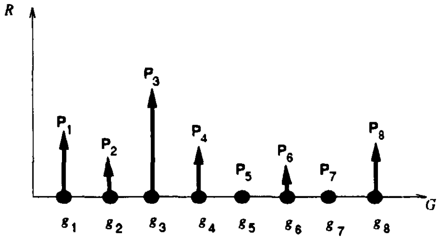

Figure 5-41: A simple illustration of the state space for the 3-parameter syntactic case. There are 8 grammars, a probability distribution on these 8 grammars, as shown above, can be interpreted as the linguistic composition of the population. Thus, a fraction P_1 of the population have internalized grammar, g_1, and so on.

(current) evolves as the capacitor discharges through the resistor. Population growth models (for example, using logistic equations) provide other examples.

The State Space:

In our case, the state space is the space of possible linguistic compositions of the population. More specifically, it is a distribution P_{pop} on the space of grammars, $\mathcal{G}$[35]. For example, consider the three parameter syntactic space described in Gibson and Wexler (1994) and analyzed in the previous chapter. This defines 8 possible "natural" grammars. Thus $\mathcal{G}$ has 8 elements. We can picture a distribution on this space as shown in fig. 5-41. In this particular case, the state space is

$$\mathcal{S} = \{\mathbf{P} \in R^8 | \sum_{i=1}^{8} \mathbf{P}_i = 1\}$$

We interpret the state as the linguistic composition of the population. For example, a distribution which puts all its weight on grammar g_1 and 0 everywhere else, indicates a homogeneous population which speaks the language corresponding to grammar g_1. Similarly, a distribution which puts a probability mass of 1/2 on g_1 and 1/2 on g_2 indicates a population (non-homogeneous) with half its speakers speaking a language corresponding to g_1 and half speaking a language corresponding to g_2.

The Update Rule:

The update rule is obtained by considering the learning algorithm, $\mathcal{A}$, involved. For example, given the state at time t, $(P_{pop,t})$, i.e., the distribution of speakers in the parental population (they are the generators of the primary linguistic data for

[35] Obviously one needs to be able to define a σ-algebra on the space of grammars, and so on. For the cases we look at, this is not a problem because the set of grammars is finite.

the next generation), one can obtain the distribution with which sentences from $\Sigma*$ will be presented to the learner. To do this, imagine that the ith linguistic group in the population (speaking language L_i) produces sentences with distribution P_i (on the sentences of L_i, i.e., sentences not in L_i are produced with probability 0). Then for any $\omega \in \Sigma*$, the probability with which it is presented to the learner is given by

$$P(\omega) = \sum_i P_i(\omega)P_{pop,t}(i)$$

Now that the distribution with which sentences are presented to the learner is determined, the algorithm operates on the linguistic data, d_n, (this is a dataset of n example sentences drawn according to distribution P) and develops hypotheses ($\mathcal{A}(d_n) \in \mathcal{G}$). Furthermore, one can, in principle, compute the probability with which the learner will develop hypothesis h_i after n examples:

$$\textbf{Finite Sample: } Prob[\mathcal{A}(d_n) = h_i] = p_n(h_i) \tag{5.33}$$

This finite sample situation is always well defined. In other words, the probability p_n exists[36].

Learnability requires $p_n(g_t)$ to go to 1, for the unique target grammar, g_t, if such a grammar exists. In general, however, there is no unique target grammar since we have non-homogeneous linguistic populations. However, the following limiting behavior might still exist:

$$\textbf{Limiting Sample: } \lim_{n\to\infty} Prob[\mathcal{A}(d_n) = h_i] = p_i \tag{5.34}$$

Thus, the child, according to the arguments described earlier, internalizes grammar $h_i \in \mathcal{G}$ with probability $p_n(h_i)$ (for a finite sample analysis) and with probability p_i "in the limit". We can find p_i for every i, and the next generation would then have a proportion p_i (or $p_n(h_i)$, if one wanted to do a finite sample analysis) of people who have internalized the grammar h_i. Consequently, the linguistic composition of the next generation is given by $P_{pop,t+1}(h_i) = p_i$(or $p_n(h_i)$). In this fashion,

$$P_{pop,t} \longrightarrow^{\mathcal{A}} P_{pop,t+1}$$

[36] This is easy to see for deterministic algorithms, $\mathcal{A}_{det}$. Such an algorithm would have a precise behavior for every data set of n examples drawn. In our case, the examples are drawn in i.i.d. fashion according to a distribution P on $\Sigma*$. It is clear that $p_n(h_i) = P[\{d_n | \mathcal{A}_{det}(d_n) = h_i\}]$. For randomized algorithms, the case is trickier, but the probability still exists. We saw in the previous chapter, how to compute p_n for randomized memoryless algorithms.

Remarks

1. The finite sample case probability always exists. Suppose, we have solved the maturation problem, i.e., we know the rough amount of time, the learner takes to develop its mature (adult) hypothesis. This is tantamount to knowing (roughly, if not exactly) the number of examples, N, the child would have heard by then. In that case $p_N(h)$ is the probability that the child internalizes the grammar h. This ($p_N(h)$) is the percentage of speakers of L_h in the next generation. Note that under this finite sample analysis, for a homogeneous population, with all adults speaking a particular language (corresponding to grammar, g, say), $p_N(g)$ will not be 1—that is, there will be a small percentage who have misconverged. This percentage might blow up over generations; and we potentially have unstable languages. This is in contrast to the limiting analysis of homogeneous populations which is trivial for learnable families of grammars.

2. The limiting case analysis is more problematic, though more consistent with learnability theories "in the limit." First, the limit in question need not always exist. In such a case, of course, no limiting analysis is possible. If however, the limit does exist, then p_i is the probability that a child learner attains the grammar p_i in the limit—and this is the proportion of the population with this internal grammar in the next generation.

3. In general, the linguistic composition for the $(t+1)$th generation is given in similar fashion from the linguistic composition for the tth generation. Such a dynamical system exists for every assumption of a)$\mathcal{A}$, and b)$\mathcal{G}$ and c)P_i's the probability with which sentences are produced by speakers of the ith grammar[37]. Thus we see that ,

$$(\mathcal{G}, \mathcal{A}, \{P_i\}) \longrightarrow \mathcal{D}(\text{ dynamical system})$$

4. The formulation is completely general so far. It does not assume any particular linguistic theory, or learning algorithm, or distribution with which sentences are drawn. Of course, we have implicitly assumed a learning model. i.e., positive examples are drawn in i.i.d. fashion and presented to the learner (algorithm). Our formalization of the grammatical dynamical systems follows as a logical consequence of this learning framework. One can conceivably imagine other learning frameworks—these would potentially give rise to other kinds of dynamical systems; but we don't formalize them here.

At this stage, we have developed our case in abstraction. The next obvious step is to choose specific linguistic theories, and learning paradigms, and compute our

[37] Note that this probability could evolve with generations as well. That will complete all the logical possibilities. However, for simplicity, we assume that this does not happen.

dynamical system. The important questions are: can we really compute all the relevant quantities to specify the dynamical system?? Can we evaluate the behavior (the phase-space characteristics) of the resulting dynamical system?? Does this allow us to shed light on linguistic theories?? We show some concrete examples of this in this chapter. Our examples are conducted within the principles and parameters theory of modern linguistics.

5.2 Language Change in Parametric Systems

The previous section led us through the important steps in formalizing the process of language change, leading ultimately to a computational paradigm within which such change can be meaningfully studied. We carry out our investigations within the principles and parameters framework introduced in the previous chapter. In Chapter 4, we investigated the problem of learnability within this framework. In particular, we saw that the behavior of any memoryless algorithm can be modeled as a Markov chain. This analysis will allow us to solve equations 1 and 2, and obtain the update equations of our dynamical system. We now proceed to do this.

1) the *grammatical theory:* Assume there are n parameters—this leads to a space $\mathcal{G}$ with 2^n different grammars in it.

2) the *distribution* with which data is produced: If there are speakers of the ith language, L_i, in the population, let them produce sentences according to the distribution, P_i, on the sentences of this language. For the most part, we will assume, in our simulations, that this is uniform on degree-0 sentences (exactly as we did in our analysis of the learnability problem).

3) the learning *algorithm:* Let us imagine that the child learner follows some memoryless (incremental) algorithm to set parameters. For the most part, we will assume that the algorithm is the TLA or one of the variants discussed in the previous chapter.

From One Generation to the Next: The Update Rule

Suppose the state of the parental population is $P_{pop,n}$ on $\mathcal{G}$. Then one can obtain the distribution P on the sentences of $\Sigma*$ according to which sentences will be presented to the learner. Once such a distribution is obtained, we can compute the transition matrix T according to which the learner updates its hypotheses with each new sentence (as shown in the previous chapter). From T, one can finally compute the following quantities:

$$Prob[\text{ Learner's hypothesis } = h_i \in \mathcal{G} \text{ after } m \text{ examples}] == \{\frac{1}{2^n}(1,\ldots,1)'T^m\}[i]$$

Similarly, making use of limiting distributions of Markov chains (see Resnick, 1992) one can obtain the following (where ONE is a $\frac{1}{2^n} \times \frac{1}{2^n}$ matrix with all ones).

$$Prob[\text{ Learner's hypothesis } = h_i \text{ ``in the limit''}] = (1,\ldots.1)'(I - T + ONE)^{-1}$$

These expressions allow us to compute the linguistic composition of the population according to our analysis of the previous section.

Remarks:

1. The limiting distribution needs to be interpreted. Markov chains corresponding to population mixes do not have an absorbing state. Instead they have recurrent states. These states will be visited infinitely often. There might be more than one state that will be visited infinitely often. However, the percentage of time, the learner will be in a particular state might vary. This is provided by the equation above. Since, we know the fraction of the time the learner spends in each grammatical state in the limit, we assume that this is the probability with which it internalize the grammar corresponding to that state in the Markov chain.

2. The finite case analysis always works. The limiting analysis need not work. However, the limiting analysis works only when there is more than one target. That is, if there is only one target grammar, for learnable algorithms. all children would converge to that target in the limit, and the population characteristics would not change with generations.

We provide now the basic computational framework for modeling language change.

1. Let π_1 be the initial population mix. i.e.. the percentage of different language speakers in the community. Assuming. then. that the ith group of speakers produce sentences with probability P_i, we can obtain P with which sentences in $\Sigma*$ occur for the next generation of children.

2. From P, we can obtain the transition probabilities for the child learners and the limiting distribution π_2 for the next generation.

3. The second generation produce sentences with π_2. We can repeat step 1 and obtain π_3; in general a population mix π_i will over a generation change to a mix of π_{i+1}.

5.3 Example 1: A Three Parameter System

The previous section developed the necessary mathematical and computational tools to completely specify the dynamical systems corresponding to memoryless algorithms operating on finite parameter spaces. In this example, we investigate the behavior of these dynamical systems. Recall that every choice of $(\mathcal{G}, \mathcal{A}, \{P_i\})$ gives rise to a unique dynamical system. We start by assuming:

1) $\mathcal{G}$: This is a 3-parameter syntactic subsystem described in the previous chapter (Gibson and Wexler, 1994). Thus $\mathcal{G}$ has exactly 8 grammars.

2) $\mathcal{A}$: The memoryless algorithms we consider are the TLA, and variants by dropping either or both of the single-valued and greediness constraints.

3) $\{P_i\}$: For the most part, we assume sentences are produced according to a uniform distribution on the degree-0 sentences of the relevant language, i.e., P_i is uniform on (degree-0 sentences of) L_i.

5.3.1 Starting with Homogeneous Populations:

Here we investigate how stable the languages in the parametric system are in the absence of noise or other confounding factors like foreign speech. Thus we start off with a linguistically homogeneous population producing sentences according to a uniform distribution on the degree-0 sentences of the target language (parental language). We compute the the distribution of the children in the parameter space after 128 example sentences (recall, by the analysis of the previous chapter, the learners converge to the target with high probability after hearing these many sentences). Some small proportion of the children will have misconverged; the goal is to see whether this small proportion can drive language change—and if so, in what direction.

$\mathcal{A}$ = TLA; P_i = Uniform; Finite Sample = 128

The table below shows the result after 30 generations. Languages are numbered from 1 to 8 according to the scheme in the appendix of chapter 4.
Observations: Some striking patterns are observed.

1. First, all the +V2 languages are relatively stable, i.e., the linguistic composition did not vary significantly over 30 generations. This means that every succeeding generation acquired the target parameter settings and no parameter drifts were observed over time.

2. Populations speaking -V2 languages all drift to speaking +V2 languages. Thus a population speaking L_1 starts speaking mostly L_2. A population speaking language L_7 gradually shifts to a population with 54 percent speaking L_2 and 35 percent

Initial Language	Change to Language?
$(-V2)$ 1	2 (0.85), 6 (0.1)
$(+V2)$ 2	2 (0.98); stable
$(-V2)$ 3	6 (0.48), 8(0.38)
$(+V2)$ 4	4 (0.86); stable
$(-V2)$ 5	2 (0.97)
$(+V2)$ 6	6 (0.92); stable
$(-V2)$ 7	2 (0.54), 4(0.35)
$(+V2)$ 8	8 (0.97); stable

Table 5.3: Language change driven by misconvergence. A finite-sample analysis was conducted allowing each child learner 128 examples to internalize its grammar.. Initial populations were linguistically homogeneous, and they drifted (or not) to different linguistic compositions. The major language groups after 30 generations have been listed in this table.

speaking L_4 (with a smattering of other speakers) and seems (?) to remain basically stable in this mix thereafter. Note that this relative stability of +V2, and the tendency of -V2 languages to drift to +V2 ones, are contrary to assertions in the linguistic literature. Lightfoot (1991), for example, claims the tendency to lose V2 dominates the reverse tendency in the world's languages. Certainly, both English and French lost the V2 parameter setting—an empirically observed phenomenon that needs to be explained. Right away, we see that our dynamical system does not evolve in the expected pattern. The problem could be due to incorrect assumptions about the parameter space, the algorithm, initial conditions, or distributional assumptions about the sentences. This needs to be examined, no doubt, but we have just seen a concrete example of how assumptions about grammatical theory, and learning theory, have made evolutionary predictions—in this case the predictions are incorrect, and our model is falsified.

3. The rates at which the linguistic composition changes varies significantly. Consider for example the change of L_1 to L_2. Fig. 5-42 below shows the gradual decrease in speakers of L_1 over successive generations along with the increase in L_2 speakers. We see that over the first 6 or seven generations very little change occurs, thereafter over the next 6 or seven generations the population switches at a much faster rate. Note that in this particular case, the two languages differ only in the V2 parameter; so the curves essentially plot the gain of V2. In contrast, consider fig. 5-43 which shows the decrease of L_5 speakers and the shift to L_2. Here we notice a sudden change; over a space of 4 generations, the population has shifted completely. The time course of language change has been given some attention in linguistic analyses of diachronic syntax change, and we return to this in a later section.

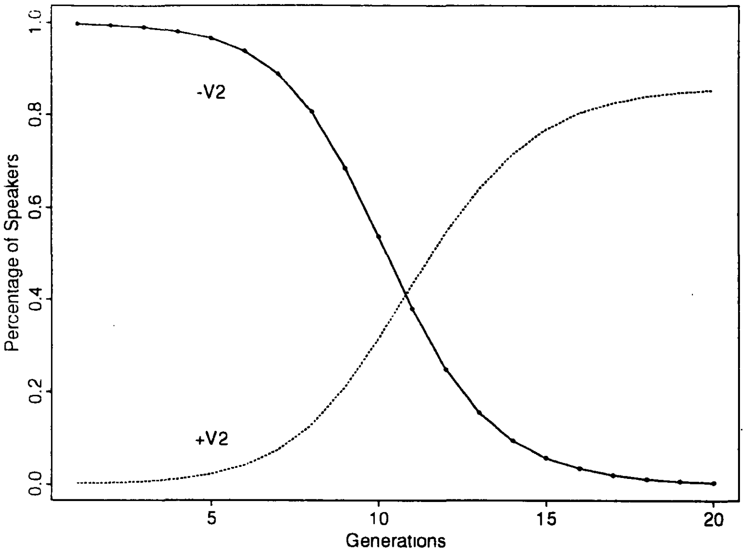

Figure 5-42: Percentage of the population speaking languages L_1 and L_2 as it evolves over the number of generations. The plot has been shown only upto 20 generations, as the proportions of L_1 and L_2 speakers do not vary significantly thereafter. Notice the "S" shaped nature of the curve (Kroch, 1989, imposes such a shape using models from population biology, while we obtain this as an emergent property of our dynamical model from different starting assumptions). Also notice the region of maximum change as the V2 parameter is slowly set by increasing proportion of the population. L_1 and L_2 differ only in the V2 parameter setting.

4. We see that in many cases, the homogeneous population splits up into different linguistic groups, and seem to remain stable in that mix. In other words, certain combinations of language speakers seem to asymptote towards equilibrium (atleast by examining the 30 generations simulated so far). For example, a population of L_7 speakers shifts (over 5-6 generations) to one with 54 percent speaking L_2 and 35 percent speaking L_4 and remains that way with no shifts in the distribution of speakers. Is this really a stable mix? Or will the population shift suddenly after another 100 generations? Can we characterize the stable points ("limit cycles")? Other linguistic mixes are inherently unstable mixes. They might drift systematically to stable situations, or might shift dramatically.

In table 5.3, why are some languages stable while others are unstable? It seems that the instability and the drifts observed are to a large extent an artifact of the learning algorithm used. Remember that TLA suffers from the problem of local maxima. We notice that those languages whose acquisition is not impeded by local

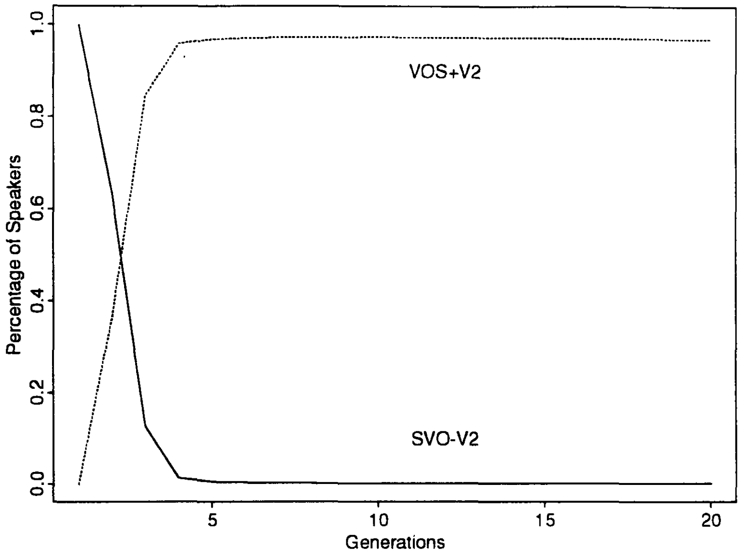

Figure 5-43: Percentage of the population speaking languages L_5 and L_2 as it evolves over the number of generations. Notice how the shift occurs over a space of 4 generations.

maxima (the +V2 languages) are stable. Languages which have local maxima are unstable; in particular they drift to the local maxima over time. Consider L_7. If this is the target, then there are two local maxima (L_2 and L_4) and these are precisely the states to which the system drifts over time. The same is true for languages L_5 and L_3. In this respect, the behavior of L_1 is unusual since it actually does not have any local maxima; yet it tends to flip the V2 parameter over time.

Remark. We regard local maxima of a language L_i to be alternative absorbing states (sinks) in the Markov chain for that target language. This differs slightly from the conception of local maxima in Gibson and Wexler (1994), a matter discussed at some length in Niyogi and Berwick (1993), and in the previous chapter. Thus according to our definition L_4 is not a local maxima for L_5 and consequently no shift is observed.

$\mathcal{A}$ = Greedy, No S.V.; P_i = Uniform; Finite Sample = 128

The previous discussion of grammatical evolution with TLA begs the question: what if the learner used some alternative learning algorithm which did not suffer from the problem of local maxima? To investigate this, we consider a simple variant of the TLA obtained by dropping the single valued constraint. This implies that the learner is no longer constrained to flip one parameter at a time. On being presented with a

Initial Language	Change to Language?
$-V2$ 1	2 (0.41), 4 (0.19), 6 (0.18), 8 (0.13)
$+V2$ 2	2 (0.42), 4 (0.19), 6 (0.17), 8 (0.12)
$-V2$ 3	2 (0.40), 4 (0.19), 6 (0.18), 8 (0.13)
$+V2$ 4	2 (0.41), 4 (0.19), 6 (0.18), 8 (0.13)
$-V2$ 5	2 (0.40), 4 (0.19), 6 (0.18), 8 (0.13)
$+V2$ 6	2 (0.40), 4 (0.19), 6 (0.18), 8 (0.13)
$-V2$ 7	2 (0.40), 4 (0.19), 6 (0.18), 8 (0.13)
$+V2$ 8	2 (0.40), 4 (0.19), 6 (0.18), 8 (0.13)

Table 5.4: Language change driven by misconvergence. A finite-sample analysis was conducted allowing each child learner (following the TLA with single-value dropped) 128 examples to internalize its grammar. Initial populations were linguistically homogeneous, and they drifted to different linguistic compositions. The major language groups after 30 generations have been listed in this table. Notice how all initially homogeneous populations tend to the same composition.

sentence it cannot analyze, it chooses any of the alternative grammars and attempts to analyze the sentence with it. Greediness is retained; thus the learner retains its original hypothesis if the new one is also not able to analyze the sentence. Table 5.4 shows the distribution of speakers after 30 generations.

Observations: In this situation there are no local maxima, and the pattern of evolution takes on a very different nature. There are two distinct observations to be made.

1. All homogeneous populations (irrespective of what language they speak) eventually drift to a strikingly similar population mix. What is unique about this mix? Is it a stable point (or attractor)? Further simulations, and theoretical analysis is needed to resolve this question.

2. All homogeneous populations drift to a population mix of only +V2 languages. Thus, the V2 parameter is gradually set over succeeding generations by all people in the community (irrespective of which language they speak). In other words, there is as before a tendency to gain V2 rather than lose it (we emphasize again, that this is contrary to linguistic intuition).

Fig. 5-44 shows the changing percentage of the population speaking the different languages starting off from a homogeneous population speaking L_5. As before, learners who have not converged to the target in 128 examples are the driving force for change here. Note again the time evolution of the grammars. For about 5 generations there is only a slight decrease in the percentage of speakers of L_5. Then the linguistic patterns switch over the next 7 generations to a relatively stable mix.

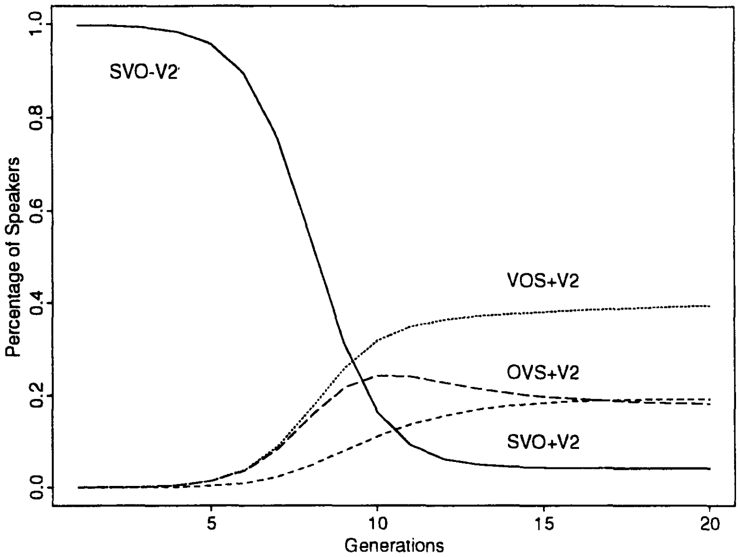

Figure 5-44: Time evolution of grammars using greedy algorithm with no single value.

Initial Language	Change to Language?
Any Language	1 (0.11), 2 (0.16), 3 (0.10), 4 (0.14)
(Homogeneous)	5 (0.12), 6 (0.14), 7 (0.10), 8 (0.13)

Table 5.5: Language change driven by misconvergence. A finite-sample analysis was conducted allowing each child learner (following 1) random walk and 2) the TLA with greediness dropped) 128 examples to internalize its grammar. Initial populations were linguistically homogeneous, and they drifted to different linguistic compositions. The major language groups after 30 generations have been listed in this table. Notice, again, how all initially homogeneous populations tend to the same composition.

$\mathcal{A}$ = a) R.W. b) S. V. only; P_i = Uniform; Finite Sample = 128

Here we simulated the evolution of the dynamical systems corresponding to two algorithms, both of which have no greediness constraint. The two algorithms are 1) the *random walk* described in the previous chapter and 2) TLA with *single-value retained* but no greediness constraint.

In both cases, the population mix after 30 generations is the same. irrespective of the initial language of the homogeneous population. This is shown in table 5.5.

Observations:

1. The first striking observation is that both algorithms yield dynamical systems which arrive at the same population mix after 30 generations. The path by which they arrive at this mix is, however, not the same (see fig. 5-45).

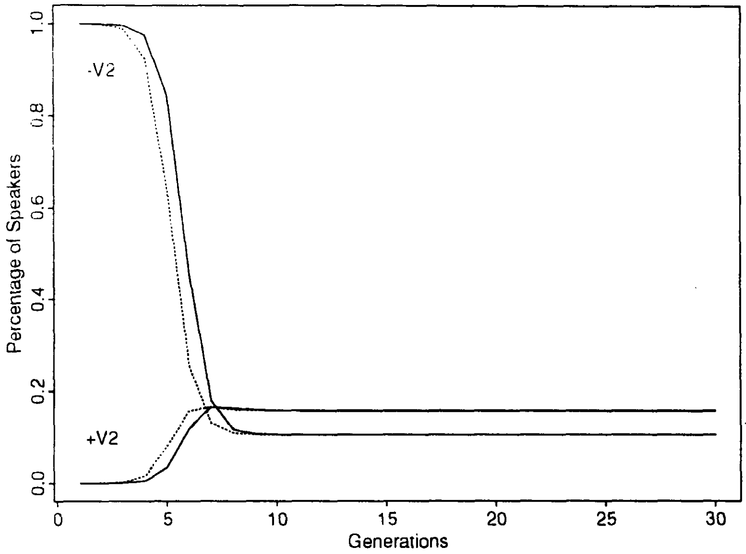

Figure 5-45: Time evolution of linguistic composition for the situations where the learning algorithm used is the TLA (with greediness dropped, corresponding to the dotted line) , and the Random Walk (solid line). Only the percentage of people speaking L_1 (-V2) and L_2 (+V2) are shown. The initial population is homogeneous and speaks L_1. The percentage of L_1 speakers gradually decreases to about 11 percent. The percentage of L_2 speakers rises to about 16 percent from 0 percent. The two dynamical systems (corresponding to S.V. and R.W.) converge to the same population mix. However, the trajectory is not the same—the rates of change are different, as shown in this plot.

2. It is also noteworthy, that in these cases, all initially homogeneous populations converge to a single population mix. Further, this population mix contains all languages in significant proportion. This is in contrast to the previous situations, where we saw that non-V2 languages were eliminated.

Rates of Change

The figures depicting the evolutionary trajectories of the dynamical systems often have an S-shaped behavior (though the "smoothness" of this trajectory varied). In this section, we examine a few factors which affect the time-course (more generally, trajectories) of our grammatical dynamical systems. We begin by noting that linguists have, in the past, pursued this question.

Some Linguistic thoughts on the time-course of language variation:

Bailey (1973) proposed a "wave" model of linguistic change. Among other things, this proposes that linguistic replacements follow an S-shaped curve in time. In Bailey's own words (taken from Kroch, 1990)

> A given change begins quite gradually; after reaching a certain point (say, twenty percent), it picks up momentum and proceeds at a much faster rate; and finally tails off slowly before reaching completion. The result is an S-curve: the statistical differences among isolects in the middle relative times of the change will be greater than the statistical differences among the early and late isolects.

The idea that linguistic changes follow an S-curve has been proposed by Osgood and Sebeok (1954), Weinreich, Labov, and Herzog (1968). More specific logistic forms were proposed by Altmann (1983), and Kroch (1982,1989). The idea of the logistic functional form is borrowed from population biology where it is demonstrable that the logistic governs the replacement of organisms and of genetic alleles that differ in Darwinian fitness. However Kroch concedes that "unlike in the population biology case, no mechanism of change has been proposed from which the logistic form can be deduced".

Crucially, in our case, we suggest a specific acquisition-based model of language change. The combination of grammatical theory, learning algorithms, and distributional assumptions on sentences drive change—the specific form of the change (which might or might not be S-shaped, and might have varying rates) is thus a derivative of more fundamental assumptions. This is in contrast with the above-mentioned theories of change.

The effect of maturational time

One obvious factor influencing the evolutionary trajectories is the maturational time, i.e., the number (N) of sentences the child is allowed to hear before forming its mature hypothesis. This was kept at 128 in all the systems shown so far. Fig. 5-46 shows the effect of N on the evolutionary trajectories. As usual, we plot only a subspace of the population. In particular, we plot the percentage of L_2 speakers in the population with each succeeding generation. The initial composition of the population was homogeneous (with people speaking L_1). It is worthwhile to make a few observations:

1. The initial rate of change of the population is highest for the situation where the maturational time is the least, i.e., the learner is allowed the least amount of time to develop its mature hypothesis. This is hardly surprising. If the learner were allowed access to a lot of examples to make its mature hypothesis, most of the learners would have reached the target grammar. Very few would

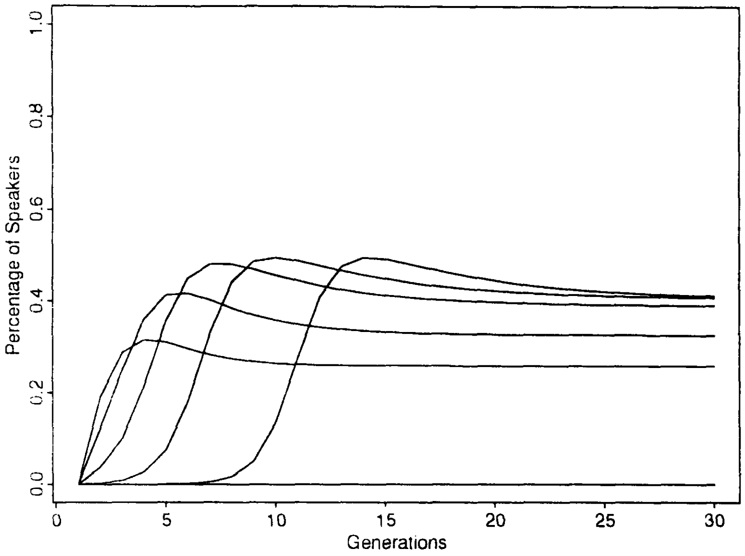

Figure 5-46: Time evolution of linguistic composition for the situations where the learning algorithm used is the TLA (with single-value dropped). Only the percentage of people speaking L_2 (+V2) is shown. The initial population is homogeneous and speaks L_1. The maturational time (number,N, of sentences the child hears before internalizing a grammar) is varied through 8, 16, 32, 64, 128, 256, giving rise to the six curves shown in the figure. The curve which has the highest initial rate of change corresponds to the situation where 8 examples were allowed to the learner to develop its mature hypothesis. The initial rate of change decreases as the maturation time N increases. The value at which these curves asymptote also seems to vary with the maturation time. and increases monotonically with it.

have misconverged. and the linguistic composition would have changed little over the next generation. On the other hand, if the learner were allowed very few examples to develop its hypothesis. many would misconverge, causing great change over one generation.

2. The "stable" linguistic compositions seem to depend upon maturational time. For example, if the learner is allowed only 8 examples, the number of L_2 speakers rises quickly to about 0.26. On the other hand, if the learner is allowed 128 examples, the number of L_2 speakers eventually rises to about 0.41.

3. Note that the trajectories do not have an S-shaped curve.

4. The maturational time is related to the order of the dynamical system.

The effect of sentence distributions P_i.

Another important factor influencing the evolutionary trajectories is the distribution P_i with which sentences of the ith language, L_i, are presented to the learner. In a certain sense, the grammatical space and the learning algorithm determine the order of the dynamical system. The sentence distributions on the other hand, are like the parameters of the dynamical system (we comment on this point later). Clearly the sentence distributions affect rates of convergence within one generation as we saw in the previous chapter. Further, by putting greater weight on certain word forms rather than others, they might influence the systemic evolution in certain directions.

To illustrate this idea, we consider an example. We study the interaction between L_1 and L_2 speakers in the community as the sentence distributions with which these speakers produce sentences changes. Recall that so far, we have assumed that all speakers produce sentences with uniform distributions on degree-0 sentences of the language. Now, we consider an alternative distribution as below:

1. Let $L_{1,2} = L_1 \cap L_2$.

2. P_1 : Speakers of L_1 produce sentences so that all degree-0 sentences of $L_{1,2}$ are equally likely and their total probability is p. Further. sentences of $L_1 \setminus L_{1,2}$ are also equally likely, but their total probability is $1 - p$.

3. P_2 : Speakers of L_2 produce sentences so that all degree-0 sentences of $L_{1,2}$ are equally likely and their total probability is p. Further, sentences of $L_2 \setminus L_{1,2}$ are also equally likely, but their total probability is $1 - p$.

4. Other P_i's are all uniform in degree-0 sentences.

Thus, the distributions P_i's are parameterized by a single parameter, p, which determines the amount of measure on the sentence patterns in common between the languages L_1 and L_2. Fig. 5-47 shows the evolution of the L_2 speakers as p varies. The learning algorithm used was the TLA, and the initial population was homogeneous (speaking language L_1). Thus, the initial percentage of L_2 speakers in the community was 0. Notice how the system moves in different ways as p varies. When p is very small (0.05), i.e., strings common to L_1 and L_2 occur infrequently, the long term implication is that L_2 speakers do not grow in the community. As p increases. more strings of L_2 occur, and the system is driven to increase the number of L_2 speakers until $p = 0.75$ when the population evolves into a completely L_2 speaking community. After this, as p increases further, we notice (see $p = 0.95$) that the L_2 speakers increase but can never rise to 100 percent of the population. there is still a residual L_1 speaking component. This is natural, because for such high values of p. a lot of strings common

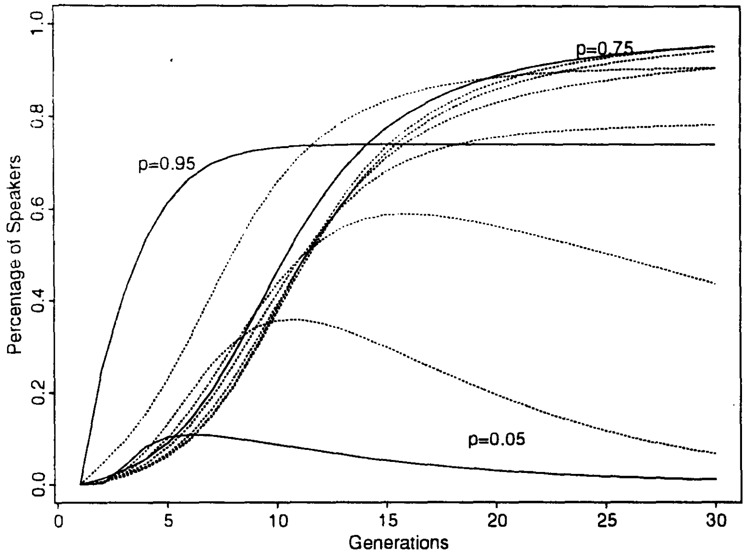

Figure 5-47: The evolution of L_2 speakers in the community for various values of p (a parameter related to the sentence distributions P_i, see text). The algorithm used was the TLA, the inital population was homogeneous, speaking only L_1. The curves for $p = 0.05, 0.75$, and 0.95 have been plotted as solid lines.

to L_1 and L_2 occur all the time. This means that the learner could converge to L_1 just as well, and some learners indeed begin to do so increasing the number of the L_1 speakers.

This example shows us that if we wanted a homogeneous L_1 speaking population to move to a homogeneous L_2 speaking population, by choosing our distributions appropriately, we could drive the grammatical dynamical system in the appropriate direction. This suggests another important application of our dynamical system approach. We can work backwards, and examine the conditions needed to generate a change of a certain kind. By checking whether such conditions could possibly have existed in history, we can falsify a grammatical theory, or a learning paradigm. Note that this example showed the effect of sentence distributions, and how to tinker with them to obtain desired evolution. One could, in principle, tinker with the grammatical theory, or the learning algorithm in the same fashion—leading to a powerful new tool to aid the search for an adequate linguistic theory.

5.3.2 Non-homogeneous Populations: Phase-Space Plots

For our three-parameter system, we have been able to characterize the update rules for the dynamical systems corresponding to a variety of learning algorithms. Each such dynamical system has a specific update procedure according to which the states evolve, from some *initial* state. In the earlier section, we examined the evolutionary trajectories when the population was homogeneous. A more complete characterization of the dynamical system would be achieved by obtaining *phase-space* plots of this system. Such phase-space plots are pictures of the state-space $\mathcal{S}$ filled with *trajectories* obtained by letting the system evolve from various initial points (states) in the state space.

Phase-Space Plots: Grammatical Trajectories

We have described earlier, the relationship between the state of the population in one generation and the next. In our case, let Π denote an 8-dimensional vector variable (state variable). Specifically, $\Pi = (\pi_1, \ldots, \pi_8)'$ (with $\sum_{i=1}^{8} \pi_i$) as we discussed before. The following schema reiterates the chain of dependencies involved in the update rule governing system evolution. The state of the population at time t (in generations), allows us to compute the transition matrix T for the Markov chain associated with the memoryless learner. Now, depending upon whether we want 1) an asymptotic analysis or 2) a finite sample analysis, we compute 1) the limiting behavior of T^m as m (the number of examples) goes to infinity (for an asymptotic analysis), or 2) the value of T^N (where N is the number of examples after which maturation occurs). This allows us to compute the new state of the population. Thus $\Pi(t+1) = g(\Pi(t))$ where g is a complex non-linear relation.

$$\Pi(t) \Longrightarrow P \text{ on } \Sigma* \Longrightarrow T \Longrightarrow T^m \Longrightarrow \Pi(t+1)$$

If we choose a certain initial condition Π_1, the system will evolve according to the above relation and one can obtain a trajectory of Π in the 8 dimensional space over time. Each initial condition yields a unique trajectory and one can then plot these trajectories obtaining a phase-space plot. Now, each such trajectory corresponds to a line in the 8-dimensional plane given by $\sum_{i=1}^{8} \pi_i = 1$. It is obviously not possible to display such a high dimensional object but we plot in fig. 5-48 the projection of a particular trajectory onto a two dimensional subspace given by $(\pi_1(t), \pi_2(t))$ (in other words, the proportion of speakers of L_1 and L_2) at different points in time.

As mentioned earlier, with a different initial condition, we get a different grammatical trajectory. The state space is thus filled with all the different trajectories

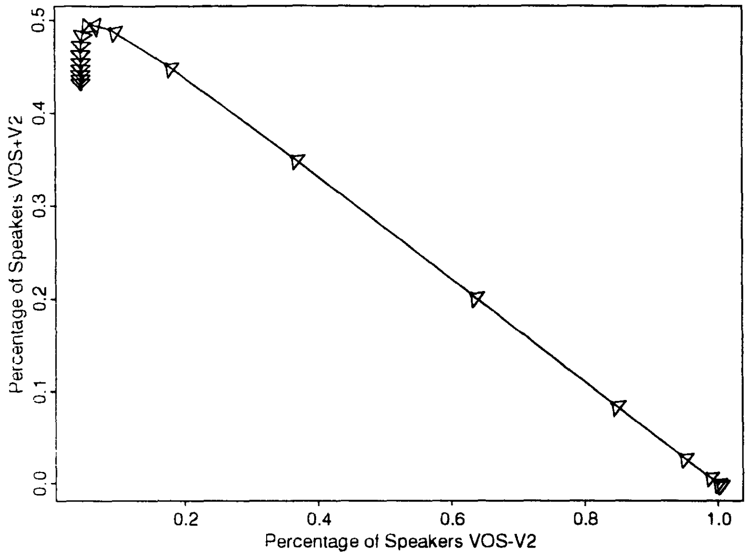

Figure 5-48: Subspace of a Phase-space plot. The plot shows $(\pi_1(t), \pi_2(t))$ as t varies, i.e., the proportion of speakers speaking languages L_1 and L_2 in the population. The initial state of the population was homogeneous (speaking language L_1). The algorithm used was the TLA with the single-value constraint dropped.

corresponding to different initial conditions. Fig. 5-49 shows this.

Issues of Stability

We notice from the phase-space plots that many of the initial conditions yield trajectories which seem to converge to a point in the state space. In the dynamical systems terminology, this would correspond to a fixed point of the system. In other words, this is a population mix which would remain that way. Some natural questions arise at this stage. What are the conditions for stability? How many fixed points are there in the system? How do we solve for them? These are interesting questions but detailed answers are not within the scope of this thesis. We would like to state here a fixed point theorem which allows us to characterize the stable population mixes.

First, some notational preliminaries. As before, let P_i be the distribution on the sentences of the ith language L_i. From P_i, we can construct T_i, the transition matrix whose elements are given by the explicit procedure documented in the previous chapter. This matrix, T_i, models the behavior of the TLA learner if the target language was L_i (with sentences from the target produced with P_i). Similarly, one can obtain the matrices for variants of the TLA. Note that fixing the P_i's fixes the T_i's and these

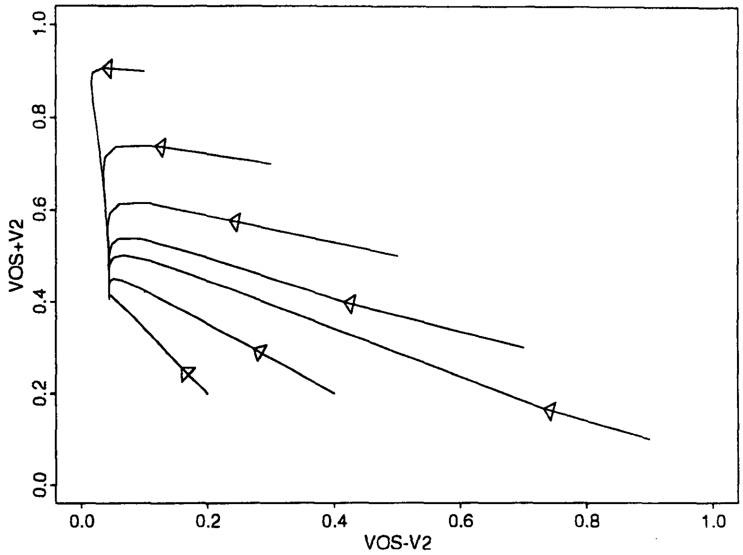

Figure 5-49: Subspace of a Phase-space plot. The plot shows $(\pi_1(t), \pi_2(t))$ as t varies for different initial conditions (non-homogeneous populations). The algorithm used by the learner is the TLA with single-value constraint dropped.

can be considered to be the parameters[38] of the dynamical system. If the state of the (parental) population at time t is $\Pi(t)$, then it is possible to show that the (true) transition matrix of the TLA (or TLA-like) learner is $T = \sum_{i=1}^{8} \pi_i(t)T_i$. For the finite case analysis, the following theorem holds:

Theorem 5.3.1 (Finite Case) *A fixed point (stable point) of the grammatical dynamical system (obtained by a TLA like learner operating on the 8 parameter space with k examples to choose its mature hypothesis) is a solution of the following equation:*

$$\Pi' = (\pi_1, \ldots, \pi_8) = (1, \ldots, 1)'(\sum_{i=1}^{8} \pi_i T_i)^k$$

Proof (Sketch): This equation is obtained simply by setting $\Pi(t+1) = \Pi(t)$. Note however, that this is an example of a non-linear multi-dimensional iterated function map. The analysis of such a dynamical system is quite non-trivial, and our theorem by no means captures all the possibilities. ∎

[38] There might be some confusion at the two different notions of parameters floating around. Just to clarify further; we have n linguistic parameters which define the 2^n languages and define the state-space of the system. We also have the P_i's which characterize the way in which the system evolves and are therefore the parameters of the complete grammatical dynamical system.

We can similarly state a theorem for the limiting case analysis.

Theorem 5.3.2 (Limiting Analysis) *A fixed point (stable point) of the grammatical dynamical system (obtained by a TLA like learner operating on the 8 parameter space (given infinite examples to choose its mature hypothesis) is a solution of the following equation:*

$$\Pi' = (\pi_1, \ldots, \pi_8) = (1, \ldots, 1)'(I - \sum_{i=1}^{8} \pi_i T_i + ONE)^{-1}$$

where ONE is the 8×8 matrix with all its entries equal to 1.

Proof: Again this is trivially obtained by setting $\Pi(t+1) = \Pi(t)$. The expression on the right provides an analytical expression for the update equation in the asymptotic case. See Resnick (1992) for details. All the caveats mentioned in the proof section of the previous theorem apply here as well. ▮

Remark: We have just scratched the surface as far as the theoretical characterization of these grammatical dynamical systems are concerned. The main purpose of this chapter is to show that these dynamical systems exist as a logical consequence of assumptions about the grammatical space, and a learning theory. We have demonstrated some preliminary simulations with these systems. From a theoretical perspective, it would be very interesting to better understand such systems. Strogatz (1993) suggests that non-linear multidimensional (more than 3 dimensions) mappings are likely to be chaotic. Such investigations are beyond the scope of this thesis, and might be a fruitful area for further research.

5.4 Example 2: The Case of Modern French:

The previous example considered a 3-parameter system for which we derived several different dynamical systems. Our goal was to concretely instantiate our philosophical arguments in sections 2 and 3, and provide a flavor of the many different factors which influence the evolution of these grammatical dynamical systems. In this section, we briefly consider a different parametric system (studied by Clark and Roberts, 1993). The historical context in which we study this is the evolution of Modern French from Old French.

Extensive simulations in the earlier section reveal that while the learnability problem of the 3-parameter space can be solved by stochastic hill climbing algorithms, the long term evolution of these algorithms have a behavior which is at variance with the diachronic change actually observed in historical linguistics. In particular, we saw how there was a tendency to gain rather than lose the V2 parameter setting. While this could be an artifact of the class of learning algorithms considered, a more likely explanation is that loss of V2 (observed in many of the world's languages like French etc.) is due to an interaction of parameters and triggers other than that considered in the previous section. We investigate this possibility and begin. by first providing the parametric theory.

5.4.1 The Parametric Subspace and Data

We now consider a syntactic space involving the following 5 (boolean-valued) parameters. We do not attempt to describe these parameters. The interested reader should consult Haegeman (1991) for details.

1. p_1: Case assignment under agreement ($p_1 = 1$) or not ($p_1 = 0$).
2. p_2: Case assignment under government ($p_2 = 1$) or not (($p_2 = 0$). Relevant triggers for this parameter include "Adv V S", "S V O".
3. p_3: Nominative clitics.
4. p_4: Null Subject. Here relevant triggers would include "wh V S O".
5. p_5: Verb-second V2. Triggers include "Adv V S" , and "S V O".

These 5 parameters now define a space of 32 parameterized grammars. Each grammar in this parameterized system can be represented by a string of 5 bits depending upon the values of $p_1, \ldots, p_5$. We need obviously to look at the surface strings (sentences) generated by each such grammar. For the purpose of explaining how Old French changed to Modern French over time, Clark and Roberts consider the following sentences. We provide these sentences below. The parameters settings which need to be made in order to generate each sentence is provided in brackets.

The Relevant Data;

adv V S [*1**1]; SVO [*1**1] or [1***0]: wh V S O [*1***]: wh V s O [**1**] ; X (pro)V O [*1*11] or [1**10]; X V s [**1*1]: X s V [**1*0]: X S V [1***0]; (s)VY [*1*11]

The parameter settings provided in brackets determine the grammars which generate the sentence. Thus the sentence (adv V S: *quickly ran John*– incorrect word order in English) is generated by all grammars which have case assignment under government ($p_2 = 1$) and verb second movement ($p_5 = 1$). The other parameters can be set to any value. Clearly there are 8 different grammars which can generate (parse) this sentence. Similarly there are 16 (8 corresponding to parameter settings of [*1**1] and 8 corresponding to parameter settings of [1***0]) grammars which generate (S V O) and 4 grammars which generate ((s) V Y).

Remark. Note that the set of sentences considered here is only a subset of the the total number of (degree-0) sentences generated by the 32 grammars in question. Clark and Roberts have only considered this subset and attempted to construct learning algorithms and models of diachronic change using genetic algorithms. In order to facilitate direct comparison with their results, we have not attempted to expand the data set or fill out the space any further. As a result, all the grammars do not have unique extensional properties, i.e. some generate the same sentences and are thus equivalent.

5.4.2 The Case of Diachronic Syntax Change in French

Within this parameter space, it is historically observed that the language spoken in France underwent a parametric change from the twelfth century to modern times. In particular. a loss of V2 and prodrop is observed. We provide two examples of this. In keeping with standard practice, the asterisk denotes an ungrammatical sentence.

Loss of null subjects: pro-drop

a. *Ainsi s'amusaient bien cette nuit. (Modern French)
thus (they) had fun that night.

b. Si firent (pro) grant joie la nuit. (Old French)
thus (they) made great joy the night.

Loss of V2

a. *Puis entendirent-ils un coup de tonerre. (Modern French)
then they heard a clap of thunder.

b. Lors oirent ils venir un escoiz de tonoire. (Old French)
then they heard come a clap of thunder

It has been argued that this transition was brought about by introduction of new word orders during the fifteenth and sixteenth centuries resulting in generations of children acquiring slightly different grammars and eventually culminating in the grammar of modern French. A brief reconstruction of the historical process (after Clark and Roberts. 1993) is provided.

Old French [11011] The language spoken in the twelfth and thirteenth centuries had verb-second movement and null subjects, both of which were dropped by the twentieth century. The set of sentences generated by the parameter settings corresponding to Old French are:

adv V S - [*1**1]: SVO - [*1**1] or [1***0]; wh V S O [*1***]; X (pro)V O [*1*11] or [1**10]

Note that from the data set, it appears that the Case agreement and nominative clitics parameters remain ambiguous. In particular, Old French is in a subset-superset relation with another language (generated by the parameter settings of 11111). Clearly some kind of subset principle (Berwick, 1985) has to be used by the learner for otherwise it is not clear how the data would allow the learner to converge to the Old French grammar in the first place. Note that TLA or TLA like schemes would not converge uniquely to the grammar of Old French.

The string (X)VS occurs with 58% and SV(X) occurs with 34% in Old French texts. It is argued that this frequency of (X)VS is high enough to cause the V2 parameter to trigger to +V2.

Middle French In Middle French, the data is not consistent with any of the 32 target grammars (equivalent to a heterogeneous population). Analysis of texts from that period reveal that some old forms (like Adv V S) decreased in frequency and new forms (like Adv S V) increased. It is argued in Clark and Roberts that such a frequency shift causes "erosion" of V2, brings about parameter instability and ultimately convergence to the grammar of Modern French. In this transition period (i.e. when Middle French was spoken/written) the data is of the following form:

adv V S [*1**1]; SVO [*1**1] or [1***0]; wh V S O [*1***]; wh V s O [**1**]; X (pro)V O [*1*11] or [1**10]: X V s [**1*1]: X s V [**1*0]: X S V [1***0]; (s)VY [*1*11]

Thus, we have old sentence patterns like Adv V S (though it decreases in frequency and becomes only 10%), SVO, X (pro)V O and whVSO. The new sentence patterns which emerge at this stage are adv S V (increases in frequency to become 60%), X subjclitic V, V subjclitic (pro)V Y (null subjects) , whV subjclitic O.

Modern French [10100] By the eighteenth century, French had lost both the V2 parameter setting as well as the null subject parameter setting. The sentence patterns consistent with Modern French parameter settings are SVO [*1**1] or [1***0], X S V [1***0], V s O [**1**]. Note that this data, though consistent with Modern French, will not trigger all the parameter settings. In this sense, Modern French (just like Old French) is not uniquely learnable from data. However, as before, we shall not concern ourselves overly with this, for the relevant parameters (V2 and null subject) are uniquely set by the data here.

5.4.3 Some Dynamical System Simulations

We can obtain dynamical systems for this parametric space, for a TLA (or TLA-like) algorithm in a straightforward fashion. We show the results of two simulations conducted with such dynamical systems.

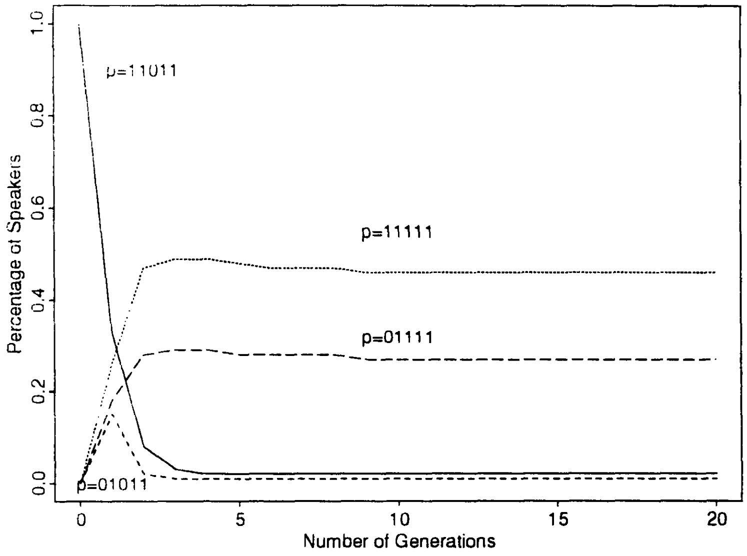

Figure 5-50: Evolution of speakers of different languages in a population starting off with speakers only of Old French.

Homogeneous Populations [Initial–Old French]

We conducted a simulation on this new parameter space using the Triggering Learning Algorithm. Recall that the relevant Markov chain in this case has 32 states. We start the simulation with a homogeneous population speaking Old French (parameter setting = 11011). Our goal was to see if misconvergence alone, could drive Old French to Modern French.

Just as before, we can observe the linguistic composition of the population over several generations. It is observed that in one generation, 15 percent of the children converge to grammar 01011; 18 percent to grammar 01111; 33 percent to grammar 11011 (target) and 26 percent to grammar 11111 with very few having converged to other grammars. Thereafter, the population consists mostly of speakers of these 4 languages, with one important difference: 15 percent of the speakers eventually *lose V2*. In particular, they have acquired the grammar 11110. Shown in fig. 5-50 are the percentage of the population speaking the 4 languages mentioned above as they evolve over 20 generations. Notice that in the space of a few generations, the speakers of 11011, and 01011 have dropped out altogether. Most of the population now speaks language 1111 (46 percent) and 01111 (27 percent). Fifteen percent of the population speaks 11110 and there is a smattering of other speakers. The population remains roughly stable in this configuration thereafter.

Observations:

1. On examining the four languages to which the system converges after one generation, we notice that they share the same settings for the principles [Case assignment under government], [pro drop], and [V2]. These correspond to the three parameters which are uniquely set by data from Old French. The other two parameters can take on any value. Consequently 4 languages are generated all of which satisfy the data from Old French.

2. Recall our earlier remark that due to insufficient data, there were equivalent grammars in the parameter system. It turns out that in this particular case, the grammars (01011) and (11011) are identical as far as their extensional properties are concerned; as are the grammars (11111) and (01111).

3. There is subset relation between the two sets described in (2). The grammar (11011) is in a subset relation with (11111). This explains why after a few generations most of the population switches to either (11111) or (01111) (the superset grammars).

4. An interesting feature of the simulation is that 15 percent of the population eventually acquires the grammar (11110), i.e., they have lost the V2 parameter setting. This is the first sign of instability of V2 that we have seen in our simulations so far (for greedy algorithms which are psychologically preferred). Recall that for such algorithms, the V2 parameter was very stable in our previous example.

Heterogeneous Populations (Mixtures)

The earlier section showed that with no new (foreign) sentence patterns the grammatical system starting out with only Old French speakers showed some tendency to lose V2. However, the grammatical trajectory did not terminate in Modern French. In order to more closely duplicate this historically observed trajectory, we examine alternative inital conditions. We start our simulations with an initial condition which is a mixture of two sources; data from Old French and data from New French (reproducing in this sense, data similar to that obtained from the Middle French period). Thus children in the next generation observe new surface forms. Most of the surface forms observed in Middle French are covered by this mixture.

Observations:

1. On performing the simulations using the TLA as a learning algorithm on this parameter space, an interesting pattern is observed. Suppose the learner is exposed to sentences with 90 percent generated by Old French grammar (11011) and 10 percent by Modern French grammar (10100), within one generation 22 percent of the learners have converged to the grammar (11110) and 78 percent to the grammar (11111). Thus the learners set each of the parameter values to 1 except the V2 parameter

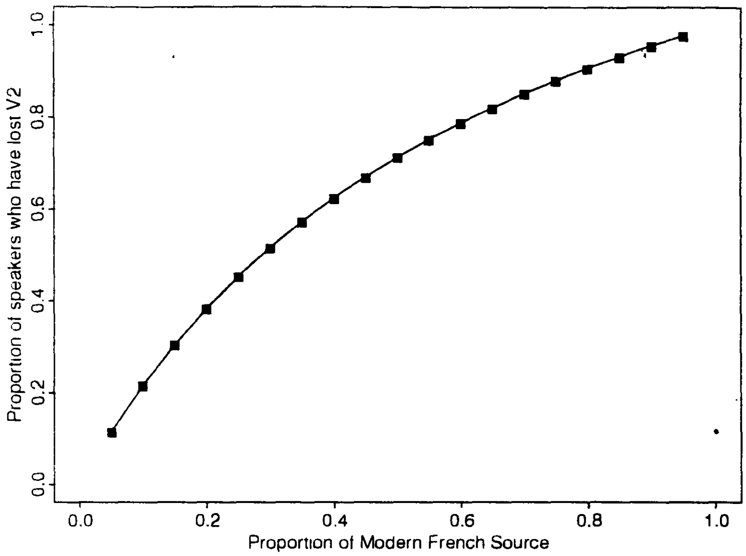

Figure 5-51: Tendency to lose V2 as a result of new word orders introduced by Modern French source in our Markov Model.

setting. Now Modern French is a non-V2 language; and 10 percent of data from Modern French is sufficient to cause 22 percent of the speakers to lose V2. This is the behavior over one generation. The new population (consisting of 78 percent speaking grammar (11111) and 22 percent speaking grammar (11110)) remains stable for ever.

2. Fig. 5-51 shows the proportion of speakers who have lost V2 after one generation, as a function of the proportion of sentences from the Modern French Source. The shape of the curve is interesting. For small values of the proportion of the Modern French source, the slope of the curve is greater than 1. Thus there is a greater tendency of speakers to lose V2 than to retain it. Thus 10 percent of novel sentences from the Modern French source causes 20 percent of the population to lose V2; similarly 20 percent of novel sentences from the Modern French source causes 40 percent of the speakers to lose V2. This effect wears off later. This seems to capture computationally the intuitive notion of many linguists that a small change in inputs provided to children could drive the system towards larger change.

3. Unfortunately, there are several shortcomings of this particular simulation. First, we notice that mixing Old and Modern French sources does not cause the desired (historically observed) grammatical trajectory from Old to Modern French (corresponding in our system to movement from state (11011) to state (10100) in our Markov Chain). Although we find that a small injection of sentences from Modern

French causes a larger percentage of the population to lose V2 and gain subject clitics (which are historically observed phenomena), nevertheless, the entire population retains the null subject setting and case assignment under government. It should be mentioned that Clark and Roberts argue that the change in case assignment under government is the driving force which allows alternate parse-trees to be formed and causes the parametric loss of V2 and null subject. In this sense, it is a more fundamental change.

4. If the dynamical system is allowed to evolve, it ends up in either of the two states (11111) or (11110). This is essentially due to the subset relations these states (languages) have with other languages in the system. Another complication in the system is the equivalence of several different grammars (with respect to their surface extensions) e.g. given the data we are considering, the grammars (01011) and (11011) (Old French) generate the same sentences. This leads to multiplicity of paths, convergence to more than one target grammar and general inelegance of the state-space description.

Future Directions: There are several possibilities to consider here.

1. Using more data and filling out the state-space might yield greater insight. Note that we can also study the development of other languages like Italian or Spanish within this framework and that might be useful.

2. TLA-like hill climbing algorithms do not pay attention to the subset principle explicitly. It would be interesting to explicitly program this into the learning algorithm and observe the evolution thereafter.

3. There are often cases when several different grammars generate the same sentences or atleast equally well fit the data. Algorithms which look only at surface strings are unable then to distinguish between them resulting in convergence to all of them with different probabilities in our stochastic setting. We saw an example of this for convergence to four states earlier. Clark and Roberts suggest an elegance criterion by looking at the parse-trees to decide between these grammars. This difference between *strong* generative capacity and *weak* generative capacity can easily be incorporated into the Markov model as well. The transition probabilities, now, will not depend upon the surface properties of the grammars alone, but also upon the elegance of derivation for each surface string.

4. Rather than the evolution of the population, one could look at the evolution of the distribution of words. One can also obtain bounds on frequencies with which the new data in the Middle French Period must occur so that the correct drift is observed.

5.5 Conclusions

In this chapter. we have argued that any combination of (grammatical theory, learning paradigm) leads to a model of grammatical evolution and diachronic change. A learning theory (paradigm) attempts to account for how children (the individual child) solve the problem of language acquisition. By considering a population of such "child learners". we have arrived at a model of the *emergent*, global, population behavior. The key point is that such a model is a logical consequence of grammatical, and learning theories. Consequently. whenever a linguist suggests a new grammatical, or learning theory. they are also suggesting a particular evolutionary theory—and the consequences of this need to be examined.

Historical Linguistics and Diachronic Criteria

From a programmatic perspective, this chapter has two important consequences. First, it allows us to take a formal, analytic view of historical linguistics. Most accounts of language change have tended to be descriptive in nature (though significant exceptions are the work of Lightfoot, Kroch, Clark and Roberts, among others). In contrast, we place the study of historical linguistics (diachronic phenomena) on a scientific[39] platform. In this sense, our conception of historical linguistics is closest in spirit to evolutionary theory and population biology[40] (which attempts to describe the origin and changing patterns of life) and cosmology (which attempts to describe the origin and evolution of the physical universe).

Second, it allows us to formally pose a *diachronic* criterion for the adequacy of grammatical theories. A significant body of work in learning theory, has already sharpened the *learnability* criterion for grammatical theories—in other words, the class of grammars $\mathcal{G}$ must be learnable by some psychologically plausible algorithm from primary linguistic data. Now we can go one step further. The class of grammars $\mathcal{G}$ (along with a proposed learning algorithm $\mathcal{A}$) can be reduced to a dynamical system whose evolution must match that of the true evolution of human languages (as reconstructed from historical data).

[39] By scientific, we mean, the construction of models with explanatory, and predictive powers–models which can be falsified in the sense of Popper.

[40] Indeed, most previous attempts to model language change, like that of Clark and Roberts (1993), and Kroch (1990) have been influenced by the evolutionary models.

In This Chapter

In this chapter, we have attempted to lay the framework for the development of research tools to study historical phenomena. To concretely demonstrate that the grammatical dynamical systems need not be impossibly difficult to compute (or simulate), we explicitly showed how to transform parameterized theories, and memoryless learning algorithms to dynamical systems. The specific simulations of this chapter are far too incomplete to have any long term linguistic implications. though, we hope, it certainly forms a starting point for research in this direction. Nevertheless, there were certain interesting results obtained in this chapter.

1. We saw that the V2 parameter was more stable in the 3-parameter case, than it was in the 5 parameter case. This suggests that the loss of V2 (actually observed in history) might have more to do with the choice of parameterizations than learning algorithms, or primary linguistic data (though, we suggest great caution, before drawing strong conclusions on the basis of this study).

2. We were able to shed some light on the time course of evolution. In particular, we saw how this was a derivative of more fundamental assumptions about initial population conditions, sentence distributions, and learning algorithms.

3. We were able to formally develop notions of system stability. Thus, certain parameters could change with time, others might remain stable. This can now be measured, and the conditions for stability or change can be investigated.

4. We were able to demonstrate how one could tinker with the system (by changing the algorithm, or the sentence distributions, or maturational time) to allow evolution in certain directions. This would suggest the kinds of changes needed in linguistics for greater explanatory adequacy.

Further Research

This has been our first attempt to define the boundaries of the problem. There are several directions of further research.

1. From a linguistic perspective. the most interesting thing to do. would perhaps be the examination of alternative parameterized theories. and to track the change of certain languages in the context of these theories (much like our attempt to track the change of French in this chapter). Some worthwhile attempts would include a) the study of parametric stress systems (Halle and Idsardi, 1992) and in particular, the evolution of modern Greek stress patterns from proto-Indo European; b) the investigation of the possibility that creoles correspond to fixed points in parametric dynamical systems, a possibility which might explain the striking fact that all creoles (irrespective of the linguistic origin, i.e., initial linguistic composition of the popula-

tion) have the same grammar; c) the evolution of modern Urdu, with Hindi syntax, and Persian vocabulary.

2. From a mathematical perspective, one could take this research in many directions including a) the formalization of the update rule for other grammatical theories and learning algorithms, and the characterization of the dynamical systems implied therein b) the investigation of stability issues more closely, and characterizing better the phase-space plots c) recall that our dynamical systems are multi-dimensional nonlinear iterated function mappings—a recipe for chaotic behavior, and a possibility to investigate further.

It is our hope that research in this line will mature to make useful contributions, both to linguistics, and in view of the unusual nature of the dynamical systems involved, to the study of such systems from a mathematical perspective.

Chapter 6
Conclusions

Abstract

This chapter concludes our thesis by articulating the perspective which emerges over the investigations of the previous chapters. We discuss the implications of some of our specific results, their role in illuminating our point of view, and directions for future research.

In this thesis, we investigated the problem of learning from examples. Implicit in any scientific investigation is a certain point of view— crucial to our point of view were:

1. the belief (and recognition) that the brain computes functions (input- output maps). Consequently, a function approximation framework is relevant, and in the context of learning, it is of some value to understand the complexity of learning to approximate (or identify) functions from examples.

2. a focus on the informational complexity of learning such functions. Roughly speaking , if one wishes to learn from examples, then how many examples does one need?

From this starting point, we proceeded to examine the informational complexity of learning from examples in a number of different contexts. Several themes have emerged over the course of this thesis.

6.1 Emergent Themes

Hypothesis Complexity: The number of examples needed depends upon the complexity of the hypothesis class used. In view of Vapnik and Chervonenkis' work (and numerous other works in statistics), this is reasonably well recognized (though people continue to flout this in the design of underconstrained models for learning systems).

More crucially, there is an inherent tension between the approximation error and estimation error, and a tradeoff between the two is involved whenever one chooses a model of a certain complexity. We demonstrated this explicitly in the case of feed-forward regularization networks, but the point is general. In language learning, this tension plays a crucial role, and guided our choice of the kinds of linguistic theories worth examining from a scientific perspective. We later investigated the *sample complexity* of learning within the principles and parameters framework of modern linguistics. It is worthwhile to observe that within this framework as well, the model complexity can be measured in some fashion, e.g. the number, and nature of the principles (parameters), the Kolmogorov complexity of the grammatical class, and so on. The exact nature of the relationship between this model complexity, and the number of examples needed to learn was not investigated explicitly, and remains an important area for further research.

Manner and Nature of Examples: The informational complexity of learning from examples, clearly depends upon nature of the examples, and the manner in which they are provided to the learner. In every case we have treated in this thesis, examples were (x, y) pairs consistent with some target function. There were slight differences, however, between the specific instances examined in the different chapters. For the case of regularization networks, these examples were contaminated with noise. For the case of languages investigated later, only positive examples were presented (i.e., all examples had the y-value of 1).

A more interesting observation to make on the question of examples is our inherently stochastic formulation of the problem. Examples were typically randomly drawn. This was according to some unknown distribution for regularization networks, and the language learner: and according to some known distribution in the case of the active function approximator (learner) of chapter 3. Such a stochastic formulation is very much in keeping with the spirit of PAC learning, which has influenced much of this work. Furthermore, it allows us to to take recourse to laws of large numbers, and better characterize rates of convergence, and thereby sample complexity.

A few further observations need to be made. First, a stochastic formulation is not always utilized for investigation of learning paradigms. For example, in typical language learning research in an inductive inference setting, the learner is required to converge on *every* training sequence. The rates of convergence of such a learner are hard to characterize unless one puts a measure on the training sequences. This brings us back to a probabilistic framework, and indeed, such extensions have been considered in the past. Second, we observed that if examples are chosen by the learner (rather than passively drawn), they could potentially learn faster. Of course, this need not always be the case, and explicit formal studies are required to decide one way or the other. Third, the actual number of examples required (in a passive setting) depends upon the distribution with which data is presented to the learner. In the case of regularization networks, we were able to obtain distribution-free bounds, but these are only bounds, and as various researchers have noted, are often weak. For language learning in finite parameter cases, we see this dependence on distributions explicitly. We notice here that no distribution-free bound exists.

The Learning Algorithm Used: The informational complexity of learning also depends upon the kind of algorithm the learner uses in making its hypotheses about the target. A poorly motivated algorithm might not even converge to the target (as the data goes to infinity), let alone do this in reasonable time. Again, this is not particularly surprising, and the point becomes vacuous without explicit characterization of the relationship (between algorithm and sample complexity) in some form. The degree of constraints on the learning algorithms we examined, varied from chapter to chapter. For the case of regularization networks, our results were valid for any algorithm which minimized a mean-square error term. Though, we did not prove it in the thesis, it turns out that algorithms minimizing cross entropy terms (for pattern classification) are covered in the analysis as well. In our investigation of active learning, we considered the approximation scheme (a component of the learning algorithm) explicitly. As a matter of fact, all comparisons between passive and active methods of data collection were made between learners using the same approximation scheme (thereby eliminating the influence of the approximation scheme on sample complexity). Active and passive learners represent two significantly different kinds of learning algorithms. We saw how to derive an active scheme from a passive one in a function approximation setting, and how such a scheme could then potentially reduce the informational complexity of learning. For language learning, we were able to show that all memoryless algorithms could be modeled as a Markov chain. However, the transition probabilities of these chains depended on the specific nature of the algorithms. We explicitly computed these transition matrices for a number of variants of the TLA (single step, greedy ascent) and showed how the sample complexity seemed to vary for the same task. It should also be noted that our analysis scheme in language learning (Markov chains) were derived from the learning algorithms used. In this sense, the sample complexity results were sharp (exact). This is in contrast to bounds on the sample complexity which can be obtained by using uniform convergence type arguments, or other techniques.

Learnability and Evolution: An important connection between learning systems and evolutionary systems emerged toward the end of this thesis. Both kinds of systems are adaptive ones. However, according to our analysis here, learning occurs at the level of the individual, evolution at the level of the population. Clearly, the two interact— and an information-theoretic point of view is important for an understanding of such interactions. The manner, nature, and number, of examples, the complexity of the hypothesis spaces, the learning algorithms used have implications for global evolutionary trajectories of populations of learners. In this sense, a theory of learning which attempts to explain individual behavior logically implies a certain group behavior. We demonstrated this connection explicitly for the human language system. This kind of evolutionary analysis of learning systems could serve as an important research tool in a number of different contexts. Certainly, in economic systems, one could examine the evolution (adaptation) of the global (macro) economy as a result of the behavior (also adaptive) of the individual economic agents.

6.2 Extensions

The previous section described the broad results, and the emergent perspective of this thesis. With this perspective, one could proceed in several directions.

1. *Model Selection:* At a fundamental level, one could examine the question of model selection in general. In the cases we considered, the models (family of functions, or hypothesis classes) were homogeneous (in fact, often parameterized) , i.e., all functions in our hypothesis class had the same representation (as regularization networks, parameterized grammars, or spline functions etc.). The task of learning reduced primarily to the task of estimating the values of the parameters. What if we have qualitatively different kinds of models? Instead of choosing the best hypothesis $h \in \mathcal{H}$ as all our learning problems were posed, what if we were interested in choosing the best class $\mathcal{H} \in \mathcal{H}_{super}$? To make matters a little concrete, suppose one were interested not in ***choosing the best regularization network*** for a certain problem, but in deciding ***whether regularization networks were best*** for that problem (other candidate models might include multi-layer perceptrons, or polynomials, etc..)? In the case of languages, one might be interested in choosing between bigrams, context-free grammars, parameterized theories, etc. How does one characterize the complexity of the super class $\mathcal{H}_{super}$ whose individual elements are not functions but classes of functions (models)? For that matter, how does one measure the distance between two models, i.e., $d(\mathcal{H}_1, \mathcal{H}_2)$? This matter is of some interest in recent times as increased computational power has made it possible for researchers to literally "throw models at the data" in their frantic search for "good" ones?

This is also a subject of interest to researchers in the field of *data mining.*

2. *Informational Complexity of Grammars:* Another fruitful area of research is the informational complexity of learning grammars. As has been mentioned earlier, most language learning research tends to focus on the Gold paradigm of identification in the limit, without due attention to the rates at which the learner attains the target. Given, the arguments of "poverty of stimulus" invoked in the modern approach to linguistics, an informational perspective is bound to be of some value in choosing between alternate theories. For example, what is the sample complexity of learning bigrams, trigrams, lexical-functional grammars, metrical stress patterns, optimality theory etc.? How does it depend upon the algorithms used, noise, sentence distributions? What are psychologically plausible algorithms? What are "real" sentence distributions like? Quantitative answers to these questions would considerably aid the search for the right linguistic theory. One could also potentially decompose the language learning problem into approximation and estimation parts. For example, by analogy with our analysis of regularization networks, we can pose the following simple problem. Let M_n be class of all finite state grammars with at most n states (analogous to H_n: networks with at most n hidden units). Let $M = \cup_{n=1}^{\infty} M_n$. Let $\mathcal{M}$ (analogous to $\mathcal{F}$) be some class which can be approximated by M (it could simply be M itself). Let examples be sentences drawn from some target grammar $m \in \mathcal{M}$. Then how many states (n) must we have, and how many examples must we draw so that with high confidence, the learner's grammar (extensionally) is ϵ close to m with high confidence?

3. *Evolutionary Systems:* We argued, in chapter 5, that specific assumptions about linguistic theories, and learning paradigms, leads automatically to a model of language change. We were able to transform memoryless algorithms operating on finite parameter spaces into explicit dynamical systems. There are several interesting directions to pursue within this area of research. First, one could attempt to obtain similar dynamical systems corresponding to other assumptions about linguistic theories, and learning algorithms. Second, from a purely mathematical perspective, it would be interesting to study the classes of dynamical systems motivated by linguistics. For example, we saw that the the systems for finite parameter spaces were non-linear, and multi-dimensional. The mathematical characterization of such systems are far from trivial. Finally, of course, one must attempt to put such evolutionary models to good scientific use, by validating against real cases of language change. Such an enterprise, will hope-

fully result in a mathematically productive, and scientifically fruitful study of historical linguistics. In general, the connection between ***individual learning***, and *group evolution* is an interesting one. It can be studied in other contexts, and formal connections between learning theory, and evolutionary theory need to be developed further.

4. *Computational Complexity:* This thesis focused almost exclusively on the number of examples needed so that the learner's hypothesis is close to the target. The computational complexity of choosing a hypothesis (once the requisite number of examples have been provided) is a matter of great importance, and largely ignored in this thesis. For example, our main theorem in Chapter 2 assumes that the learner will be able to find the global minimum of the mean-square error term. In general, this problem, as we have noted, is likely to be NP-hard. Similarly, in Chapter 3, the active learner reduces the informational complexity at the cost of increasing the computational burden. For the cases we examined, an analytical solution to the sequential optimal recovery problem allowed us to obtain tractable solutions. In general, however, the complexity of solving the optimal recovery equations (and recovering the optimal point to sample at each stage) could well be intractably high. Further, in the case of language learning, we obtained sample complexity bounds which were tuned to specific algorithms known to be feasible, and psychologically plausible. These algorithms, of course, don't learn every possible parameterized space. The complexity of learning a parameterized space, in general, could well be NP-hard (scalability with respect to number of parameters, and examples). These are directions worth pursuing. After all, a truly realistic cognitively plausible theory of human learning should require not only a feasible number of examples, but should also have low computational (cognitive) burden.

Bibliography

[1] D. E. Rumelhart A. S. Weigand and B. A. Huberman. Generalization by weight elimination with applications to forecasting. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[3] W. Arai. Mapping abilities of three-layer networks. In *Proceedings of the International Joint Conference on Neural Networks*, pages I–419–I–423, Washington D.C., June 1989. IEEE TAB Neural Network Committee.

[4] M. L. Athavale and G. Wahba. Determination of an optimal mesh for a collocation-projection method for solving two-point boundary value problems. *Journal of Approximation Theory*, 25:38–49, 1979.

[5] R.L. Rivest B. Eisenberg. On the sample complexity of pac-learning using random and chosen examples. In *Proceedings of the 1990 Workshop on Computational LearningTheory*, pages 154–162, San Mateo, CA, 1990. Morgan Kaufmann.

[6] A. Barron and T. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4), 1991.

[7] A.R. Barron. Approximation and estimation bounds for artificial neural networks. Technical Report 59, Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, March 1991.

[8] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. Technical Report 58, Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, March 1991.

[9] A.R. Barron. Approximation and estimation bounds for artificial neural networks. Technical Report 59. Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, March 1991a.

[10] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transaction on Information Theory*, 39(3):930–945, May 1993.

[11] E. B. Baum and D. Haussler. What size net gives valid generalization? In *IEEE Int. Symp. Inform. Theory*, Kobe, Japan, June 1988.

[12] E. B. Baum and D. Haussler. What size net gives valid generalization? In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems I*, pages 81–90. Morgan Kaufmann Publishers, Carnegie Mellon University, 1989.

[13] R.E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

[14] J. Berntsen, T. O. Espelid, and A. Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software*, 17(4), December,1991.

[15] R. C. Berwick. *The Acquisition of Syntactic Knowledge*. MIT Press, 1985.

[16] A. Blum and R. L. Rivest. Training a three-neuron neural net is np-complete. In *Proceedings of the 1988 Workshop on Computational LearningTheory*, pages 9–18, San Mateo, CA, 1988. Morgan Kaufma.

[17] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Classifying learnable geometric concepts with the vapnik-chervonenkis dimension. In *Proc. of the 18th ACM STOC, Berkeley, CA*, pages 273–282, 1986.

[18] S. Botros and C. Atkeson. Generalization properties of Radial Basis Function. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[19] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[20] R. Brunelli and T. Poggio. HyperBF Networks for Gender Classification. In *Proceedings of the Image Understanding Workshop*, pages 311–314, San Mateo, CA, 1992. Morgan Kaufmann.

[21] N. Chomsky. *Aspects of the Theory of Syntax.* Cambridge, MA: MIT Press, 1965.

[22] N. Chomsky. *Lectures on Government and Binding.* Dordecht: Reidel, 1981.

[23] N. Chomsky and M Schutzenberger. The algebraic theory of context-free languages. *Computer Programming and Formal Systems*, pages 53–77, 1963.

[24] C. K. Chui and X. Li. Approximation by ridge functions and neural networks with one hidden layer. CAT Report 222, Texas A and M University, 1990.

[25] R. Clark and I. Roberts. A computational model of language learnability and language change. *Linguistic Inquiry*, 24(2):299–345, 1993.

[26] D. Cohn. Neural network exploration using optimal experiment design. AI memo 1491, Massachusetts Institute of Technology, 1994.

[27] D. Cohn and G. Tesauro. Can neural networks do better than the vc bounds. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, pages 911–917, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[28] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross validation. *Numer. Math*, 31:377–403, 1979.

[29] G. Cybenko. Approximation by superposition of a sigmoidal function. *Math. Control Systems Signals*, 2(4):303–314, 1989.

[30] C. deMarcken. Parsing the lob corpus. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, PA: ACL, 243-251, 1990.

[31] R. DeVore, R. Howard, and C. Micchelli. Optimal nonlinear approximation. *Manuskripta Mathematika*, 1989.

[32] R.A. DeVore. Degree of nonlinear approximation. In C.K. Chui, L.L. Schumaker, and D.J. Ward, editors, *Approximation Theory, VI*, pages 175–201. Academic Press, New York, 1991.

[33] R.A. DeVore and X.M. Yu. Nonlinear n-widths in Besov spaces. In C.K. Chui, L.L. Schumaker, and D.J. Ward, editors, *Approximation Theory, VI*, pages 203–206. Academic Press, New York, 1991.

[34] L. Devroye. On the almost everywhere convergence of nonparametric regression function estimate. *The Annals of Statistics*, 9:1310–1319, 1981.

[35] B. E. Dresher and J. Kaye. A computational learning model for metrical phonology. *Cognition*. pages 137–195. 1990.

[36] R. M. Dudley. *Real analysis and probability*. Mathematics Series. Wadsworth and Brooks/Cole. Pacific Grove. CA. 1989.

[37] R.M. Dudley. Universal Donsker classes and metric entropy. *Ann. Prob.*, 14(4):1306–1326. 1987.

[38] R.M. Dudley. Comments on two preprints: Barron (1991), Jones (1991). Personal communication. March 1991.

[39] N. Dyn. Interpolation of scattered data by radial functions. In C.K. Chui, L.L. Schumaker, and F.I. Utreras, editors, *Topics in multivariate approximation*. Academic Press, New York, 1987.

[40] N. Tishby E. Levin and S. A. Solla. A statistical approach to learning and generalization in layered neural networks. *Proceedings of the IEEE*, 78(10):1568–1574, October 1990.

[41] B. Efron. The jacknife, the bootstrap, and other resampling plans. *SIAM, Philadelphia*, 1982.

[42] B. Eisenberg. Sample complexity of active learning??? Master's thesis, Massachusetts Institute of Technology, Cambridge,MA, 1992.

[43] G. Altmann et al. A law of change in language. In B. Brainard, editor, *Historical Linguistics*, pages 104–115, Studienverlag Dr. N. Brockmeyer., 1982. Bochum, FRG.

[44] R.L. Eubank. *Spline Smoothing and Nonparametric Regression*, volume 90 of *Statistics, textbooks and monographs*. Marcel Dekker, Basel, 1988.

[45] R. Frank and S. Kapur. On the use of triggers in parameter setting. Tech. Report 92-52. IRCS, University of Pennsylvania, 1992.

[46] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[47] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[48] E. Gibson and K. Wexler. Triggers. *Linguistic Inquiry*, 25, 1994.

[49] F. Girosi. On some extensions of radial basis functions and their applications in artificial intelligence. *Computers Math. Applic.*, 24(12):61–80, 1992.

[50] F. Girosi and G. Anzellotti. Rates of convergence of approximation by translates. A.I. Memo 1288, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

[51] F. Girosi and G. Anzellotti. Rates of convergence for radial basis functions and neural networks. In R.J. Mammone, editor, *Artificial Neural Networks for Speech and Vision*, pages 97–113, London, 1993. Chapman & Hall.

[52] F. Girosi, M. Jones, and T. Poggio. Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines. A.I. Memo No. 1430, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[53] H. Gish. A probabilistic approach to the understanding and training of neural network classifiers. In *Proceedings of the ICASSP-90*, pages 1361–1365, Albuquerque, New Mexico, 1990.

[54] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[55] Z. Govindarajulu. *Sequential Statistical Procedures*. Academic Press, 1975.

[56] U. Grenander. On empirical spectral analysis of empirical processes. *Ark. Matemat.*, 1:503–531, 1951.

[57] L. Haegeman. *Introduction to Government and Binding Theory*. Blackwell: Cambridge, USA, 1991.

[58] M. Halle and W. Idsardi. General properties of stress and metrical structure. In *DIMACS Workshop on Human Language*, Princeton, NJ, 1991.

[59] H. Hamburger and Kenneth Wexler. A mathematical theory of learning transformational grammar. *Journal of Mathematical Psychology*, pages 137–177, 12.

[60] R.L. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res*, 76:1905–1915, 1971.

[61] R.L. Hardy. Theory and applications of the multiquadric-biharmonic method. *Computers Math. Applic.*, 19(8/9):163–208, 1990.

[62] E. Hartman, K. Keeler, and J.M. Kowalski. Layered neural networks with gaussian hidden units as universal approximators. (submitted for publication), 1989.

[63] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. Technical Report UCSC-CRL-91-02, University of California, Santa Cruz, 1989.

[64] J.A. Hertz, A. Krogh, and R. Palmer. *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, CA, 1991.

[65] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[66] J. B. Hampshire II and B. A. Pearlmutter. Equivalence proofs for multilayer perceptron classifiers and the bayesian discriminant function. In J. Elman D. Touretzky and G. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, CA, 1990. Morgan Kaufman.

[67] B. Irie and S. Miyake. Capabilities of three-layered Perceptrons. *IEEE International Conference on Neural Networks*, 1:641–648, 1988.

[68] D. Isaacson and J. Masden. *Markov Chains.* John Wiley, New York, 1976.

[69] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. F. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[70] L.K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistic*, 1990. (to appear).

[71] L.K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for Projection Pursuit Regression and neural network training. *The Annals of Statistics*, 20(1):608–613, March 1992.

[72] S. Judd. *Neural Network Design and the Complexity of Learning.* PhD thesis, University of Massachusetts, Amherst, Amherst, MA, 1988.

[73] S. Kapur. *Computational Learning of Languages.* PhD thesis, Cornell University, Ithaca, NY, 1992.

[74] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 1993 STOC*, pages 392–401, 1993.

[75] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *Proceedings of the 1990 FOCS*, pages 382–391, 1990.

[76] D. Kimber and P. M. Long. The learning complexity of smooth functions of a single variable. In *Proceedings of the 1992 Workshop on Computational Learning Theory*, pages 153–159, San Mateo, CA, 1992. Morgan Kaufmann.

[77] A. S. Kroch. Grammatical theory and the quantitative study of syntactic change. In *Paper presented at NWAVE 11, Georgetown Universtiy*, 1982.

[78] A. S. Kroch. Function and gramar in the history of english: Periphrastic "do.". In Ralph Fasold, editor, *Language change and variation*. Amsterdam:Benjamins. 133-172, 1989.

[79] Anthony S. Kroch. Reflexes of grammar in patterns of language change. *Language Variation and Change*, pages 199–243, 1990.

[80] A. Krzyzak. The rates of convergence of kernel regression estimates and classification rules. *IEEE Transactions on Information Theory*, IT-32(5):668–679, September 1986.

[81] S. Kulkarni, S.K. Mitter, and J.N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, (to appear).

[82] A. Lapedes and R. Farber. How neural nets work. In Dana Z. Anderson, editor, *Neural Information Processing Systems*, pages 442–456. Am. Inst. Physics, NY, 1988. Proceedings of the Denver, 1987 Conference.

[83] D. Lightfoot. *How to Set Parameters*. MIT Press, Cambridge, MA, 1991.

[84] H. Linhart and W. Zucchini. *Model Selection*. John Wiley and Sons,, 1986.

[85] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22, April 1987.

[86] N. Littlestone, P. M. Long, and M. K. Warmuth. On-line learning of linear functions. In *Proceedings of the 1991 STOC*, pages 465–475, 1991.

[87] G. G. Lorentz. Metric entropy, widths, and superposition of functions. *Amer. Math. Monthly*, 69:469–485, 1962.

[88] G. G. Lorentz. *Approximation of Functions*. Chelsea Publishing Co., New York, 1986.

[89] D. Mackay. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, Pasadena, CA, 1991.

[90] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman and Company, San Francisco, 1982.

[91] H.N. Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1:61–80, 1993.

[92] H.N. Mhaskar and C.A. Micchelli. Approximation by superposition of a sigmoidal function. *Advances in Applied Mathematics*, 13:350–373, 1992.

[93] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[94] C. A. Micchelli and G. Wahba. Design problems for optimal surface interpolation. In Z. Ziegler, editor, *Approximation theory and applications*, pages 329–348. Academic Press, New York, 1981.

[95] C.A. Micchelli and T.J. Rivlin. A survey of optimal recovery. In C.A. Micchelli and T.J. Rivlin, editors, *Optimal Estimation in Approximation Theory*, pages 1–54. Plenum Press, New York, 1976.

[96] T. M. Mitchell, J. G. Carbonell, and R. S. Michalski. *Machine Learning: A Guide to Current Research.* Kluwer Academic Publishers, 1986.

[97] J. Moody. The VC dimension versus the statistical capacity of multilayer networks. In S. J. Hanson J. Moody and R. P. Lippman, editors, *Advances in Neural information processings systems 4*, pages 928–935, San Mateo, CA, 1992. Morgan Kaufman.

[98] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[99] John E. Moody. The effective number of parameters: An analysis of generalization and regularization in non-linear learning systems. In S. J. Hanson J. Moody and R. P. Lippman, editors, *Advances in Neural information processings systems 4*, pages 847–854, San Mateo, CA, 1991. Morgan Kaufman.

[100] P. Niyogi. Active learning of real valued functions. Preprint, 1993.

[101] P. Niyogi and R. C. Berwick. Formalizing triggers: A learning model for finite parameter spaces. Tech. Report 1449, AI Lab., M.I.T.. 1993.

[102] M. Opper and D. Haussler. Calculation of the learning curve of bayes optimal class algorithm for learning a perceptron with noise. In *Proceedings of COLT, Santa Cruz, CA*, pages 75–87, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[103] D. Osherson, M. Stob, and S. Weinstein. *Systems That Learn*. MIT Press, Cambridge, MA, 1986.

[104] A. Pinkus. *N-widths in Approximation Theory*. Springer-Verlag, New York, 1986.

[105] G. Pisier. Remarques sur un resultat non publiè de B. Maurey. In Centre de Mathematique, editor, *Seminarie d'analyse fonctionelle 1980–1981*, Palaiseau, 1981.

[106] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), September 1990.

[107] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

[108] T. Poggio and F. Girosi. Networks for Approximation and Learning. In C. Lau, editor, *Foundations of Neural Networks*, pages 91–106. IEEE Press, Piscataway, NJ, 1992.

[109] D. Pollard. *Convergence of stochastic processes*. Springer-Verlag, Berlin, 1984.

[110] M.J.D. Powell. The theory of radial basis functions approximation in 1990. Technical Report NA11, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, December 1990.

[111] S. Resnick. *Adventures in Stochastic Processes*. Birkhauser, 1992.

[112] M. D. Richard and R. P. Lippman. Neural network classifier estimates bayesian a-posteriori probabilities. *Neural Computation*, 3:461–483, 1991.

[113] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11:416–431, 1983.

[114] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[115] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, October 1986.

[116] M. Stone. Cross-validatory choice and assessment of statistical predictors(with discussion). *J. R. Statist. Soc.*, B36:111–147, 1974.

[117] S. Strogatz. *Nonlinear Dynamics and Chaos.* Addison-Wesley, 1993.

[118] K. K. Sung and P. Niyogi. An active formulation for approximation of real valued functions. In *Advances in Neural information processings systems 7 (to appear)*. 1994.

[119] R. W. Liu T. P. Chen, H. Chen. A constructive proof of approximation by superposition of sigmoidal functions for neural networks. Preprint, 1990.

[120] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[121] A.F. Timan. *Theory of approximation of functions of a real variable.* Macmillan, New York, 1963.

[122] J. F. Traub, G. W. Wasilkowski, and H. Wozniakovski. *Information Based Complexity.* Academic Press, New York, 1988.

[123] L.G. Valiant. A theory of learnable. *Proc. of the 1984 STOC*, pages 436–445, 1984.

[124] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer-Verlag, Berlin, 1982.

[125] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequences of events to their probabilities. *Th. Prob. and its Applications*, 17(2):264–280, 1971.

[126] V.N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for the uniform convergence of averages to their expected values. *Teoriya Veroyatnostei i Ee Primeneniya*, 26(3):543–564, 1981.

[127] V.N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

[128] G. Wahba. *Splines Models for Observational Data.* Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

[129] K. Wexler and P. Culicover. *Formal Principles of Language Acquisition.* MIT Press, Cambridge, MA, 1980.

[130] H. White. Connectionist nonparametric regression: Multilayer perceptrons can learn arbitrary mappings. *Neural Networks*, 3(535-549), 1990.

7'57-1