MINIMUM INFORMATION ESTIMATION OF STRUCTURE

by

GEORGE WILLIAM HART

B.S., Massachusetts Institute of Technology, (1977)
M.A., Indiana University, (1979)

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

June 1987

Signature of Author _____
Department of Electrical Engineering and Computer Science
May 1, 1987

Certified by _____
Professor Fred C. Schweppe
Thesis Supervisor

Certified by _____
Assistant Professor John N. Tsitsiklis
Thesis Supervisor

Accepted by _____
Professor Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

MINIMUM INFORMATION ESTIMATION OF STRUCTURE

by

George W. Hart

Submitted to
The Department of Electrical Engineering and Computer Science
May 1, 1987, in partial fulfillment of the
requirements for the degree of Doctor of Philosophy

ABSTRACT

In this thesis we formulate and consider the problem of estimation for applications where the structure of a system is the major unknown to be determined. We propose a logical structure for estimation theory to deal with problems in which the output of an estimator is a formal object such as a graph, a tree, a set, a sequence, or other non-vector structure.

The major result is a framework for approaching a wide range of difficult estimation problems that focuses on *formal language descriptions* of the possible structures and inputs, and measures of the *information* content of such descriptions. A *Minimum Information* (MI) Estimator is developed which chooses a structure that minimizes an information measure over a set of possible descriptions.

The approach can be viewed as a method for dealing with the computational difficulties of algorithmic information theory, and Solomonoff's theory of induction. It is a logical development of the work on information criteria of Wallace and Boulton, Rissanen, and others, but reorganized to center on formal language based descriptions using phrase structured grammars. This technique provides a very general framework suitable for diverse applications in many fields in which information needs to be extracted from data, including detection, estimation in vector spaces, signal processing, system identification, inference, induction, pattern recognition, and artificial intelligence.

In many ways, the description-based MI technique provides a unifying view of these fields. A single general estimator can be tailored as required by the problem domain and the types of data, models, and *a priori* information particular to it. It accommodates either probabilistic or nonprobabilistic models of structure distribution. The issue of model order, or structural complexity, is treated in a uniform manner which addresses the essential tension in the estimation of structure: *simple models* versus *good fit* to data.

Many existing criteria for selecting structures are examined and seen to be special cases of the proposed criteria. Case studies are presented in which estimators are developed for finite state machines, Markov sources, piecewise constant signals, clusters in scatter plots, and visual images. Simulations, with excellent results, are presented showing the estimated structures.

Thesis Supervisors:

Dr. Fred C. Schweppe, Professor of Electrical Engineering
Dr. John N. Tsitsiklis, Assistant Professor of Electrical Engineering

ACKNOWLEDGMENTS

It is a pleasure to acknowledge many people who have contributed in very many ways. First and foremost, I am grateful to my thesis committee, co-chaired by Professors Fred Schweppe and John Tsitsiklis, and with Professor Alvin Drake as reader, for their guidance, support, encouragement and receptivity. They have each contributed far more than they realize to this work.

Thanks go also to Professors Ron Rivest, Al Willsky, Lennart Ljung, Gary Marx, Peter Elias, and Bob Gallager, along with Ray Solomonoff, Chee-Seng Chow, Ed Puckett, Meir Feder, Rob Enders, Jim Anderson, and Howard Branz for many helpful discussions, criticisms, and suggestions, always widening my perspective. I am especially thankful to Chow for a careful reading of the manuscript, resulting in many improvements.

I gratefully acknowledge the Electric Power Research Institute (EPRI) for their patient financial support of such broadly ranging research, and to Professor Fred Schweppe and Ralph Abbott for their efforts in continuing it.

Special intellectual debts are owed to Professors F. Roger Higgins and Noam Chomsky for stimulating my interest in formal systems and a broader range of structural issues. John Solman at Lincoln Laboratory must be credited for first helping me to decipher the residential power consumption plots which eventually led to this thesis. Finally, I must thank Dr. Edward C. Kern for providing the impulse which returned me to graduate school.

CONTENTS

LIST OF FIGURES

Chapter 1

DESCRIPTION-BASED ESTIMATION

1.1 The Introduction

There are many structure estimation problems which classical estimation theory can not solve. In this thesis we create a category for problems of estimation in complex spaces, where system structure is the major unknown to be determined. We present a framework for estimation when the output of an estimator is a non-vector formal structure, such as a graph, a tree, a set, or a sequence, and demonstrate how it offers a new perspective in a variety of applications. The major result of this thesis is a framework for approaching a wide range of difficult estimation problems that focuses on *formal language descriptions* of the possible structures and inputs, and *measures of the information content* of such descriptions. A *Minimum Information* (MI) Estimator is presented which chooses a description that minimizes an information measure over a set of possible descriptions.

A virtue of this approach is that it forces the analyst to focus on the fundamental issue of structural complexity. The issue of model order, or structural

complexity, is treated in a uniform manner which addresses the essential tension in the estimation of structure: *simple models* versus *good fit* to data. Once put in these terms, appropriate ways to add structure to ill-posed problems become apparent.

The class of problems we consider are commonly found in a wide range of applications, but have not previously been grouped in one framework. The general field of pattern recognition perhaps has the greatest overlap. We will deal with discrete spaces of interrelated possibilities, for which probabilistic models are not always pertinent, and classical estimation techniques generally fail. The most familiar structure estimation problem to scientists and engineers is probably that of least-squares curve-fitting a polynomial to measurements. The structure to estimate is a sequence of coefficients, of unknown length. The more complex structures of higher-order polynomials can always be made to fit the data better than simple structures.

This class of structure-estimation problems differs fundamentally from classical estimation problems in vector spaces for two reasons:

1) Classical statistical inference techniques within an n-dimensional vector space pre-constrain the number of parameters to be determined as n. However, in typical problems in which structures are to be estimated, there is no upper bound on the allowable complexity of the estimate. The number of elements in an appropriate tree or graph must depend upon the data. This requires that attention be focused on the problem of selecting an appropriate level of *complexity* in the estimate.

2) Spaces of structures generally lack the algebraic properties of vector spaces. Graphs and trees, for example, can not be meaningfully added, subtracted, or scaled by real numbers. The usual notions of expectations, bias, variance, and optimization by differentiation are therefore not available.

There is a third property, typical of structure estimation problems, but not essential to them. In most situations where structure is to be estimated, the original problem formulation is ill-posed, because it is underdetermined. Additional constraints must be imposed in order to formulate a well-defined problem. Our problem formulation provides a framework for adding these constraints in a manner which results in satisfactory estimates.

The overall framework presented below is closely tied to the notions of Algorithmic Information Theory (AIT), especially Solomonoff's work on induction. However, our focus is on practical methods for estimation, rather than the uncomputable notions of AIT. A comparison between the two frameworks is given in Section 8.2.7. A major difference is that we allow the class of structures of interest to be specified, rather than restricting ourselves to the Turing Machines of AIT. In this regard, the approach used is similar to the information criteria of Wallace and Boulton, and Rissanen, but more versatile, in that it centers on formal-language-based descriptions. This technique provides a very general and adaptable framework suitable for diverse applications in many fields in which information needs to be extracted from data, including:

1. Detection
2. Estimation in Vector Spaces
3. Signal Processing
4. System Identification
5. Inference
6. Induction
7. Pattern Recognition
8. Artificial Intelligence

In many ways, the description-based MI technique provides a unifying view of these fields. A single general estimator can be tailored as required by the problem domain and the types of data, models, and *a priori* information particular to it.

With appropriate choices for its parameters, it reduces to well known techniques for estimating structure as special cases (e.g. hypothesis testing, finding the "knee" of the curve, MAP and ML estimators). It can therefore be viewed as interpolating between these many estimation techniques.

The most important contribution of this work may be in the problem formulation itself. We emphasize that a wide range of problems which are not normally posed as structural estimation problems should be thought of in those terms. Because structural estimation problems involve rather different issues from classical estimation problems, one is then led to focus on the issues of complexity and fit within the context of the problem. A description-based framework offers a fresh point of view for finding a solution. An abbreviated list of problems which might be reexamined in these terms is included in the final chapter.

Our goal throughout is twofold. First, we wish to explore the fundamental issues and relations in structure estimation from as wide a range of perspectives as possible. Secondly, we wish to illustrate practical techniques which we feel can be applied to solve many real-world engineering problems. We will occasionally speak of "the system designer" when emphasizing decision points where options must be selected according to the requirements of an application.

1.2 The Logical Structure of Estimation Theory

In developing a theory of estimation for structures, we wish to take as much as possible from the large body of classical statistical inference techniques for vector spaces. As emphasized above, many of these techniques are not suitable for more general application because they rely on the special algebraic properties of vector

spaces, and require that the estimated quantity have a prespecified number of degrees of freedom.

Upon examination, four general characteristics of estimation emerge when methods peculiar to vector spaces are eliminated from the large body of classical methods. These four components form the foundation of our approach.

1. *Description Mechanism.* A formal representation system must be available which provides a description for each of the elements in the domain of possible estimates. It must also allow the set of possible observations, which serve as inputs to the estimator, to be described.
2. *Interpretation.* Associated with each of the possible structures and observations must be an interpretation relative to an application. The interpretation generally includes some notion of simplicity of models relative to each other and of the degree of fit between particular models and particular observations.
3. *Information Measures.* In order to formalize and to effect a tradeoff between simplicity of a structure and its degree of fit with the data, it must be possible to measure both of these quantities with a scalar measure on the set of descriptions. The sum of these measures is then an information criterion to be minimized over the set of possible descriptions.
4. *Optimization Techniques.* Some method must be selected for finding either the optimal structure by the criterion, or an approximately optimal one.

A novel aspect of the MI method, compared to classical estimation techniques, is the primary role which is given to the function of *description.* We justify this by claiming that a formal description system is essential to any estimator. The output of any estimator must be a *description* of its estimate. While the importance of descriptions has always been emphasized in the field of artificial intelligence, it has received little notice in the large body of classical estimation techniques used in vector spaces. This is because the language for describing points in a vector space is simply the language of ordered n-tuples, which is so familiar and appropriate to the domain that its role as a descriptive language is overlooked.

In examining estimation problems in spaces with less structure, the need for a descriptive language becomes immediately apparent. An estimator which outputs, for example, a finite-state machine, or an analysis of a digitized visual image, must incorporate a language appropriate to the types of structures found in the domain. The output of the estimator must be a description of an object in the domain. With this point of view, returning to classical estimation techniques, we see the outputs of all estimators as statements in an appropriate description language. For example, in signal processing, the Karhunen-Loève projection of input signals onto an orthogonal basis of waveforms can be viewed as providing an interpretation which allows signals to be described using the language of ordered n-tuples. We develop other languages for describing signals in Chapter 6.

Because a unified approach to estimation requires a flexible description mechanism, a rigorous approach will require the application of formal languages. A natural choice (but not the only choice) of formal language model for this purpose is the phrase-structured grammar (PSG). A PSG allows sentences to be formed compositionally from expressions, which are in turn formed from sub-expressions, etc., down to a level of primitive tokens. This formalism is flexible enough to provide descriptions of the elements of many complex spaces. Some objects are simply described with a short expression, while others require long detailed descriptions. If, on the other hand, the space has the property that all the objects can be described with expressions of the same length and construction (e.g. n-tuples), this is easily accommodated also.

An important property of PSGs is that their compositional organization allows complex descriptions of complex structures to be formed hierarchically, by conjoining descriptions of the substructures of the object. Trees can be described

by conjoining descriptions of their subtrees; graphs can be described by conjoining descriptions of their arcs, nodes or subgraphs. In this respect, PSGs are a good model of the constructions used in natural languages. Conversely, the types of natural language constructs which automatically come to mind when describing a space or its elements can be invoked and adapted in the design of the formal languages required for an estimator in that space. Thus the design of formal languages for applications will be seen to be a very natural process.

Once designed, the PSG also aids in defining neighborhood relations in the set of structures. The notion that structures with similar descriptions are themselves similar provides an organization to the space of structures which can be exploited in optimization. Local search techniques, based on syntactically defined neighborhood relations, are used to minimize the MI criterion in the various case studies.

The second essential component of our estimators is an information measure. Intuitively, this will be a measure of the complexity of a description. It is formalized as a function from descriptive sentences to the real numbers. Clearly, some function in this class must be involved in any estimator which is defined by the optimization of a scalar criterion. Generally, we will define information measures such that long complex descriptions have a high measure of information, and short descriptions have a low measure. In certain cases, length itself (measured as the number of characters) can be used as information. With the minimum information criterion, our estimators will select concise, pithy ways of describing the data.

In applications for which probabilistic models are appropriate, a useful information measure can be found in the standard entropic notion of self-information. The MI estimator then reduces to well known methods of statistical inference.

However, in many "real world" problems where a complex set of possibilities are available, we contend that probability measures do not provide a useful or insightful model. One often has no way to assess probabilities, and difficulty interpreting the very notion, when a problem can not be modelled as one in which a true structure is to be repeatedly selected from an ensemble of structures.

In the many applications where probabilistic models are inappropriate, a variety of non-entropic information measures are available. Perhaps the simplest such notion is to measure the information in a sentence as its length, i.e. by counting the number of characters it contains. With this measure, a minimum information estimator seeks the shortest sentence out of a class. We design our description language in a way which allows our *a priori* expectations to enter the estimator nonprobabilistically. The formal language should have the property that common sub-structures, or expected sub-structures, have relatively short descriptions. Then, given two possible analyses of an input, the one which incorporates the expected sub-structures will be preferred, everything else being equal.

Of course, everything else might not be equal; sometimes unexpected structures are the desired estimate. To see how this can occur, we must be more precise about what the sentences of the language are to describe. Those sentences of the language over which we minimize the information measure do not only describe structures; they must be joint descriptions of possible structures and input data. From any sentence of the description language we must be able to exactly determine the input to the estimator. One way this might be arranged is with a clause which directly describes the input. That technique would not lead to a useful estimator however, as there would be no interaction between the description of the structure and the description of the data. Instead, the language must provide

constructions which take advantage of the fact that a good model allows data to be described concisely. For example, sentences of the language can be built of two clauses, the first describing a structure, and the second containing only the information necessary to describe the input data given that structure. In certain applications, this second clause can be thought of as a description of the noise in the input. More generally, it provides the information to eliminate the variability allowed by the structure.

The effect of this formulation is to allow the desired tradeoff between simple models, and good fit to the data. If an overly simple structure is used in the description of the input data, the remaining discrepancies between the model and the data will be substantial, and will require a lengthy description, i.e. a long noise clause which adds to the length of the sentence. Conversely, it may be possible to eliminate the noise clause altogether by means of a very complex model which exactly predicts the input data, but requires lengthy description. In either of these extremes, the burden of describing the input data is placed in just one of the two possible loci. However, for many inputs, the result will be that the shortest description distributes the information between the description of the structure and the description of the noise. The MI estimator chooses a point of balance between simplicity and fit which depends both on the data and on the *a priori* information that entered into the design of the description language and information measure.

The major questions concerning this framework, which will be addressed in later chapters, are:

1. For what applications is the method suitable?
2. How to design a description language for a particular application.
3. How to design an information measure on the language.

4. How to find a sentence which minimizes the information measure for a given input.
5. What are the provable properties of the estimator?
6. How does it relate to existing estimation techniques.
7. Why does the method work?

In considering these questions, it is important to remember the constraints placed on the method because it is to work in arbitrary spaces (that can be described with PSGs). In particular, we can not assume any notion of addition or distance in the space. Accordingly, the operations of expectation and differentiation, and the notions of estimation error, bias, and variance are not available, even in those cases where a probability distribution is given. In addition, the use of a PSG restricts us to a countable space.

For these reasons, structure estimation problems can be seen as *detection* problems from the point of view of classical statistical inference, which require the selection of a hypothesis from a finite or countable unstructured set. But we will apply the method to problems of far greater complexity than those usually classed as detection. The difference between these problems and classical detection problems is that in the classical case, the different hypotheses are considered to be elements of a set of mutually exclusive hypotheses with no structure, e.g., *target present* or *target absent* in a radar detection problem. In the structural problems that we consider however, the elements of the set are related by means of common substructures. Different hypotheses with similar internal structures are treated similarly in their formal descriptions and information measures. For example, if the hypotheses are trees, and two structures under consideration share identical subtrees, they also share certain clauses in their descriptions and certain terms in the definition of their complexity. The method attempts to exploit these relations in the estimation process.

1.3 Origins

The MI estimation framework presented here is a synthesis of ideas from a wide range of fields. It is an interdisciplinary approach, combining ideas from probability theory, classical estimation and detection theory, system identification, artificial intelligence, syntactic pattern recognition, formal language theory, classical information theory, and especially, algorithmic information theory.

Three methods of taxonomic classification proposed by Wallace and Boulton [W&B 1968; B&W 1973, 1975] contain many of the ideas generalized in this thesis. Theirs is the earliest application of information criteria to structure estimation problems that we have found. They give the first statement of a *minimum information criterion* [1968, p. 185]: "We suggest that the best classification is that which results in the briefest recording of all the attribute information." They consider the problem of grouping individuals, in the form of observation vectors, into classes, and address the structural problem of selecting an appropriate number of classes into which the data in a scatter plot should be classified. Wallace and Boulton's methods are similar to those in Chapter 5, and contain many of the essential elements of this work. However, there are three important differences: (1) a flexible formal language is not utilized, (2) they rely on specific families of probability distributions, and (3) the generalizability of the method was not recognized. Curiously, this work seems to have been largely ignored except by entomologists, botanists, and by Rissanen [1978, 1983]. This may be attributed to the specificity of their application.

The second use of an information criterion for estimating structure which we have found is in Cook *et al.* [1976]. They propose a complexity measure, and

probabilistic measure of fit for estimating stochastic context-free grammars. As discussed in Chapter 4, this work also fits into our general framework.

Another similar approach is given in Georgeff and Wallace [1984]. However, they assume a probability measure is available over the space of structures, and propose that descriptions have lengths proportional to entropic measures of self-information. The resulting estimator is then the classical MAP estimator, and the language is actually irrelevant.

Classical methods of detection for choosing among hypotheses in a probabilistic context began with Thomas Bayes [1763]. These fundamental ideas have been developed into a broad and deep field of probability theory, stochastic processes, and statistical inference which is both mathematically elegant and extremely powerful in real applications. An excellent compendium is Van Trees [1968]. The most important limitation of these techniques however, is that estimation must take place either (1) in an algebraic context, a vector space, in which the number of degrees of freedom is prespecified, or (2) in the context of detection within an unstructured set.

For problems of system identification, in which not only the parameters, but also the order of a dynamic system must be determined, additional techniques have been developed. In essence, they allow estimation in a family of vector spaces with different dimensions, with the proper dimensionality being one of the parameters to estimate. Ljung [1986] is an excellent text in this field. Two notable information criteria have been developed for system identification purposes.

The *Akaike Information Criterion* (AIC) [Akaike, 1974, 1981] provides an estimate of the expected reduction in residual errors when model order is unnecessarily increased by one dimension. When considering two models of different

order, the AIC chooses the smaller unless the reduction in residual error with the higher order model is greater than the expected value. This provides the essential tradeoff between *simplicity*, in the form of low model order, and *fit* to the data, in the form of low mean square residual error.

The second information criterion developed for system identification is Rissanen's *Minimum Description Length* (MDL) criterion [1978-1983], which has been a major influence on this work. Rissanen presents binary-string codings of autoregressive moving-average (ARMA) linear dynamic systems and data, and by minimizing code-length, derives a measure similar to the AIC. He suggests [1978, p. 465] that "by finding the model which minimizes the description length, one obtains estimates of both the *integer-valued* structure parameters, and the real-valued system parameters" [emphasis mine]. The differences between Rissanen's MDL approach and ours are given in Section 2.11.

Rissanen has more recently applied MDL estimation to the problem of data compression via estimation of the structure of a Markov source [1983–1986]. His emphasis is quite different from that in Chapter 4 below in that he again selects a particular *integer* coding with binary strings, and relies upon an integer enumeration of the possible structures rather than a versatile description mechanism. He is not concerned with reasonable estimates of structure so much as the asymptotic properties of the compressed data. Accordingly, the structure estimators fare poorly when given short input strings. This work is discussed in more detail in Section 4.1.

Artificial Intelligence (AI) emphasizes flexible descriptive formalisms, a concept essential to this framework. It is attributed to John McCarthy that "you can not expect a computer to learn a concept if you can not tell it about the concept

directly" [Winston and Brown, 1979, p. 342]. We read "estimate a structure" for "learn a concept", as concepts in AI are represented with formal structures, and learning is seen here as one case of estimation. A wide variety of representation techniques have been proposed for AI applications, including lists of relations, propositional logic statements, semantic networks, and frame systems. Note that these are all syntactic methods that require some vocabulary of symbols and conventions for arrangement which can be formalized with a PSG.

From Syntactic Pattern Recognition (SPR) we take the notion that classes of patterns can often be described with sentences generated by a PSG. Many of the particular problem domains and grammars that have been examined in the SPR framework (e.g. Fu [1975], Grenander [1976]) might well be re-examined and extended with an MI point of view. The major difference is that SPR methods only classify an input as *in a set* or *not* according to whether or not some description of the input can be parsed with a pattern grammar. In forming a description of the input, the SPR paradigm tries to eliminate "noise" or variability, in order to simplify the grammar and parser. The maximum degree of variation which is allowed within a class is then specified not only by the grammar, but also by the nature of the "noise filtering" in the encoding process. In contrast, the MI approach requires that the "noise" be explicitly described in order to measure the fit between a structure and the input. An MI estimator can thereby select among a set of descriptions the best classification of the input, rather than merely list the set of descriptions which are compatible with the input.

Formal language theory (Hopcroft and Ullman [1969, 1979], Kohavi [1978]) suggests a variety of formal description models which could serve our purposes. Of these, the context-free phrase-structured grammar appears to strike a proper

balance between flexibility and manageability. They are used exclusively in this work. Of course, in other applications it is possible that other description models will be more appropriate.

From classical information theory (Hartley [1928], Shannon [1948]) come two concepts of coding and information. A *combinatorial* notion of information arises by counting the number of bits required to code one of a finite number of possibilities with a fixed-length binary string. If there are n possibilities, $\log_2(n)$ bits of information are required, independent of the outcome. In a more restricted context where variable-length codes are permitted and a probability distribution on the possibilities is available, an *entropic* notion of information can be employed in which the information in each possible outcome is a function of its probability. These two notions of information can be combined and extended to apply to formal descriptions.

Algorithmic information theory (AIT), proposed by Solomonoff [1964], Kolmogorov [1965], and Chaitin [1966], provides a third notion of information, defined by the lengths of Turing Machine programs which generate a string. The algorithmic notion of information, should it be known for a set of possibilities, is easily adapted into this framework. It is not likely to be useful however, as it is not computable. A more important contribution of AIT is the philosophical notion that complexity of an individual object can be meaningfully measured outside the restricting context of a probability distribution. Our complexity measures on structures can be viewed as special cases of the algorithmic notion of the information in a structure. The set of Turing Machine programs has been replaced with a set of simpler, more relevant, and more tractable representation functions, which incorporate our structural unknown as a parameter.

Finally, from the field of combinatorial optimization (e.g., Papadimitriou [1982]) we take two important ideas. The first is the method of optimization to a local maximum using local search techniques and a neighborhood structure. Although our overall framework is amenable to a wide variety of optimization techniques, the case studies in later chapters are all handled reasonably with simple one-pass steepest-descent ("greedy") optimization techniques, using a syntactically defined neighborhood structure.

The second idea from combinatorial optimization is that of measuring the size of input data using an information measure. For the purpose of quantifying computational complexity, it has proven fruitful to analyze the space and time complexity of algorithms in terms of their order of growth relative to the size of the input. For example, a certain sorting algorithm might require a number of operations, and hence time, proportional to $n \log(n)$, where n is the length of the input list; an algorithm to find the minimum distance between nodes in a graph might take time proportional to n^2, where n is the number of nodes in the graph. Measuring the length of a list to sort is relatively straightforward and unambiguous, but to measure the size of a graph requires an implicit description language and information measure. In practice, graph size is variously measured as either the number of nodes or arcs. The descriptions and measures to be developed in Chapter 4 for directed labeled graphs can be viewed as more explicit and precise versions of the size measures which originated in the analysis and classification of graph algorithms.

Relative to the above fields, the major contribution of this thesis is in the synthesis of these ideas into a structure estimation framework. We focus attention

on the benefits of estimation by means of a formal description language in association with an information measure. In particular, the enormous flexibility and versatility of the resulting technique allows a single estimation criterion to apply to, and unify, a wide range of fields.

1.4 The Finite-State Machine Estimation Problem

Before proceeding further, we briefly sketch a concrete motivating example, involving complex and flexible structures, in order to illustrate some of the major issues. This problem, taken from Chapter 4, is to estimate the structure of a finite-state machine (FSM), or equivalently, to infer a regular grammar [Hopcroft & Ullman 1979]. This class of discrete finite-memory structures is chosen because the estimation required is essentially structural, rather than numeric, and it is important and familiar in many fields. Furthermore, it illustrates the thesis that when facing a complex structure-estimation problem, one often has no probabilistic models or assumptions to rely upon.

As input to the estimator, we have available a finite sequence of observations which result from an unknown function of the state transitions. As an example, consider the FSMs of Figure 1.1, any of which might generate the particular observation sequence

$$Z = \texttt{ABCABDABCABDA}$$

In these graphs, the *nodes* indicate the allowed states of the FSM, the *directed arcs* indicate the allowed state transitions, and the *labels* on the arcs indicate an observable which occurs during the associated state change. We assume the FSM operates for a time and the observables are gathered in sequence without error.

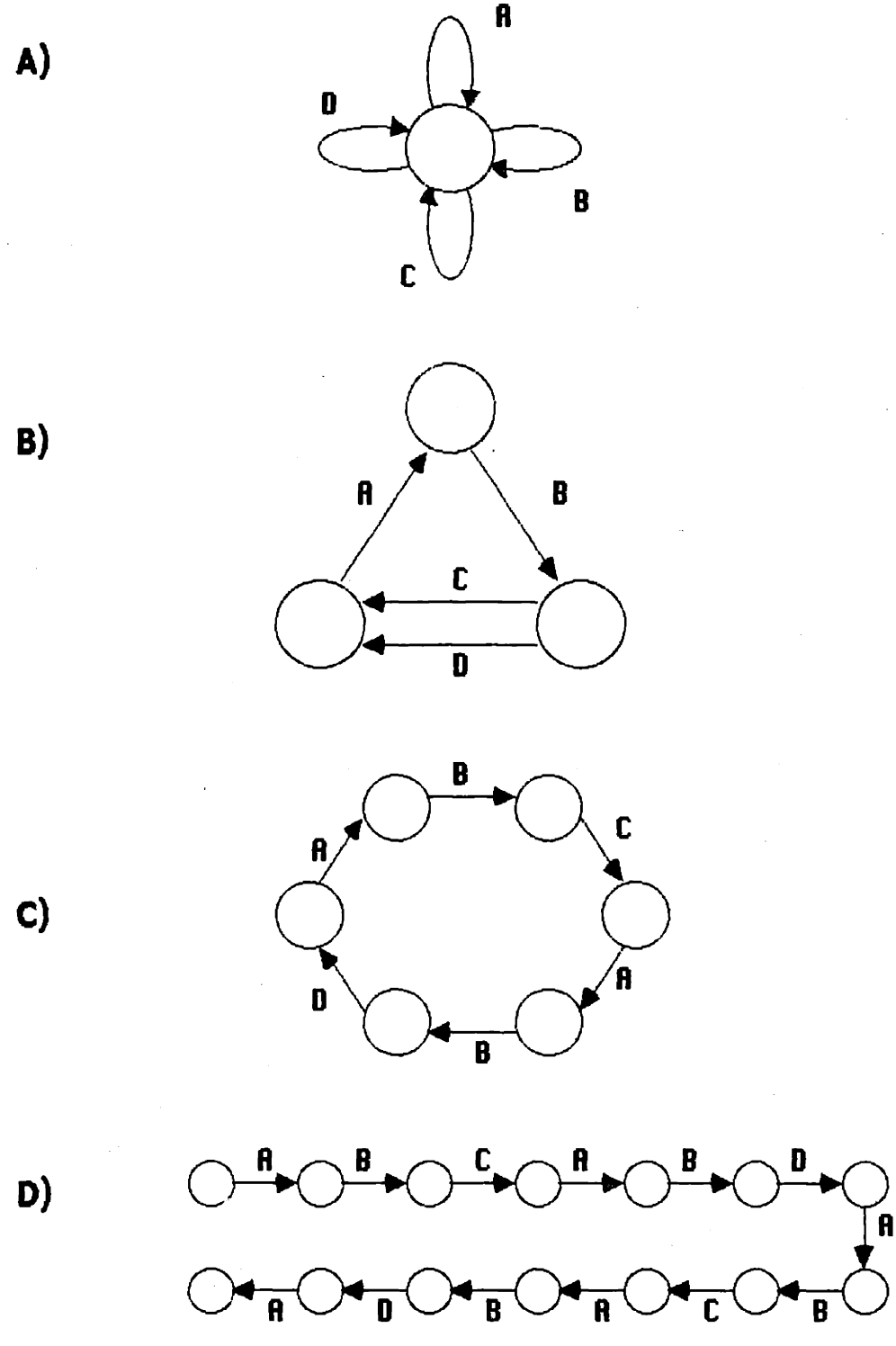


Figure 1.1 Finite State Machines Which can Generate the String ABCABDAB-CABDA

Our problem is that we are given the observation sequence, Z, and we wish to form an estimate of the FSM which generates it. But this problem is hopelessly underdetermined. An infinite number of possible structures, of many different complexities, are candidates in addition to those shown. The MI approach provides a method for regularizing this problem, suitable even if there is no relevant notion of an *a priori* probability measure over the FSMs.

As example applications, consider that the observation sequence could be events which we record while watching the behavior of a clockwork toy, body movements of a dancer, motors and valves in a washing machine, or animated characters in a computer video game. We wish to form a FSM model which in some sense captures the dependencies between the behavioral events we recorded. In these examples, we are unlikely to have a probabilistic notion, for example of folk-dance movements. In practice, our criteria will involve simple models and models which fit the data well, so we need to formalize these ideas.

The essential difficulty in designing an estimator for this problem is thus to choose from the infinite number of possibilities a reasonable balance between simple FSMs and FSMs which fit well to the data. FSM simplicity may be measured in many ways, but if we quantify it as number of states and/or arcs, then the candidates of Figure 1.1 are ordered roughly by decreasing simplicity, i.e., complexity. The notion of an FSM which fits the data can be understood as one which not only can generate the input, but which also tightly constrains the set of possible observations. Comparing Figure 1.1a with Figure 1.1d, we see how these are opposing tendencies in the space of FSMs. The simplest FSM which generates the data, Z, also generates all conceivable observation sequences over the same alphabet. It does not fit Z especially well, because it could be chosen for any input sequence.

Figure 1.1d on the other hand fits the data extremely well, but it is quite complex. Intuitively, we seek a balance between these extremes such as Figure 1.1b or 1.1c.

The description-based MI estimator of Chapter 4 can make this tradeoff. We first need a formal language which specifies a set of sentences for describing FSMs and inputs. There are many ways this could be implemented. The approach of Chapter 4 is to have primitive tokens for describing states and labels, then to build clauses describing the arcs from these elements, and finally form a complete description of the FSM out of the clauses for the different arcs. Given the FSM structure, the observation sequence is described with a clause which indicates the starting state and the route through the states. Notice how in structures with no branching, such as Figures 1.1c and 1.1d, nothing needs to be said about the route. The route only needs description at states for which there is more than one exiting transition. The complete sentence, jointly describing the FSM and the data, is the concatenation of these two clauses.

For the purposes of this introduction, the information content of a sentence can be measured by its length. With these descriptions, and this measure, simple FSMs result in the first clause being short, while good fit to the data results in the second clause being short. With the measure discussed in Chapter 4, the shortest description of Z overall uses the FSM of Figure 1.1b. In this sentence, neither of the two clauses is as short as possible, but their combined length is. A tradeoff is made between fit and simplicity. If the input data were extended, with `C` and `D` continuing their alternating pattern, then after a few repetitions, Figure 1.1c would be chosen as the estimate. It provides a better fit, but is also more complex, and a longer input is required before it can be justified. The mechanism by which this occurs is discussed in Chapter 4.

1.5 Overview

The core of the framework is presented in Chapter 3. This is preceded by background information in the first two chapters, and is followed by four chapters of case studies and a final chapter of discussion. In Chapter 2, we discuss a number of different approaches to structure determination, and various criteria of complexity, which have appeared in the estimation literature. The FSM estimation problem is used as a concrete example in pointing out the limitations of these existing methods.

Chapter 3 presents formal-language-based minimum-information structure estimators. Necessary properties for the formal languages and information measures are developed. Special properties of the languages and information measures, for which the method reduces to the criteria of Chapter 2, are pointed out. Techniques for optimizing the resulting criterion over a space of sentences are also discussed.

Chapter 4 considers several examples. First, the nonprobabilistic FSM example is presented more completely. Then it is extended to a probabilistic context in which a Markov source is estimated from a sample of its output. Finally, an estimator for a set of independent FSMs is developed, and the results of a simple computer implementation are shown.

In Chapters 5 through 7, three other case studies are presented. The first example comes from the field of pattern recognition. It is a cluster analysis problem similar to the original Wallace and Boulton [1968] study, except that no probabilistic assumptions are made. The structural issue is the number of clusters to use in the analysis. The next case study is a simple signal processing problem in which a one-dimensional function of time is segmented into piecewise constant segments. The structural question here is also one of selecting an appropriate

number of components. The final example is a two-dimensional generalization of the signal processing example, coming from the field of machine vision. The problem is to estimate the piecewise-constant structure of a visual image in the presence of noise. In the latter two examples, probabilistic noise models are assumed and incorporated into the estimator, but no probabilistic assumptions are made concerning the distribution of the structures.

Chapter 8 contains a summary of the framework, a discussion of the method from several points of view, and a section suggesting possible extensions and applications. One concern about our method is that is not always clear what the resulting estimator corresponds to. The general technique allows one to quickly and easily generate optimization criteria for a wide range of problems, but it is not always clear how to understand the criteria, or relate them to others. Some possibilities are discussed in Section 8.2, in which we show the method can not be validated on purely logical grounds, yet it can be supported by intuitive and psychological arguments.

In the Appendix, a residential energy monitoring application is described which served as the motivation for this work. It directly suggests three of the case studies.

The major contributions in this work are in the problem formulation and suggested framework for estimation. In addition, the various case studies present simple, computationally tractable solutions to difficult structural estimation problems which can be directly employed or adapted in a variety of applications.

The focus throughout is on breadth and flexibility. We try to illustrate the fundamental issues and relations in structure estimation as we see them, but this is very much a report on work in progress. There are many thorny and vexing

issues in any methodology which induces generalities from specific input data. We explore problems and solutions of this character from a range of points of view, presenting as much perspective as we can. However, we have no illusions that this work is, or may ever be, complete. In particular, there is no way to prove the validity of the framework presented, although the case studies support a strong inductive (nondemonstrative) argument for the MI approach.

Chapter 2

APPROACHES TO STRUCTURE ESTIMATION

Before formally presenting the description-based MI approach to estimation in Chapter 3, we briefly examine a number of other approaches which have been applied to problems of estimating structure. A survey of the literature in a range of fields suggests the following methods are widely used, although they are not all recognized or named as such. For reference, they are gathered together in one list:

1. Maximum Likelihood Estimation
2. Ockham's Razor
3. Falsifiability
4. Maximum A Posteriori Estimation
5. Estimating Substructures of a Largest Allowable Structure
6. Simplest with Acceptable Fit
7. Finding the Knee of the Curve
8. Hypothesis Rejection
9. Maximum Entropy Estimation
10. Ad Hoc Methods
11. Minimum Description Length Estimation

Many of these techniques pervade the literature in a variety of applications, and cannot be specifically referenced. Some incorporate classical techniques as "sub-routines", others rely on simplicity or good fit alone. These methods will be seen to

have inherent limitations which are serious for many applications, but acceptable in others. Their place within the general framework presented here is discussed in Section 3.4, where they are all shown to be special cases of MI estimators in which special description languages and information measures are assumed.

2.1 Maximum Likelihood Estimation

Although quite powerful in many situations in which the structure is fixed, Fisher's Maximum Likelihood (ML) estimation technique is generally unsuited for structure estimation problems. In most applications with unknown structures, a structure with a large number of parameters can be selected and adjusted to fit the data more closely than a structure with few parameters. As the ML criterion selects the model which maximizes the conditional probability of the observation given the structure, the complexity of the model is not penalized. Therefore, complex estimates frequently result, in which the large number of degrees of freedom effectively model the process noise or observation noise rather than the structure being sought. Comparatively, simple structures are penalized, and an estimate of the greatest allowable complexity generally results.

The resulting estimate is generally inappropriate because it emphasizes fit to the data, and ignores simplicity. Generally, ML estimators are suitable for structure estimation only if either (1) the application can tolerate very complex structures, (2) the possible structures are considered to be of equal complexity so that it is reasonable to measure structural complexity as constant, or (3) the structures are not "nested" in a way which allows more complex structures to provide better fits, e.g. if only one structure can be used in the description of the data, the most complex structure is a reasonable estimate.

In the FSM case, an unconstrained ML estimate for the input above will result in a structure such as Figure 1.1d, which exactly describes the observation, but makes no prediction. However, in many applications, models are being estimated specifically to make use of their predictive power. For such purposes, the ML estimate is the worst of the estimates in Figure 1.1.

A more precise analysis, of course, requires a careful formal model of Markov sources, as in Chapter 4. Here, it is sufficient to note that if the interpretation of Figure 1.1d includes the fact that the leftmost state is the starting state, and probabilities are somehow assigned to the different transitions out of a state, then the input, Z, results with probability 1, and Figure 1.1d is at least *a* ML estimate, if not *the* ML estimate.

2.2 Ockham's Razor

Ockham's Razor is a principle for choosing among theories that is commonly advocated by philosophically inclined analysts. This "principle of parsimony" states that one should select the simplest structure which is compatible with the data. Although compelling, and sometimes useful, especially in the philosophy of science, the principle is difficult to apply rigorously as there is no universally accepted notion of simplicity.

In our framework, the information measure on descriptions provides a precise definition of structure complexity, and it is interesting to see what Ockham's razor leads to in such a context. Although a complete presentation will not be given until Chapter 4, Figure 1.1a will be the structure of minimum information that can generate Z, by any additive measure of information, built on clauses describing states, arcs, and/or labels. It has the minimum possible number of each type

of component. Accordingly, an estimator which chooses the simplest possible structure compatible with Z will choose the FSM in Figure 1.1a. Unfortunately, this estimate, like the ML estimate in Figure 1.1d, has no predictive power. All the symbols in the set {ABCD} are possible successors to any input string. The criterion of simplicity is therefore of little use.

(Note, incidentally, that this is also true if applied literally in the philosophy of science. I claim that the simplest physical theory compatible with our observations of the world is "Anything can happen". But such a theory is quite poor relative to most of the other physical theories. Ockham's razor is a poor guide not because we don't agree on what is simple, but because we can agree on a simple, but useless, theory.)

From the point of view of MI estimation, Ockham's razor, in its most general interpretation, focuses on simplicity to the detriment of fit. As the degree of compatibility between the data and the structure does not affect the estimate, the estimate is not sufficiently tailored to the input. In a setting of constrained optimization however, Ockham's razor is more useful. In the FSM example, if Figure 1.1a and others with poor fit are eliminated from the structure space by some application-specific constraint, then Figure 1.1b might be left as the simplest structure.

2.3 Falsifiability

Again following the lead of philosophers of science, we might seek an estimation technique which results in FSMs that give strong predictions. Popper [1962] claims that the "best" theories are those which are most *falsifiable*, arguing that science progresses as we find counterexamples to old theories, and we are forced to

amend them. This view is based on the asymmetry between possible and impossible events. If we err on the side of expecting very few things are possible, then new observations can cause us to modify our theories, and hopefully progress. But, if we err on the side of expecting that too many things are possible, then we can't correct the theory, as the world only presents positive evidence. By this argument, a theory should allow for the smallest set of possibilities compatible with what has already been observed. This can be viewed as resulting in an expanding sequence of possible outcomes generated by the sequence of theories.

In the context of our problem, the natural interpretation of this approach is to select FSMs which generate *small languages*, in the sense of set inclusion. Thus, of two contending FSMs which generate Z, for which the set of strings generated by one is a subset of the strings generated by the other, the one with the smaller language is preferred by the criterion of falsifiability. There is a larger set of logically possible observations which could serve as counterexamples. In the FSM problem, this is a natural principle because our model only allows "positive evidence". A finite input sequence can never imply that some other sequence of observations can not be generated by the "true" FSM.

The falsifiability criterion is closely related to ML estimation in that it ignores the complexity of the structures and so is likely to choose a very complex model. It can be seen as a special case of a ML estimator in which all observations are considered equally likely, so the highest likelihood results from a model with the fewest possible outcomes. For structure estimation, the principle of falsifiability has little value in a setting of unconstrained optimization. For the FSM estimation problem above, the most falsifiable structure is Figure 1.1d. This FSM allows the

input to be generated, but nothing else. It makes too strong a prediction; it is too falsifiable.

(The analogous problem in the context of physical theories is the theory "The world can only be just what it has been, and it will end right now" which is always falsified immediately. Somehow this must be eliminated from the set of putative theories.)

If acceptable constraints can be placed on the set of allowable models, so that such trivially falsifiable models are eliminated, the falsifiability criterion can be quite useful, at least in nonprobabilistic settings with only positive data. Of course, falsifiability is not applicable in a general estimation context unless the input can be viewed as a sequence over a one-dimensional parameter such as time, and the notion of a predictor is relevant. Many estimation problems can be stated in this format, however.

2.4 Maximum A Posteriori Estimation

In principle, a balance between simplicity of models and fit to the data can always be effected by Bayesian estimation. Of the Bayesian estimators, only the maximum a posteriori (MAP) estimator is appropriate for estimating structure. The MAP criterion selects the structure for which the conditional probability, given the data, is maximum.

(Minimum mean-square error estimation, and other Bayesian estimators derived from cost functions on the estimation errors, are not meaningful outside the context of a vector space; structures can not be subtracted. It may be reasonable however, to formulate certain structure estimation problems in metric spaces as a

minimization of the expected value of a cost function on the distance between the true structure and the estimate. This is briefly discussed in Chapter 8.)

With MAP estimators, if complex structures are given low *a priori* probabilities, they will not be selected merely because they fit the data slightly better than simple models. The MAP formulation ensures that the improvement in fit be sufficient to outweigh the decrease in *a priori* probability. The difficulty with such an approach, of course, is that there is no way to assign *a priori* probabilities to different structures in real applications. In the vast majority of cases, the very notion of a probability distribution on the possible structures is suspect. In complex estimation problems, probability measures can be very poor models offering little insight or guidance.

In our context of FSM estimation, the problem would be to assign probabilities to the structures in Figure 1 in a way that makes sense. However, in most of the applications for which FSM models are appropriate, it is not clear how to make this into a meaningful question.

2.5 Estimating Substructures of a Largest Allowable Structure

One common technique for estimating structure is to organize a problem so that all the allowable structures are substructures of a given structure of maximum complexity. Parameter estimation is performed by classical techniques, with the number of degrees of freedom in the largest structure, and then the estimated parameters are examined to see which substructure is indicated. For example, in system identification, when identifying a system which is expected to be second, third, or fourth order, one can estimate the parameters for a fourth order system, and if the fourth order, or third and fourth order, coefficients are "sufficiently

close" to zero, choose a third, or respectively second, order system. An analogous technique is often used to choose the order of a polynomial as a least-squares fit to data. Typically, this technique is used only to select a model order, and then the parameters are reestimated with the selected number of degrees of freedom in order to determine more precise values.

A major difficulty with such techniques is in the characterization of sufficient closeness. The minimum degree of deviation from zero for which the parameter can be declared statistically significant is not generally clear. A second problem is that it is not suitable for estimation problems in which an upper bound for the complexity is not given.

2.6 Simplest with Acceptable Fit

Another technique for selecting a structure, used in system identification and elsewhere, involves preselecting a minimum acceptable fit between the model and the data. In many situations, a criterion of lack of fit, such as mean square residual error, can be stated as a system requirement based on the application. When the structural unknown is essentially one dimensional, i.e. model order, the system can be estimated using classical techniques for each of a sequence of orders. The smallest order for which the fit is within the acceptable bounds is then accepted as the structure. Note how this is the dual of the classical technique of preselecting a model structure, and minimizing the residual error within that class of models. It is also a special case of Ockham's Razor in which the criterion of compatibility with the data is determined by the fitness measure.

One problem with this technique is the restriction to one-dimensional structures. A much more serious problem is that a rigid criterion of fit is rarely given

as a true constraint. If the smallest order with acceptable fit were found to be much greater than a very simple model which just missed being acceptable "by epsilon," one would be inclined to adjust the constraint slightly and accept the simpler model. One would like to adapt the degree of fit in a way which depends on the relation between simplicity and fit for the particular class of models and data at hand. This is the intention of the following method.

2.7 Finding the Knee of the Curve

A closely related technique, also popular for system identification and polynomial order determination, involves finding *the knee of the curve.* Assuming again that the structural unknown is one dimensional, a measure of lack of fit, such as residual error, is plotted versus model order, and generally is found to decrease with increasing model order. Typically, this curve is not of constant slope, but decreases rapidly at first, and then slowly. A point in the curve at which the slope begins to "level off" is termed the *knee* and the corresponding model order is selected as the estimated structure. The slight increase in fit to the data for larger orders is deemed not to be worth the cost of the more complex structure. Relative to the method of the previous section, this has the advantage of not requiring that a degree of fit be prespecified independently of the data.

The technique is commonly applied in system identification and curve fitting, when model order is essentially one dimensional (within the family of models of interest) and mean square residual error is available as a measure of fit. When the structural question is not one dimensional, the method can often be adapted. In these more complex structure spaces, if a complexity measure from structures

to the real numbers is available, it provides a way of "one-dimensionalizing" the space.

The fundamental problem with this method is in choosing an appropriate degree of leveling off. A simple visual knee is often selected, but this is actually quite arbitrary. The same data will appear to the eye to have very different knees if the dimensions of the units on the axes are changed, e.g., from volts to millivolts, or if the scaling is changed, e.g., from linear to logarithmic.

If, however, a criterion is available for choosing the critical slope, the method can be very effective. Akaike's Information Criterion is such a criterion. The description-based MI technique provides other criteria, as will be seen in Chapter 3. Both complexity and fit are measured with an information measure, and the critical slope is unity.

2.8 Hypothesis Rejection

Fisher's method for rejecting hypotheses can be used to choose between two or more structural hypotheses. The method requires that probability distributions be derived for one or more statistics which are modelled as random variables conditioned on the hypotheses. If the actual input data falls *sufficiently far* into the "tails" of the distributions, the hypothesis is rejected.

There are serious problems with this technique, even in those contexts in which probabilistic models are natural. The most obvious difficulty is that there is no principled method of choosing an appropriate degree of distance into the "tails", or confidence level, to define the *critical regions* of the distributions. A second problem is that the method can provide no provision for confirming hypotheses, and there is generally a nonzero probability of erroneously rejecting a

"true" hypothesis. The result of this is that with a sufficiently industrious application of the method, i.e., with a large number of statistics, one is certain to reject any given hypothesis, including true ones.

A more subtle problem with the method lies within the very notion of a "tail" of a distribution. Given a statistic in a totally ordered space (e.g., real numbers, integers) with a unimodal probability distribution, there is an intuitive naturalness to grouping together a "ray" of values which does not include the mode as a "tail", but there is no logical basis for doing so. Consider that any permutation of the space defines a new statistic with an arbitrarily different set of tails. By an appropriate permutation, any arbitrary set of values of a given statistic can be grouped together and made into the critical region of an isomorphic statistic.

2.9 Maximum Entropy Estimation

The Maximum Entropy (ME) technique of Jaynes [1982], employing an analogy with statistical mechanics, selects the probability distribution with maximum entropy that is compatible with partial observations such as various moments. Although we have not seen the ME principle applied to problems of the type we term structure estimation, it has been applied to a range of problems, including image reconstruction. In examples we have considered, ME, like ML, leads to the most complex estimates possible. But, we include it here because there is a potential fit between the method and these problems, especially in the context of constrained optimization. The ME technique can simultaneously estimate an arbitrary, even infinite, number of parameters.

Note, incidentally, that Watanabe [1985] defines a *Minimum Entropy* notion of pattern recognition. This does not refer to an estimation criterion however.

The phrase summarizes the process by which a large amount of information in the data is "boiled down" to one bit of information in the output: the data fits or doesn't fit a pattern class.

2.10 Ad Hoc Structure Estimators

Many practical problems have necessitated the development of some kind of estimator in applications where structures are uncertain. In the absence of an estimation framework, or applicable criteria which can be effectively minimized, various *ad hoc* procedures have been developed and implemented. Often, the combination of necessity, expertise in a complex domain, and engineering judgement results in estimators which serve their purpose very well. The problem of FSM inference has seen many such estimators. Image processing (e.g. Pavlidis [1977]) is another field rife with *ad hoc* structural estimators.

2.11 Minimum Description Length Estimation

The use of information measures in structure estimation, as developed by Wallace and Boulton [1968, 1973, 1975], and Rissanen [1978–1986], allows a quantified tradeoff between model complexity and fit to the data. In the framework we are proposing here, these measures are defined on sentences in arbitrary languages for describing structures and data. However, in the previous work they were derived in the context of binary codes for explicit applications: clustering, ARMA model order estimation, data compression. It was not immediately obvious how to generalize the technique from these special cases.

Rissanen's MDL estimation technique determines model complexity by: (1) truncating the real-valued parameters in a class of models to an adjustable precision, thereby producing a countable set of approximate models; (2) encoding the structures within this set as a positive integers with an enumeration function; (3) assigning a probability distribution over the structures, based on binary codes for integers, using the $\log^*$ function on positive integers, as discussed in Chapter 3; and (4) measuring structure complexity and fit with probabilistic notions of information.

A major portion of his work, which is distracting from our viewpoint, is devoted to the problems of "rounding off" the uncountably infinite number of possible real-valued inputs to a finite precision, so that they can be described with the countably many binary strings. The approach is adapted from that of Wallace and Boulton [1968].

We see this is a special case of a more general description-based estimation technique. Particular choices as to the method for describing structures and measuring their information content have been chosen which are appropriate to the applications. Four crucial differences distinguish our work from the above, which generalize the approaches of Wallace and Boulton, and Rissanen:

(1) Rissanen has considered only *integer-valued* structure parameters, e.g., *model order*, whereas we propose an adaptable formal-language framework. This allows MI criteria to be quickly and easily generated for more complex estimation problems, including those with recursively specifiable structure, and no meaningful enumeration by integers. Our description languages are also made to provide neighborhood structures for local search techniques.

(2) We distinguish clearly between a description language and an information measure. They have different properties and serve different functions, but are blurred together when the language is binary strings and information is measured as length. The virtue of this

distinction becomes very clear when an MI criterion is to be modified. Our breakdown allows us to change only the portions which we want to change, but a binary coding formulation requires a code and its length to change in tandem. Measuring information, rather than constructing codes, also allows noninteger information quantities, which have to be excused somehow when considering code lengths.

(3) We choose more complex problems which are naturally suited to description with a countable language, and thereby avoid the distractions of rounding off real-valued input data. Because a space of structures generally lacks the simple ordering of the integers, considerable attention must be paid to the problem of optimization.

(4) A general MDL approach allows arbitrary binary codings as representation functions for a problem domain. This introduces a generic problem with the technique which comes about because there are no universally appropriate codes or measures of information. The coding function which defines a particular MDL estimator can be compounded with any permutation of the set of binary sequences to form a new coding function and new estimator. Almost all such choices will result in very poor estimator performance. We therefore explore those properties of representation functions which result in acceptable estimates, and recommend a structural isomorphism between the structure space, the language for describing structures, and the definition of information.

This MDL framework requires that structure be coded as binary strings in which information is measured as length. The versatility of the method was not widely noticed, and information measures have not been widely applied for structure estimation. We argue, of course, that they should be.

2.12 Conclusions

In this chapter we have reviewed a variety of techniques which have been used for estimation problems in structure spaces. Each follows from certain choices as to how to process the data. The intent of the MI approach is to make as explicit as possible the choices required for structure estimation. By doing this in as general

a context as possible, we will see that special cases of the MI parameters result in each of the structure estimation methods of this chapter.

Relative to these methods, we feel that a formal-language-based MI technique has certain general advantages. In particular, a wide range of structure estimation problems can be directly approached using the framework of a formal language and an information measure. Simplicity and fit can be given appropriate weights. The main difficulties with this approach will become apparent in Chapter 3. A formal language must be designed, information measures must be specified, and a combinatorial optimization problem must be solved. We do, however, provide guidelines and heuristics for each of these steps.

Chapter 3

FORMAL MODEL

This chapter gives a more formal presentation of the minimum information framework for estimating structures, which was outlined in Chapter 1. Section 3.1 describes the properties we require of phrase structured languages, and their interpretations as structures and data. Ideally, they should be unambiguous, uniquely interpretable, and structurally homomorphic to elements of the structure space. Information measures on PSGs are developed in Section 3.2. Their definitions follow the definitions in the grammar. The MI estimator is then defined to select a sentence with minimum information that describes the input data. Techniques for optimizing the MI criterion, especially those using the neighborhood relations induced by the PSG, are discussed in Section 3.3. Finally, special values of languages and information measures for which the MI estimator reduces to the various methods of Chapter 2 are discussed in Section 3.4.

3.1 Description Languages and Their Interpretations

Our approach to estimation centers on the design of formal languages (e.g., Hopcroft and Ullman [1969, 1979]) for describing structures and observations. To construct an estimator, the system designer must specify a formal grammar which generates a set of sentences, a *language*. The grammar also determines a structural analysis of each sentence into nested components, *clauses*, which in turn are composed of *subclauses*, etc. This set of structured sentences plays four critical roles in the overall process.

(1) Sentences generated by the grammar are *interpreted* as descriptions of structures and of inputs. There must exist functions which uniquely determine the structure and the input described by each sentence of the language. This will be specified hierarchically, by defining a function which interprets clauses in terms of the interpretation of their subclauses and their manner of combination. In this we follow Frege's *principle of compositionality*, put forth as a principle of how natural language is interpreted.

(2) Sentences of the language *contain information* which can be formally measured. The information measure is also specified hierarchically, so that the information in a clause is a function of the information in the subclauses.

(3) Sentences of the language may be similar or dissimilar according to a *neighborhood relation* which is true of pairs of sentences that are sufficiently similar. These neighborhood relations can be defined in terms of the grammatical structure of sentences

(4) Sentences of the language are *strings of symbols* and may be easily manipulated by a computer executing an estimation algorithm.

We can not give a "cookbook" method for constructing a formal language meeting these needs for any particular application. We do not give a specific theory so much as a theory schema. Certain general properties seem natural for all domains, and have been incorporated into our presentation, while the particular details of the description language will vary from application to application. Out of a range

of formal language options, we have selected *phrase structured grammars*, first formalized by Chomsky [1956], as the core of our framework.

Under the general heading of Syntactic Pattern Recognition (e.g., Fu [1982]), this class of languages has been used for a very wide range of structural domains, including bubble-chamber particle trace analysis, chromosome recognition, and fingerprint identification. Because phrase structured grammars are almost ideally suitable for the formalization of many diverse applications, we do not feel overly restrictive in focusing our attention on this one class of grammar. There is enormous flexibility within this class to tailor a grammar to a problem domain.

In principle, any one-to-one mapping between a set of structures and a set of strings provides the basis of a description formalism, but almost all such mappings will have very poor properties for estimation. We must restrict ourselves to mappings which are "natural" for the estimation purpose. This is not a definable concept, as it is relative to our purposes and understanding of the set of structures in question. One general characteristic we consider essential to natural descriptions is a structural homomorphism between the constructs of the language and the elements of the structure.

Our general expectation is that the types of formal representations which a computer scientist would "naturally" propose as data structures for describing models in the class of interest, and input data relative to the models, will usually have the appropriate properties for syntactic estimation. We can proceed then by exposing the principles developed in the training of a computer scientist, which result in "good" data structures. Much has been written in the field of Artificial Intelligence concerning representations. We require, as a minimum,

that the language should be defined with constructs that allow *naming* of relevant substructures, and make *combination* and *alternatives* explicit. Statements are built compositionally, combining clauses which describe the substructures that combine to form larger structures. When alternative substructures are permitted in a structural context, the grammar should allow alternative clauses in a parallel syntactic context. In addition, definition clauses may allow relevant items or substructures of interest to be described once and named, then referred to elsewhere in the description as required. Various types of clauses may concatenate to form larger clauses, while maintaining an homomorphism between the phrase structure of the statement, and the hierarchical structure of the objects being described.

In describing complex structures, there may be several intertwined structures around which to organize a description. For example, to describe possible states of an internal combustion engine, for the purpose of estimating a diagnosis of a fault, it is not immediately clear how to organize the related descriptions of mechanical, electrical and fluid systems. Here, the general context of the estimator will have to be considered to determine the appropriate organization.

Phrase structured grammars can meet the criteria above. They have been formalized in various ways, and we briefly outline one method of definition. Rigorous definitions can be found in the references cited above. To illustrate the terms, we give an example which is used later in Chapter 4. Formally, a sentence of a language is a *string*, or finite sequence, of *terminal symbols*, e.g. characters. The accepted sequences of symbols are specified through an intermediary set of *nonterminal symbols*, which correspond to the types of *phrases* or *clauses* in the grammar. The grammar specifies a *production rule* for each type of nonterminal symbol defining how it is constructed in terms of terminal and/or nonterminal

symbols. The grammar itself can be formalized as a four-tuple consisting of the nonterminal symbols, the terminal symbols, the production rules, and a selected nonterminal, designated as the *root* symbol. The root symbol, usually "S", defines the clause type of a sentence. The terminals and nonterminals must be disjoint.

There are many options for specifying the production rules. We restrict ourselves here to two kinds of primitive rules, and later define two other types of rules as abbreviations for sets of primitive rules. The first rule permits *concatenation* of clauses. We write

$$A \rightarrow B\ C\ D$$

to indicate that a clause of type A is formed by conjoining clauses of types B, C, and D in that order. A must be a nonterminal symbol, while the right hand side of the rule may contain any number and combination of terminal or nonterminal symbols. The second form of primitive rule indicates *alternative forms* of a clause. We write

$$A \rightarrow B|C|D$$

to indicate that a clause of type A may consist of any one of the clause types on the right hand side. Again, A must be nonterminal, and the right side of the rule may contain any number of terminals and/or nonterminals. We require each nonterminal to appear on the left side of exactly one production rule.

Finally, a *derivation* of a nonterminal symbol, A, may be defined as a sequence of string *rewritings* starting with A. In each rewriting, a nonterminal symbol is replaced according to the production rule in which it appears on the left. If the rule is a concatenation rule, A is replaced with the complete sequence of symbols on the right side of the rule. If the rule is an alternation rule, A is replaced with

any one of the symbols on the right. A derivation is *complete* if it terminates in a string of terminal symbols. We are only concerned with complete derivations below. The final rewriting of a complete derivation of the root node is a *sentence*, and the language generated by a grammar is the set of all such sentences.

We now extend the set of allowable productions with two useful constructions which combine concatenation and alternation. By defining these constructions as abbreviations, we can make use of their convenient forms without having to explicitly complicate the definitions of information in Section 3.2. The first abbreviation is to express *alternative concatenations* such as

$$A \rightarrow BCD \mid E \mid FGHI$$

which is understood as an abbreviation for a set of productions with certain nonterminal symbols implied:

$$\begin{aligned} A &\rightarrow A_1 | A_2 | A_3 \\ A_1 &\rightarrow B\ C\ D \\ A_2 &\rightarrow E \\ A_3 &\rightarrow F\ G\ H\ I \end{aligned}$$

Secondly, we allow a *superstar notation* to indicate an especially common alternative concatenation in which a clause type may be repeated any number of times, including zero. The production

$$A \rightarrow B^*$$

can be interpreted as an abbreviation for the infinite sequence of rules abbreviated by

$$A \rightarrow \lambda \mid B \mid BB \mid BBB \mid BBBB \mid \ldots$$

where λ indicates *the null sequence* with length zero.

As an example, we define a grammar for describing directed labeled graphs, such as those used in Figure 1.1, to represent finite state machines. We assume the notions *node, arc, label, source* and *sink* for such graphs are already defined. This particular grammar will be restricted to the set of directed, labeled graphs, G, with a maximum of N nodes, and labels in the four-character alphabet $\{A, B, C, D\}$. The root node of this grammar will be the nonterminal symbol `Graph`. The first production specifies that `Graph` may be rewritten as a sequence of clauses of type `Arc`, which describe the arcs of a graph. Our grammar will describe an arc with a clause describing its source, sink, and label. The source and sink are nodes of the graph, which we describe with numbers.

$$
\begin{aligned}
\texttt{Graph} &\rightarrow \texttt{Arc}^* \\
\texttt{Arc} &\rightarrow \texttt{Source Sink Label} \\
\texttt{Source} &\rightarrow \texttt{Node} \\
\texttt{Sink} &\rightarrow \texttt{Node} \\
\texttt{Node} &\rightarrow 1|2|3|\ldots|N \\
\texttt{Label} &\rightarrow A|B|C|D
\end{aligned}
$$

For any derivation, there is a corresponding *derivation tree* in which the nodes are labeled with symbols to indicate the rewritings which occur in the generation of a sentence. Each node and its daughters in the tree correspond to an application of a rewriting rule in the obvious way. The root of the tree is labeled with the root nonterminal symbol, and the leaves of the tree are labeled with terminal symbols. For example, the grammar above generates the sentence

12A21B22C

with the derivation tree shown in Figure 3.1. The figure also indicates pictorially a directed labeled graph which is our interpretation of this sentence. Note that

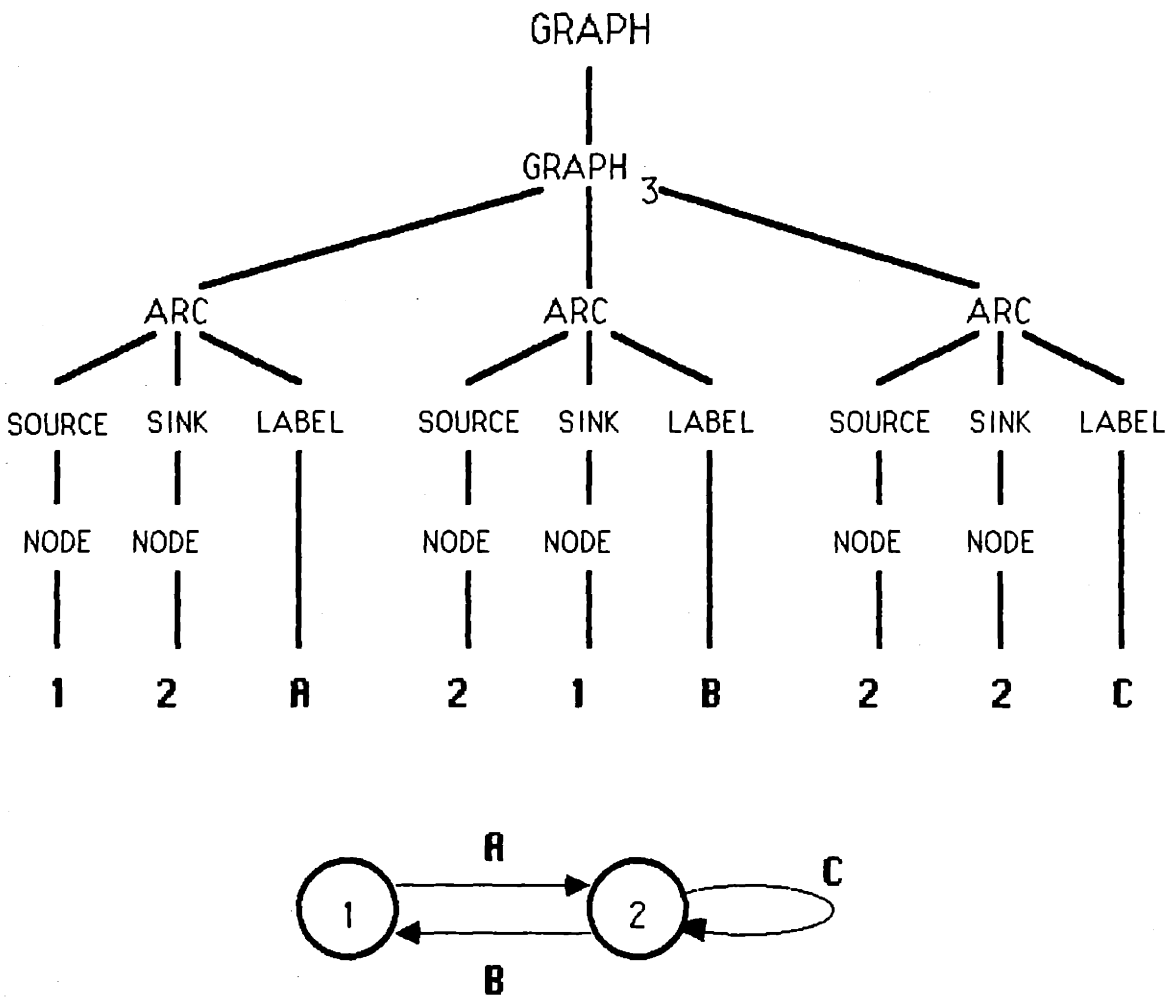


Figure 3.1 Derivation Tree, and Interpretation as a Directed Labeled Graph, of the Sentence *12A21B22C*.

---

in the derivation tree we have included the nonterminal $\texttt{Graph}_3$ implicit in the superstar notation. (This grammar is too specific in that it restricts the number of nodes to N. This will be modified in Chapter 4.)

The nonterminal symbols of the above grammar were selected tendentiously, to convey the interpretation of each clause as a description of a component of a directed labeled graph. This can be formalized for this example in the obvious way. A sentence, S, is an interpretation of a directed labeled graph, $G \in \mathbf{G}$, iff there exists a one-to-one function, f, from the nodes of the graph into the set

of Node symbols, $\{1,2,\ldots,N\}$, and a one-to-one relation between the arcs of G and the Arc clauses, such that the terminal symbols $[N_1\ N_2\ L]$ constitute an Arc clause corresponding to an arc in G labeled L from a node n_1 to a node n_2 with $N_1 = f(n_1)$ and $N_2 = f(n_2)$.

This grammar contains no examples of recursion—no node can expand in a way which includes another node of the same type—so the maximum depth of the tree is therefore bounded. For other classes of structures, although none of the case studies in this thesis, recursive grammars will be appropriate. Examples would include estimation of mathematical formulas, algorithms, fractal structures, and architectural structures.

This interpretation relation is many-to-one from the language to the set of directed labeled graphs. Every sentence is a description of only one graph, but there are many sentences to describe each graph. For a graph with V vertices and E directed edges, there are generally $\binom{N}{V}V!E!$ different descriptions, as there are $\binom{N}{V}$ sets of numbers which can be permuted $V!$ ways in identifying the nodes of G with terminals in $\{1,2,\ldots,N\}$, and for each numbering, the Arc clauses can be arranged in $E!$ permutations. (These Arc permutations will not all be distinct however in cases where the graph contains more than one arc with the same source, sink and label.) For example, the graph of Figure 3.1 can also be described in this grammar with the sentence *21B12A22C* in which the Arc clauses have been permuted, or with the sentence *48A84B88C* in which a different node numbering function is employed.

This many-to-one relation is typical for our applications. There are generally many ways to describe a given structure unless we specifically constrain the

grammar not to allow multiple descriptions. This is a standard problem when describing sets with sequences, because the elements of a set are not ordered, while the elements of a sequence are. For reasons concerning the information measure in the following section, we will often want to eliminate multiple descriptions and secure a one-to-one relation between sentences and interpretations. This will require the introduction of some dependencies between various clauses of a sentence, and reduce the independence assumed above for the rewriting of nonterminal nodes. In principle, there are three ways this can be achieved: in the syntax, the interpretation, or the information measure.

Syntactically, we can eliminate multiple descriptions with constraints which choose a preferred description over others. For example, we can reduce the $E!$ permutations of the Arc clauses to a single acceptable order if we constrain the sentences of the grammar to those in which these clauses are in a specified lexicographic order. Syntactic constraints with this effect can be added as distinct component of the grammar, they can be buried in a complex set of production rules of the above type which obscure their effect, or they can be implemented with context sensitive production rules which specify allowable contexts for each production.

Most multiple descriptions could also be eliminated by modifying the interpretation to make the alternative sentences be descriptions of other objects. For example, we can define all sentences which do not meet the lexicographic constraint on Arc ordering to be interpretations of the null graph. This results in every graph having a unique description except the null graph, which has infinitely many descriptions. We will avoid this option, as it requires an unintuitive interpretation rule for arc combination. The third method of dealing with multiple descriptions

does not eliminate them from the grammar, but adjusts the information measure so that sentences are measured as if descriptions were unique. This is generally the clearest solution to the problem, and is discussed in Section 3.2.

A better solution might be to define two types of combination rules, corresponding to the ordered and unordered properties of sequences and sets. Then two different information rules could apply. We do not formalize this here.

A more fundamental problem of multiple description is that there are many different grammars and interpretations which might be proposed for describing structures in any application. We could, for example, describe a graph with a sentence composed of clauses which describe the nodes of the graph, rather than the arcs. Each node term of the description could list the identifying number for the node, and contain a subclause describing the arcs which have the node as source (or sink). These subclauses would describe the label and sink (or source, respectively) of the arc. Another organization for the grammar and interpretation could be centered on labels.

For general applications concerning graphs, the principle of structural homomorphism suggests arcs, nodes, or labels be the principle clause type, depending on the interpretation. In more specific contexts, e.g., cyclic or complete graphs, other forms of description will be more appropriate, and a language would have to be proposed which is organized in a manner relevant to the purpose.

We will require one further property from our formal languages. For sentences to describe structures, it must be possible to reconstruct the structure of a unique derivation tree for any given sentence. This is required because we will define the information in a sentence in terms of its derivation, and we want the definition

to be unique. A grammar with this property for every sentence in its language is termed *unambiguous.*

The graph grammar above is unambiguous provided we clarify that numbers and labels are always simple terminal elements. For example, we do not use a sequence of terminal digits to represent a number which is written with several digits in decimal representation. If we were to do this, the language would be ambiguous wherever two such numbers appeared in a row, e.g., 12 and 3 versus 1 and 23. Without an intervening delimiter there is no way to determine the interface between the clauses. Instead, we allow N distinct characters in the `Node` term above. For a similar grammar which allows an unbounded number of `Node` terms, the set of terminals may be countably infinite.

The structure grammar described thus far is not sufficient for our purposes. Sentences of our languages must describe more than structures; they must also describe the input data to the estimator. The compositional way to describe data which is interpreted in terms of a model is combinationally, with a two-clause sentence describing a structure and a *realization.* The first clause can describe a structure which is either a model for generating the data or a "structural parameter" in such a model. This will be a sentence in a "sublanguage" for describing objects in the structure space. The second clause then must describe whatever other information is required to specify the input data given the model. The interpretation function gives a problem-specific definition of how this information is combined with the structure to identify the input data. The exact nature of these two components and their relation will vary with the application, and there is a very wide range of possibilities here. The general guideline for the realization clause is that it specifies one observation among the full set of variations which

are compatible with the structure. The examples in the next four chapters are the best guide we can give for indicating of the flexibility of this combination.

The root node of our data-describing grammars will be termed $S(Z, \theta)$, indicating that the grammar must be able to generate sentences which jointly describe any input, Z with various structures θ. This node can be rewritten as a combination of two clauses which describe a structure, θ, and its realization, which specifies Z given θ. This second clause we denote $S(Z|\theta)$, and the root production is

$$S(Z, \theta) \rightarrow S(\theta)\ S(Z|\theta)$$

$S(\theta)$ then expands as the root node of the sublanguage for describing structures. In the FSM example, this will be

$$S(\theta) \rightarrow \texttt{Graph}$$

using a graph grammar based on the one above. The realization of a graph as observation data is described with a `Route` term in Chapter 4.

$$S(Z|\theta) \rightarrow \texttt{Route}$$

In other applications, such as the cluster analysis example in Chapter 5, we will not be strict about the separation between the description of the model and the realization at the root node. The clauses which would assemble into the two primary clauses above can be interleaved and distributed throughout a sentence as long as the grammar and interpretation allows the structure and data to be determined uniquely.

We term the interpretation function from sentences to data, f_Z. The requirement that f_Z exists ensures that the "noise" in the input is fully and exactly

described in $S(z,\theta)$. This is a major difference from the descriptions typical of syntactic pattern recognition. This requirement will result in an information penalty if a great deal of "noise" must be invoked in order to describe the input in terms of θ. We will often be interested in just the subset of the language which describes the given input data, and define for any input, z,

$$L_z = \{S | f_Z(S) = z\}$$

Any instance of an MI estimation problem, as defined by the input, z, will engender an instance of a combinatorial optimization problem in the space L_z. We also have occasions to refer to the function $f_\theta(S)$, which we define as interpreting just the structural clause, $S(\theta)$, of a sentence, to give an element of the structure space.

To summarize, phrase-structured grammars, perhaps with context-sensitive dependencies, are the natural formal language model for structure estimation. The particular grammars chosen should be unambiguous, and allow a homomorphic structural relation between the language and the structure space. The adaptability of this class of grammar allows languages to be designed for virtually any application we can describe in natural language. However, there is a certain amount of "art" in the generation of an appropriate grammar for any particular estimation problem, and the choice is never unique.

A conceptual limitation of the method is that a countable description language requires that there be a countable structure space. Thus a real-valued parameter in a structure can not be estimated to arbitrary precision, and real-valued numbers can not be used as input data unless they are discretized to a countable number of possibilities. This is not a practical limitation however, as the actual optimization of the information measure will necessarily be performed on a computer which is also limited by the countability constraints.

3.2 Information Measures and the MI Estimator

In this section, we consider the properties of the information measure which is defined on the sentences of the language, and define the Minimum Information Criterion for estimation. Although information measures for languages generated by PSG's are easily and naturally defined, we have not been able to find a reference explicitly using this notion. (This is probably because the asymptotic properties of the measures, as sentence length increases, do not have the useful entropic properties of the corresponding notion for Markov sources.) We will therefore develop a notion of information from those developed for other contexts. Because there is no universal notion of information, appropriate information measures, like appropriate languages, are not unique. We wish to allow information to be defined appropriately for each application, in accord with Bateson's rather general definition, that information is "a difference that makes a difference" [1972, p. 315]. Intuitively, and in various formal incarnations, the notion of information satisfies two properties, *additivity* and *comparability*, in ways to be discussed below. We wish to relate these properties to the two type of primitive production rules allowed in our grammars.

Kolmogorov [1965] discusses and contrasts three types of formal information measures:

(1) *Combinatorial*, based only on the number of possibilities;

(2) *Entropic*, based on the probabilities of the different outcomes; and

(3) *Algorithmic*, based on the computational complexity of each outcome.

We will allow these three types of measures to be combined in defining information measures for the strings of a language. No specific information measure is forced

by the techniques. The system designer has the freedom to select appropriate information measures in much the way that classical estimation theory allows one to specify relevant probability measures.

There are at least two other formal notions of information. Fisher [1925] introduced a notion of *statistical information* as the inverse of the covariance of an estimate about a true parameter. It can be related to the entropic measure, as in Kullback [1959]. A notion of *useful information* in a string, relative to a class of models, is defined by Rissanen [1986] as the amount by which a string can be compressed into a shorter string using models in the class. We will not pursue these two notions.

The combinatorial notion of information originates with Hartley [1928], and is defined for communication contexts in which one of a number of outcomes may occur. If there are N possible outcomes, the combinatorial information content of an outcome is $\log N$, and is independent of which outcome occurs. The rationale given for a logarithmic measure is that it satisfies the intuitive notion of additivity when different possibilities combine independently in the construction of more complex possibilities. Furthermore, the monotonicity of the logarithm gives it a comparative property.

The entropic notion of information was developed independently by Shannon [1948] and Wiener [1948]. It is suitable for situations in which a probability measure is defined over a space of possible outcomes. The self-information in the occurrence of an event X with probability $P(X)$ is defined as $-\log P(X)$. This is now a function of the event, and is again justified on grounds of additivity and comparability. This notion of information has been developed into an elaborate system of concepts and relationships; see e.g., Gallager [1968]. In the case of

a finite number of equiprobable possibilities, the entropic notion reduces to the combinatorial notion.

The algorithmic notion of information is philosophically the closest of these three notions to our requirements. It was proposed independently by Solomonoff [1964], Kolmogorov [1965], and Chaitin [1966]. A generally acceptable set of definitions has not yet crystallized, and various approaches are given in Kolmogorov [1968], Fine [1973], Chaitin [1977], and Solomonoff [1978]. The central idea is to measure information of individual sentences of a language in terms of the lengths of Universal Turing Machine (UTM) programs which can generate the sentence. Information in this sense is a function not only of the particular item whose information is measured, but also of the particular UTM. A UTM provides an interpretation for representing strings of a language with other strings which are interpreted as *programs*. There is no unique UTM however, so algorithmic information is relative to what amounts to a particular language and interpretation, rather than a probability distribution.

Algorithmic information can be shown not to be computable, so it is not likely to be of great use in specific estimation problems. However, in principle it could be incorporated into our estimation framework if it were provided by an oracle as a table of values. Lacking such oracles, we restrict ourselves mainly to combinatorial and entropic information measures in defining information on sentences. The relation between algorithmic information and MI estimation is discussed further in Section 8.2.7.

Our definition of an information measure requires that it be a function from a language to the real numbers, defined recursively on the sentences of a language.

Following the compositional form of sentences and interpretations, we require information to be defined on clauses in terms of the information in their constituents and the type of production rule used to form the clause. Corresponding to the two primitive types of production rules, *combinational* and *alternative*, there are two forms for combining information when a clause is constructed, an *additive* rule and a *comparative* rule. For rules which concatenate subclauses, the information in the clause is the sum of the information in the subclauses.

$$I(A) = I(B) + I(C) + \ldots + I(D) \qquad \text{whenever} \quad A \rightarrow B\ C \ldots D$$

Note that for this to agree with our intuitive notion of information, it assumes a notion of *independence* between the combined clauses.

The second information rule applies when an *alternative* production rule generates a clause. We wish here to allow comparisons among the different outcomes allowed by the rule. As certain alternatives are judged to be more complex, more unlikely, or more random, than others, we wish to measure their description as containing more information. In general we will allow the rule

$$A \rightarrow A_1 | A_2 | \ldots | A_n$$

with n alternatives, to be associated with a table of n arbitrary constants, $\{I_i\}$, determining the information content associated with each of the alternatives, A_i. To this we add the information in the particular subclause chosen, to determine the information in the clause A.

$$I(A) = I(A_i) + I_i \qquad \text{when } A \text{ is rewritten as } A_i$$

With a combinatorial notion of information, $I_i = \log n$. For an entropic notion, probabilities $P(A_i)$ must be given for the n alternatives, and $I_i =$

$-\log P(A_i)$. These two measures induce a form of normalization for the information measure—the different values may not all be chosen independently. The analogous property does not hold for certain definitions of algorithmic information, and we do not require it here as a condition on the I_i values. However, we point out that one must be careful if choosing arbitrary values of I_i to balance the different types of terms reasonably, or a single term might dominate the expression when they are added. For this reason, we stick close to combinatorial and entropic measures below, but are willing to approximate them when convenient.

Finally, to complete a recursive definition of information, we need a *base clause* defining the information in the terminal symbols of the language. The appropriate value here is zero!

$$I(A) = 0 \qquad \text{when } A \text{ is a terminal symbol}$$

This may appear surprising at first, but given that we have defined information in terms of which rewriting rules apply, the particular terminal symbols in the sentence are redundant. They add no information.

Note that the additive information law has the effect of preferring nonredundant models, and models with "independent" clauses. For example, if a clause is formed by repeating a subclause verbatim a number of times, the intuitive notion of information does not multiply, but the additive rule has the effect of making the redundant statement more costly. If the interpretation of the language is such that the two clauses describe the same input, the redundant models can never be selected by the MI criterion.

Entropic measures will be used frequently in the following chapters. They have the virtue of allowing asymptotic consistency of the estimators to be shown

in certain applications in which MI estimators can be designed to operate on larger and larger inputs. They also provide a way to specify nonzero probabilities to a countably infinite set of alternatives. If, for example, the term A is given the production rule

$$A \rightarrow A_1|A_2|A_3|\ldots$$

the combinatorial approach is not available, as the logarithm of the number of possibilities is infinite. But by selecting a probability distribution over the positive integers, we can let $I_i = -\log_2 P(i)$. Rissanen [1983], based on universal binary codes developed by Elias [1975], suggests a particular distribution for positive integers in which

$$P(i) = 2^{-(\log^* i + \log_2 c)}$$

where $c \approx 2.865$. Rissanen defines $\log^*$ (nonstandardly) as

$$\log^* i = \log_2 i + \log_2 \log_2 i + \log_2 \log_2 \log_2 i + \ldots$$

summing all the positive terms. From this we define

$$I^*(i) = \log^* i + \log_2 c$$

which is reasonably approximated to give

$$I^*(i) \approx \log_2 ci$$

Note that it has the effect of making smaller integers less costly to incorporate in a description than larger ones. Elias and Rissanen demonstrate that this distribution has a certain asymptotic universality with respect to this property, and Rissanen makes extensive use of it, as he codes all structures as integers.

This measure can be used for the superstar abbreviation which summarizes an infinite number of alternatives. Suppose the term A expands into n clauses of type B in the application of the rule

$$A \rightarrow B^*$$

We can measure

$$I(A) = I^*(n) + \sum_{i=1}^{n} I(B_i)$$

where the $I(B_i)$ terms depend on how each of the B clauses expand. (This is not exactly correct, as I^* was defined above for positive integers, and we are using it here for nonnegative integers. Rather than introduce separate notation, we can read $I^*(n+1)$ for $I^*(n)$ in these cases.) Because the I^* term assumes a probability distribution which decreases monotonically with n, this measure will often be inappropriate. In some situations, we may prefer to use an entropic measure based on a probability distribution which peaks at some particular number of B's.

For infinite alternatives, another form of information measure is uniform up to some bound which depends on each sentence. As each sentence is finite, there is a maximum value, N, for the values of n over the set of expansions of A in any given sentence. A combinatorial measure of information,

$$I_N(A) = \log_2 N + \sum_{i=1}^{n} I(B_i)$$

is natural if we have no preferences within this set. However, this is not a recursively specifiable information rule, as $I(A)$ for a given term depends on N, which is a property of the whole sentence, not just this clause. This is easily remedied with a context-sensitive rule

$$A \rightarrow B \mid BB \mid BBB \mid \ldots \mid \overbrace{B \ldots B}^{N}$$

which requires that N be specified elsewhere in the sentence with a rule

$$N \rightarrow 1|2|3|\ldots$$

The net effect of this is to add $I^*(N)$ to the total information just once, and allow a combinatorial (or other) measure over a finite set of alternatives, each time A appears.

In situations where our grammar allows multiple descriptions of the same interpretation, we may want to modify the definition of an information measure. For example, if we wish to use a combinatorial measure of the information in a term A, but we know that our language allows k equivalent descriptions of each interpretation, rather than add constraints to the grammar we may adjust $I(A)$ by subtracting $\log k$. This amounts to a combinatorial notion of the number of interpretations, rather than the number of alternative clauses. Formally, this is carried out by allowing a third type of rule, defining $I(A)$ as the information in the set, rather than the sequence, of constituents of A. It can be viewed as an abbreviation for a different grammar which describes structures in a unique way.

Finally, we present the minimum information estimator. We select the sentence of minimum information which describes the input data, and extract the structure it uses as the estimate.

$$\hat{\theta}(Z) = f_\theta(\operatorname*{argmin}_{s \in L_Z} I(s))$$

Because our first production rule is

$$S(Z,\theta) \rightarrow S(\theta)\ S(Z|\theta)$$

the additive information rule for concatenation rules automatically rephrases the MI estimator as

$$\hat{\theta}(Z) = \underset{\theta}{\operatorname{argmin}} [I(S(\theta)) + I(S(Z|\theta))]$$

where it is understood that the two clauses in this expression are part of the same sentence, $S(Z, \theta)$. This gives the essential tradeoff: the first term emphasizes simple structures, while the second term encourages good fit.

In summary, we allow a very general class of measures as information measures. They are defined recursively on derivation trees and incorporate additive and comparative notions. A large information content can correspond to notions of complexity, improbability, and/or randomness. Because we do not require information to have the normalization properties of an entropic measure, we gain a very wide class of estimators. When information is measured as length, as is natural with prefix-free binary codes, the MI estimate seeks out the structure coded in the shortest description of the data. Rissanen develops this special case of MI estimation, emphasizing the coding point of view. We have recommended a clear separation between descriptive formalisms and information measures for reasons discussed in Section 2.11.

3.3 Optimization

Given a language, an information measure, and an input, the problem of how to find the particular statement in the description language with minimum information now looms before us. In principle, the problem can be solved as long as the interpretation function on sentences is computable, and the language is recursively enumerable in order of increasing information. The language can be

searched in order of simplest to most complex until the first statement is found which describes the input. An upper bound on the required computation time can be found in applications which allow at least one description of any input to be easily determined. For practical purposes however, enumeration is not a reasonable option as the number of sentences of a given measure generally grows exponentially with the size.

In the types of applications we have considered, this pattern is typical, and we can expect the optimization problem to be NP-hard (see e.g., Garey and Johnson [1979]). Therefore we will not seek an algorithm which is guaranteed to exactly optimize the MI criterion, in any but the most trivial problems. Instead, we aim for the more modest goal of finding an algorithm which will find reasonable solutions in a reasonable amount of time. The full range of heuristic techniques for finding approximate solutions to NP-hard optimization problems should therefore be considered.

A widely applied and often successful method is the *local search* technique. A set of *transformations* for incrementally improving an estimate are repeatedly applied until a local optimum is reached from which none of the transformations result in improvement. As it is straightforward to compare the information measure of two statements of the input once they are specified, local search techniques within L_Z are appropriate for consideration. These techniques require that one prespecify a set of transformations, $\{T_i\}$, which when given a statement in L_Z, "generate" alternate statements about Z in L_Z.

In addition, given Z, one must be able to form at least one statement in L_Z from which to begin searching. Such a statement might involve the simplest allowable structure, and describe Z entirely with the realization clause. At the

other extreme, it may incorporate a very complex model which can only generate Z, so that only the simplest realization clause is required.

The algorithm proceeds in general by transforming a candidate sentence, s, (or the members of a set of candidates) to determine related sentences $\{T_i(s)\}$. Of these new statements, one or more for which $I(T_i(s)) < I(s)$ are adopted as new candidates. In one version of the method, the set of transformations is scanned in some prespecified sequence until the first is found which results in some information improvement. In the "greedy" approach, the entire list is scanned and the best sentence or sentences in terms of the information measure are adopted as new candidates. The procedure terminates at a local minimum of I with respect to the transformations.

There are three general difficulties with local search techniques. Because this is a "heuristic" approach, we can not expect that the global optimum will be found, and the quality of the local optimum relative to the global optimum can not usually be determined. Secondly, it may be quite difficult to decide on a set of transformations to implement; usually heuristic arguments are the only guide. Thirdly, these algorithms are quite difficult to analyze before they are implemented, so it is typical that the set of transformations must be adjusted after initial trials. Nevertheless, the algorithm is generally easy to implement, and in many situations, the local optimum is satisfactory. In addition, local search techniques have the advantage that they easily accomodate arbitrary constraints in the search.

As an aid in selecting a set of local transformations, we can suggest that the grammar for describing structures be used as a guide. Syntactic operations which insert or delete clauses, or replace clauses with alternatives, are suggested

by direct examination of the production rules. More complex transformations may merge, split, or variously exchange parts of different clauses. If descriptions are structurally homomorphic to their interpretations, then these syntactic operations will correspond with various structural changes in the estimated structure. Associated with changes in $S(\theta)$ must be related changes in $S(Z|\theta)$ which insure that $T_i(s) \in L_Z$. Conversely, if changes are made in the $S(Z|\theta)$ clause, $S(\theta)$ will require compensation. The examples in following chapters operate directly on the structure clause, and then modify the realization clause appropriately.

In general the syntax will suggest an enormous number of possible transformations, of which only a small fraction can be implemented. One formal condition is that the transformations should be complete in the sense that the entire space can be reached from the starting point. There is little else that can be said in general however, and at this point an understanding of the problem domain must be invoked in order to select effective transformations.

The above methods, because of the syntactic nature of the search space, can be seen as special cases of *genetic algorithms* [Holland 1975, Davis 1985]. The genetic approach treats descriptions of solutions as analogous to chromosomes, and generates new candidate solutions from existing candidates via mutations, crossovers, and inversions, invoking the metaphor of evolution through adaptation. While the transformations we have implemented transform a single sentence into another sentence (i.e., mutation), the genetic approach also suggests transformations in which parts of two distinct sentences are combined (i.e., crossover) to give "offspring" which may combine favorable portions of the two "parents". Although we have not employed transformations of this class, they may be valuable

in other applications, as they have been observed to allow escapes from certain types of local optima.

Given that approximate methods are being employed in the optimization, it is tempting to inject approximations at another level as well. Rather than approximately minimize the exact MI criterion, it is usually much easier to direct the search with an approximation to the criterion which drops various small order terms. The logarithmic information terms created by superstars in production rules are tempting candidates when they appear in addition to a linear term. The validity of these approximations is arguable in each example, but no serious harm appears to result from the approximations made in the various case studies.

In the case studies of Chapters 4–7, greedy local search algorithms are used to optimize the MI criterion, with good success. Other optimization techniques will certainly be appropriate for other problems. The particular nature of the relations between statements, inputs, and interpretations will determine what is appropriate for each application. Branch and bound algorithms, stochastic relaxation, and dynamic programming techniques, for example, may all be suitable in certain cases.

3.4 Special Cases

It is insightful to relate this estimation technique to more traditional estimation techniques. Because general description systems and information measures are extremely flexible tools, it is not surprising that many other types of estimation criteria can be phrased as MI criteria. In this section, we point out how the various structure estimation techniques of Chapter 2 are special cases of MI estimators. In some cases a fairly contrived language and unnormalized information

measure is required, however. In framing these methods into the terms of the complexity and fit measures of MI estimation, we will see that the methods fall into the following trichotomy:

1. Maximum Likelihood estimation and Popper's Falsifiability criterion focus on fit between models and data, while ignoring complexity.
2. Ockham's Razor, the method of choosing a substructure of a most complex structure, the method of choosing the simplest model with error in some tolerance range, and hypothesis rejection, all focus on choosing the simplest structure in some set of compatible structures. Fit is only examined in a qualifying manner, as *acceptable* or *unacceptable*.
3. MAP estimation, the "knee" of the curve technique and Minimum Description Length estimation make a quantified balance between simplicity and fit.

As Rissanen [1978, 1983] points out, a Maximum Likelihood (ML) estimator is the special case of an MI estimator in which the information measure on the fit is entropic, based on a "transition mechanism" with a conditional probability $P(Z|\theta)$. The information measure on structure descriptions is ignored or treated as constant. Letting

$$I(S(Z|\theta)) = -\log(P(Z|\theta))$$

and

$$I(S(\theta)) = C$$

the MI estimator reduces to the ML equation:

$$\hat{\theta}(Z) = \underset{\theta}{\operatorname{argmax}}\, P(Z|\theta)$$

Popper's falsifiability criterion can also be formulated as an MI estimate, at least in the case where the number of observations compatible with each structure

is finite. This criterion can be interpreted as choosing the structure, out of those compatible with the data, which allows the smallest number of other possible observations. It ignores the complexity of the structure. This can be formalized as a variant of the ML criterion in which a combinatorial measure of information is used to measure the description of the given input out of the set of possible inputs. If there are N_θ possible observations compatible with θ, and we measure

$$I(S(Z|\theta)) = \begin{cases} \log N_\theta, & \text{if } Z, \theta \text{ compatible} \\ \infty, & \text{otherwise} \end{cases}$$

and $I(\theta) = C$, then the MI criterion reduces to

$$\hat{\theta}(Z) = \underset{\theta}{\operatorname{argmin}}\, N_\theta$$

The effect of the ∞ term is just to restrict the set of structures to those compatible with the data. We include it here for completeness, but its effect is more likely to appear in the grammar and interpretation functions.

Ockham's Razor is a dual to the ML estimator. It selects the simplest structure compatible with the data, by some interpretation of compatibility, and ignores fit. To restrict the estimation to compatible structures, we define

$$I(S(Z|\theta)) = \begin{cases} C, & \text{if } \theta, Z \text{ compatible;} \\ \infty, & \text{otherwise} \end{cases}$$

Simplicity of structures is quantified as the minimum information required in their description. The MI criterion then reduces to

$$\hat{\theta}(Z) = \underset{\theta}{\operatorname{argmin}}\, I(S(\theta))$$

where the minimization takes place over only those structures compatible with Z.

The method of selecting a substructure out of a largest allowable structure can also be seen as choosing the simplest structure compatible with the data. Recall that this is the method of using traditional estimation techniques within the parameter space of a complex structure, and eliminating from the estimate those portions of the structure with coefficients "sufficiently close" to zero. This can be interpreted as a variant of the Ockham's Razor approach. Complexity is being measured as the number of components in the estimated structure. A structure is considered compatible with the data if it contains at least all the components with coefficients not sufficiently close to zero.

For example, the following method is often applied to the problem of estimating the order, θ, of a polynomial which fits a data vector, Z. Given an upper bound, N, on the order, coefficients, $c_0, c_1, \ldots, c_N$ are estimated using least squares techniques. A tolerance, ϵ, is given, and the lowest order, θ, is chosen for which $c_i < \epsilon$ for all i in the range $\theta + 1 \leq i \leq N$. This is equivalent to an MI estimate of θ where $I(S(\theta)) = \theta$ and

$$I(S(Z|\theta)) = \begin{cases} \infty, & \text{if } c_i > \epsilon \text{ for some } i > \theta \\ C, & \text{otherwise.} \end{cases}$$

The method of choosing the simplest structure with an acceptable fit is another variant on this method. Here $I(S(Z|\theta))$ is again quantified with zero or infinity, but the criterion of compatibility depends on mean squared error, or some other measure of fit. Any complexity measure on θ may be specified, and the effect of the MI estimate is to choose the simplest structure with an acceptable fit.

Hypothesis rejection is yet another variant on the method of choosing the simplest structure with an acceptable fit. The fit here is defined as acceptable

when the test statistic falls outside of the critical range, and measured as infinite when in the critical region. The null hypothesis is measured as simpler than other hypotheses, and the MI principle reduces to Fisher's hypothesis rejection technique.

Rissanen [1978, 1983] points out the relation between MI estimators and Bayesian estimators. If an entropic notion of information can be used to define $I(S(\theta))$ and $I(S(Z|\theta))$, then the MI estimator becomes the logarithmic form of a MAP estimator.

$$\hat{\theta}(Z) = \underset{\theta}{\operatorname{argmax}} [\log P(\theta) + \log P(Z|\theta)]$$

By identifying the corresponding components, we see that the MI estimator is often isomorphic to a MAP estimator in which an *a priori* distribution over the space M is defined by the grammar for $S(\theta)$.

In many applications Bayesian estimation would be a natural tool except that we can find no grounds for choosing any particular *a priori* distribution in a complex hierarchically-structured space in which different models have vastly differing numbers of degrees of freedom. The formal language technique suggests that by describing the space syntactically we can induce an implicit distribution by normalizing

$$P(\theta) = -\log I(S(\theta))$$

Selecting concise grammatical structures to describe the types of model structures which the system designer expects to find corresponds to reducing their *a priori* probabilities. This strict Bayesian interpretation is not always so natural or insightful however. In particular, it is not generally possible to construct a probability distribution from a general information measure, unless $\sum_\theta 2^{-I(\theta)}$ is finite so it can be normalized.

The "knee of the curve" technique selects a structure according to a specified slope in the error vs. complexity curve. This can be effected by weighting the $I(S(Z|\theta))$ and $I(S(\theta))$ terms with α and β respectively. Minimizing the weighted sum then is equivalent to choosing the slope $-\frac{\alpha}{\beta}$.

Minimum Description Length estimation, as formulated by Rissanen, is another form of MI estimation which balances between simplicity and fit in a quantified manner. As discussed in Section 1.3, we have generalized the method by focusing on flexible formal languages, distinct from information measures. Another difference is that we do not require the information measures to be normalizable.

Maximum Entropy estimation is shown to be a special case of MI estimation, at least for the case of a finite sample space, in Rissanen [1983] and in Feder [1986].

In summary, the various approaches for estimating structure which have appeared in the literature all imply simplicity, fit, or both, as goals. Expressly stating the balance between the terms implicit in these techniques exposes their relations, and their status as special cases of a general MI estimator.

Chapter 4

FINITE STATE MACHINES AND MARKOV SOURCES

In this chapter we consider the problems of estimating the structure of a Finite State Machine (FSM) and a Markov Source (MS). We also consider a problem, motivated by considerations in the Appendix, of estimating a set of FSMs with interleaved outputs. In Section 4.1, we examine existing methods for solving these and similar FSM and MS problems. With this background, the description-based MI technique is applied to the FSM problem in Section 4.2. Languages for describing FSMs are presented along with the related information measures, and methods for optimizing the MI criteria are discussed. It is shown that the resulting MI estimators have a number of desirable properties. In section 4.3, the FSM model is extended, with a probabilistic component, to MS models, and a form of optimality is demonstrated. The multiple FSM problem is presented in Section 4.4, and the results of a simple simulation are shown.

The basic FSM problem was illustrated in Section 1.4. The estimator is given a string (or a set of strings) which is modelled as the output of a grammar. The estimator must produce a grammar capable of generating the strings. In the form we consider, there is only "positive evidence". A more general problem allows

"negative evidence"—we could be presented with a second set of strings and told that the estimated grammar may not generate them. In either case, the problem will be underdetermined. As illustrated in Section 1.4, there are generally an infinite number of grammars compatible with given observations in any nontrivial problem. Our approach is to regularize the problem by introducing simplicity and fitness measures based on descriptions of grammars and descriptions of the input as realizations of grammars.

4.1 Approaches to Grammatical Inference

The relation between grammars, FSMs, and automata are presented in many sources, e.g., Kohavi [1970]. The problem of estimating a FSM structure which generates observations is equivalent to the problem of inferring a *regular grammar*. The "decisions" which occur at the branch points of these models are considered unknown, but nonprobabilistic. If probability distributions are assigned to apply independently at the various decision points in the structures, Markov Source (MS), or *stochastic regular grammar* estimation problems result. A third method of modelling the "decisions" is to specify an input string to the model, and have a rule for making state transitions as a function of the current state and the next input symbol. This gives rise to problems of *automaton identification*, rather than grammatical inference. The estimator is then given, or may create, an explicit control sequence of input symbols along with the observation sequence. Hennie [1968], Trakhtenbrot [1973], and Rivest [1987] contain discussions of methods for these problems. We do not consider these controlled FSMs here.

Another class of problems related to FSM and MS estimation involve estimation within the set of *phrase-structured grammars* (PSG). This class of grammar

is demonstrably more powerful than finite-state grammars [Chomsky 1956]. In Chapter 3, these were presented and used in a nonprobabilistic manner, but if probability distributions are assigned over the various options in each alternative production, a *stochastic context free grammar* results which assigns a probability to each string it generates.

An enormous variety of grammatical inference techniques have been published for grammars of the above classes. An excellent recent review of this literature is given by Angluin and Smith [1983]. Earlier reviews can be found in Fu and Booth [1975], and Biermann and Feldmann [1972]. A large portion of the literature, beginning with Gold [1967], concerns *enumeration* methods in which all grammars in a class are examined in order of increasing complexity to find the simplest which is compatible with the data. While not computationally tractable, these methods can at least show certain negative results when the data does not contain enough information for any algorithm to infer the simplest grammar. From our point of view, these methods are inadequate in that they ignore the issue of *fit*, except in an all or nothing manner. Many other methods are of an *ad hoc* nature, or apply only to restricted subsets of the grammars in a class. We will not discuss these here as we are interested in general techniques which address the issues of simplicity and fit.

Summarized below are references which make tradeoffs between simplicity and fit, and can be placed into a MI framework. Their authors do not present them in terms of a formal descriptive framework for describing data and grammars, but they can be translated directly into those terms by constructing the authors' tacit description languages. These works all concern probabilistic grammars, which

define probability distributions over the language they generate. The input data is measured in terms of these distributions to determine entropic measures of fit.

Solomonoff [1964] addresses the general question of induction, and develops an algorithmic notion of *a priori* distribution, based on the lengths of Universal Turing Machine programs which generate binary strings. He applies this to the problem of extrapolating sample data which can be generated with PSG and Markov-like models. Although Solomonoff's general approach allows arbitrary UTMs to generate data, he suggests that for data which can be generated by these more restrictive models, the shortest UTM program typically will only make use of the mechanisms allowed by these more restrictive models. Accordingly, he develops coding schemes for describing models of these classes and their outputs. We will not summarize these examples here, as the codes are somewhat arbitrary and unenlightening, but we mention them as they were a major influence in our thinking. Solomonoff's examples can be interpreted from the MI point of view as using combinatorial information measures. He also describes a local search technique for one example.

Cook *et al.* [1976] propose an information-theoretic cost function which measures the complexity of a stochastic context-free grammar and the discrepancy between the input sample and the grammar. This cost function is used to direct a local search through a set of grammars, starting at a very complex grammar constructed to have an exact fit to the input data. The local transformations employed have the effect of introducing new nonterminal symbols, combining rules into alternative productions, and removing superfluous productions. In every respect, their method fits into the framework presented here, even to the extent that they note the complexity measures on grammars can be derived through a

grammar for describing grammars, as we would recommend. Cook *et al.* apply the method to a variety of examples with good results, finding simple grammars with good fit to the data. Certain grammatical configurations which confuse the local transformations from finding global minima are also noted. Horning [1969] is cited by several authors for a similar approach, but is reported to use an exhaustive search method to search the space of grammars.

Gaines [1976] presents a framework for MS estimation which involves specific complexity and fitness measures, but does not try to trade off between them. Rather than propose a combined measure to optimize, he separately measures complexity as number of states, and fit entropically, and then plots the fit/complexity curve. *Admissible models* are defined as those which are optimal in fit among those of a given complexity, or minimal in complexity among those of a given fit. An algorithm generates the simplest admissible models by enumerating, and testing, all structures up to a half dozen or so states. "Knee of the curve" techniques are invoked in several examples to select a particular structure. Gaines applies the method to examples of MSs and automata.

Van der Mude and Walker [1978] address the same MS problem, but give a specific method of balancing simplicity and fit. Transition probabilities in the grammar are restricted to rational values, and complexity is measured as the sum of the denominators required to express the probabilities in lowest terms. This gives a measure they term a probability, but which is not normalized. When the conditional probabilities for transitions out of a state are uniform for each state, their complexity measure reduces to a count of the number of arcs in the Markov source diagram. They develop a branch and bound algorithm, using state-splitting transformations, to exactly optimize the criterion, and apply it to small examples.

Rissanen [1983, 1986] uses the MDL criterion for adaptive data compression purposes. He presents an algorithm to encode an input string into a shorter string, by making use of redundancies in the strings which can be captured by Markov models. Rissanen is not directly concerned with the problem of estimating the structure of a Markov source, but an estimator of this type is embedded within his data compression algorithms. In keeping with the integer-based approach outlined in Section 2.11, he assumes an enumeration is given for the set of FSMs. FSMs are ordered and indexed in any way which gives FSMs with more states higher indices than an FSM with fewer states. This type of integer description contrasts strongly with the formal graph language method for describing FSMs outlined in Section 3.1. There is no structural relation between FSMs and their representations, and the method leads to peculiar estimates, except asymptotically. This is not a criticism of Rissanen's work however, as his goal is asymptotically optimal data compression, not reasonable structure estimates from small data sets.

Given the integer enumeration of FSMs, the information to describe an FSM is specified by the I^* function of Chapter 3, and the information in a string given the structure is specified entropically. The FSM which minimizes the total information is then a MI estimate of the structure, but one that incorporates a peculiar description mechanism and complexity metric. In general, we can expect such estimates to have little correlation with the structures we might choose given data, such as in Figures 1.1b and 1.1c of Chapter 1.

Rissanen [1986] allows only a restricted class of FSMs, odd from our structure estimating point of view, though reasonable for adaptive data compression purposes. The structure of an FSM is constrained so that out of each state, exactly one arc exits for each label from the output alphabet. Furthermore, each of these

arcs must have a nonzero probability, so that any string of observations may be generated starting at any given state of the FSM, and the resulting state of the FSM is uniquely defined. The advantage of this constraint is that from any state, the next observation may always be described with a finite code length, or amount of information.

This gives a form of one-step predictor which is used to avoid separately describing an FSM while describing its output. The first symbol may be coded arbitrarily. For each symbol from this point on, the compressed data for the first n input symbols determines the observations uniquely. This, in turn, defines a member of Rissanen's restricted class of FSMs, along with its "current state," via the MDL estimator above. This FSM and its current state are interpreted as a predictor for the $n + 1^{st}$ output character, which is coded using a minimum expected-length code for the arcs out of the current state. As the decoder can follow the same logic, the description of the FSM is never explicitly coded as such, yet it can vary with time adaptively. The cost of this system is in the relatively suboptimal routing codes for small amounts of data. The class of models does not allow a FSM to have states with only one exiting transition, for which the next state and observation is uniquely determined. Some non-zero probability is always assigned to each possible observation, and a nonzero-length codeword is required where another FSM would require no code.

Relative to the above works, we will propose a method of grammatical inference in which a language is explicitly designed for describing grammars in the class of interest. An information measure on sentences of the language then formalizes the intuitive notion of simple grammars. For probabilistic grammars, an entropic information measure is appropriate for describing fit. For FSMs and nonprobabilistic PSGs, other information measures are required.

4.2 MI Estimation of FSMs

To describe a FSM, we use a language like the Graph language of Chapter 3, which describes labeled directed graphs, making use of the isomorphism between these graphs and FSMs. We will make some modifications to the language to remove the upper bound on the number of nodes. Because we organize this language around the arcs of a graph, the resulting information measure on structures will have a dominant term which is linear in the number of arcs. This contrasts with many other complexity measures which grow with the number of nodes, e.g., those of Gaines, and Rissanen, above. This decision is based on the subjective criterion that as an estimate of FSM structure, a sparse graph is significantly simpler than a dense graph with the same number of nodes, while a dense graph is not significantly simpler than a sparse graph with the same number of arcs.

It is interesting to observe that other compositional alternatives for organizing the grammar around nodes or labels also seem to require information dominated by a term which is linear in the number of arcs. This suggests that the standard node count as a measure of complexity, whatever its merits may be in computational complexity studies, is inappropriate for structure estimation purposes.

To describe a FSM, we can use the following grammar:

$$
\begin{aligned}
\texttt{Graph} &\rightarrow \texttt{Arc}^* \\
\texttt{Arc} &\rightarrow \texttt{Source Sink Label} \\
\texttt{Source} &\rightarrow \texttt{Node} \\
\texttt{Sink} &\rightarrow \texttt{Node} \\
\texttt{Node} &\rightarrow 1|2|3|\ldots \\
\texttt{Label} &\rightarrow A|B|C|D
\end{aligned}
$$

A change of terminology from Graph, Arc, and Node, to FSM, Transition, and State, might be appropriate, but we will continue with the graphical terms. The

set of labels in the rule for `Label` is assumed to match the input alphabet for the data given in any instance of an estimation problem.

As there are three alternative productions in the grammar (including the `Arc`* term), there are three points at which we must specify parameters to the information measure. Lacking strong arguments otherwise, we use a combinatorial notion of information for the `Label` rule, the I^* measure for the `Graph` rule, and the context-sensitive I_N measure for the `Node` rule. We do not claim that these measures can be given more than intuitive justifications, however. These are summed on the derivation tree for a sentence describing a graph. With K arcs and N nodes, allowing an alphabet of M labels, this gives

$$I(\texttt{Graph}) = I^*(N) + I^*(K) + K(2\log_2 N + \log_2 M) - \log_2 K!$$

where the last term corrects for the fact that we are describing a set, not a sequence, of arcs.

By including the $-\log_2 K!$ term, we are making a choice as to how FSM complexity grows with the number of arcs. Very dense graphs become relatively inexpensive if this term is included. Note that once a graph is described to be very dense, it is simpler to describe the arcs missing, rather than those present, and complexity decreases as the number of arcs increases. If this is unreasonable in the application, this term should be dropped. If the term is included, we must ensure that each of the K `Arc` terms be distinct (because we are counting the number of permutations). This results in a constraint that $K \leq N^2 M$, and the information is always positive.

To describe a realization of a FSM as an output string, we make use of the given structure, and describe a *starting state* and a set of *decisions* which together

define a *route* through the FSM. The starting state can be described as a Node term. Each decision can be recorded with an index term, which is interpreted relative to the "current state", as defined by the earlier decisions. The index specifies one of the arcs out of the current state. It may be described as an integer between 1 and J_s, if there are J_s arcs out of the current state, s.

$$\begin{aligned} \texttt{Route} &\rightarrow \texttt{Start Decision}^* \\ \texttt{Start} &\rightarrow \texttt{Node} \\ \texttt{Decision} &\rightarrow 1|2|\ldots|J \end{aligned}$$

We have been somewhat sloppy here, as the value of J depends upon the "current state" in a way which is not reflected in the above grammar. We repair this below. The Graph and the Route together constitute a sentence for describing the input sequence.

$$\begin{aligned} S(Z,\theta) &\rightarrow S(\theta)\ S(Z|\theta) \\ S(\theta) &\rightarrow \texttt{Graph} \\ S(Z|\theta) &\rightarrow \texttt{Route} \end{aligned}$$

As we are describing FSMs, rather than MSs, a combinatorial notion of information is natural for the Start and the Decision terms, and we again use the I^* measure for the superstar. This gives

$$I(\text{Route}) = \log_2 N + I^*(L) + \sum_{s=1}^{N} V_s \log_2 J_s$$

where L is the length of the route (and the input sequence), and V_s is the number of times state s is visited, so $L = \sum_{s=1}^{N} V_s$. Note that this is identical to an entropic measure for a MS with the same graph structure in which the starting state is distributed uniformly, the conditional transition probability distributions are each uniform, and the structure can only generate the input string via one route.

There are several ways the context sensitivity of the `Decision` terms can be made explicit. The clearest is probably to record a sequence of nodes rather than branch indices, and include a constraint that every pair of adjacent nodes must appear as a `Source-Sink` pair in an `Arc` term. A combinatorial notion of information again gives the expression above.

The final information criterion to minimize in selecting a FSM then becomes

$$I(S(Z,\theta)) = I^*(N) + I^*(K) + K(2\log_2 N + \log_2 M) - \log_2 K!$$
$$+ \log_2 N + I^*(L) + \sum_{s=1}^{N} V_s \log_2 J_s$$

with $K \leq N^2 M$ as discussed above.

To see how this criterion trades off between simplicity and fit, we return to the example of Section 1.4. Figure 4.1 shows eleven FSMs which can generate the input

$$Z = ABCABDABCABDA$$

It includes the four figures considered in Chapter 1. Next to each is recorded the information measures $I(S(\theta))$, $I(S(Z|\theta))$, and their total, $I(S(Z,\theta))$. Of these, the structure B has the lowest total, and is therefore most preferred by the MI criterion. We can in fact show with a small amount of enumeration that it is the global minimum for this data, out of all FSMs, not just those shown. This is because if another arc is added in a structure with three or more nodes, then the term in the structural description which is linear in K increases more than the best possible reduction in the route term. So we only need to examine a small number of FSMs with four arcs and at most three nodes. These are A, B, E, F, and G in Figure 4.1. Other conceivable FSMs with more than four arcs and no more than three nodes are ruled out by the following considerations.

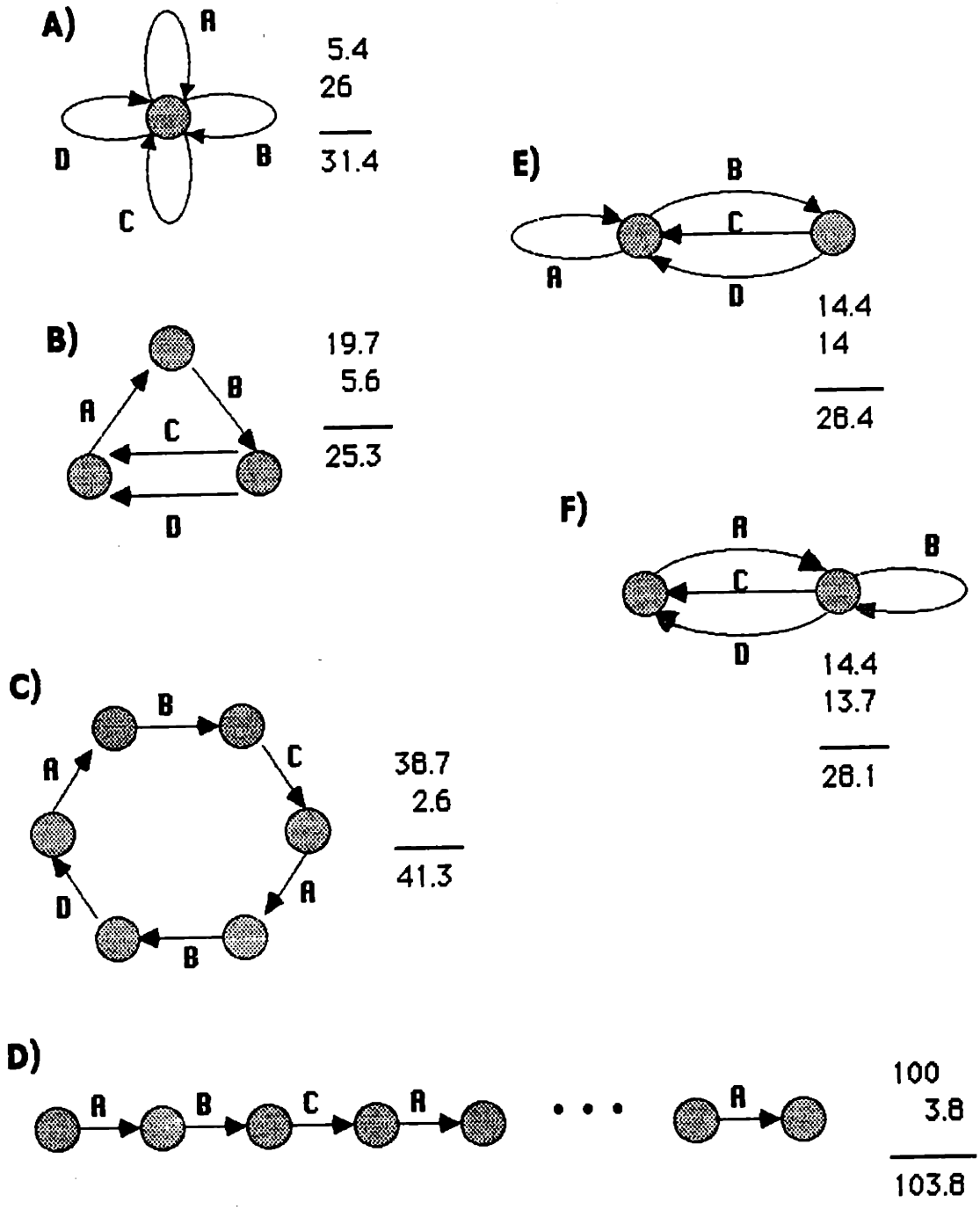


Figure 4.1 Finite State Machines which can Generate the Input String *ABCABDABCABDA*

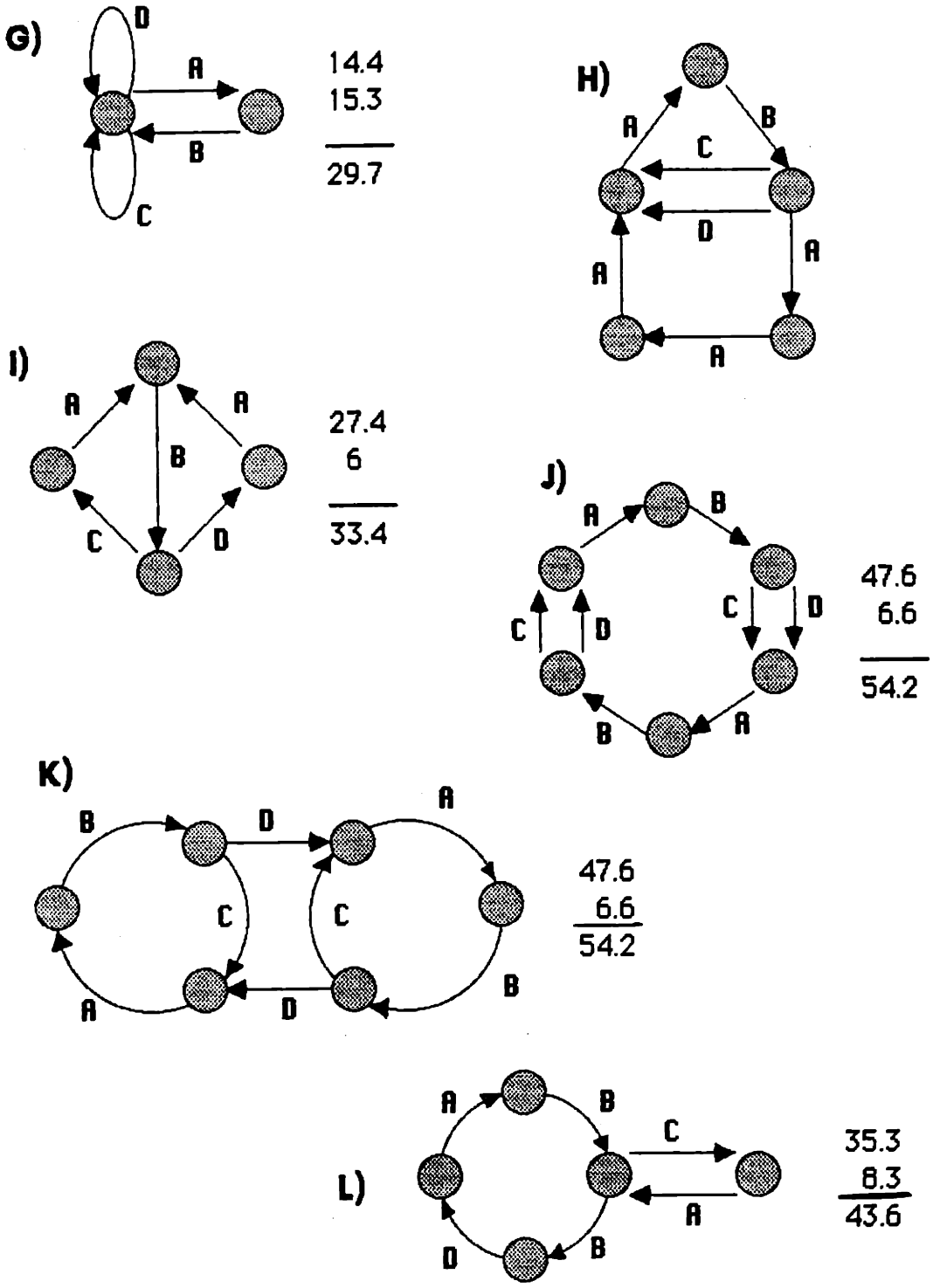


Figure 4.1 Continued Finite State Machines which can Generate the Input String *ABCABDABCABDA*

The grammar above allows graphs which we would never want as FSM estimates, for example with two identical `Arc` terms, or with several noncommunicating components. It is easy to see that the MI estimate will never select a graph with a duplicated arc, as an otherwise similar graph lacking the second arc reduces the information in both the `Graph` term and any `Decision` terms occurring at the source of the duplicated arc. Disconnected graphs are never the output of the MI estimator because the route is restricted to the component connected with the `Start` state, so an estimate which eliminated the unvisited component(s) can be used to describe the input data with a smaller information measure. More generally, it is easy to see that the MI estimate will never include superfluous structure which is not visited in the route, since eliminating it gives a structure which results in a smaller information measure. This rules out Figure 4.1H for this data, because it contains superfluous structure relative to Figure 4.1B. We consider all these "simplicity properties" desirable in an FSM estimator.

Another satisfactory property of our estimator is that it only gives *Nerode minimal* structures. Nerode [1958] defines an equivalence relation on FSMs which partitions the set of FSMs into groups which generate the same set of strings. He shows that within each group there is a unique minimal FSM with the smallest number of states. We can show that MI estimates of FSMs will be minimal in this sense. To illustrate the concepts, consider Figures 4.1I, J, and K. These FSMs all are Nerode equivalent to the Nerode minimal Figure 4.1B.

Nerode's proof involves associating each state in an FSM with the set of strings which can be generated starting at that state. A nonminimal FSM has more than one state from which the same set of strings can be generated. By "merging" states that are equivalent in this sense, the minimal FSM results, with

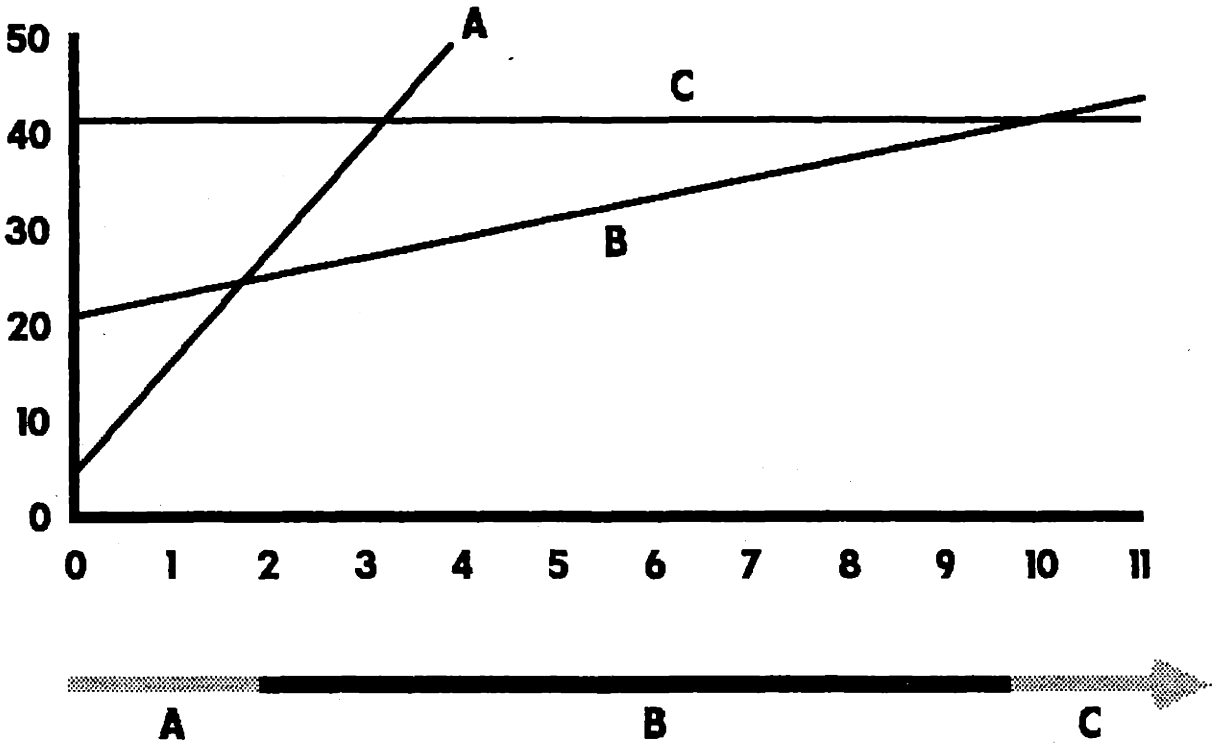


Figure 4.2 Growth of Information as Function of Input String Length, for Input $(ABCABD)^n$, for Three FSMs of Figure 4.1

no two states equivalent. In Figure 4.1I the leftmost and rightmost states are equivalent, and merging them together gives Figure 4.1B. In Figures 4.1J and K, diametrically opposite states are equivalent. The MI estimator avoids Nerode nonminimal FSMs because they require: (1) longer structural descriptions, having more arcs and states than their minimal equivalent; (2) longer descriptions of the starting state, as there are more choices; and (3) equally long descriptions of the decisions, as equivalent states have the same number of outgoing arcs.

Let us return to the FSMs of Figure 4.1 ,and see how they compare as the length of the input is increased. As a concrete example, consider an input string, Z_n, consisting of n repetitions of the sequence $ABCABD$. The terms in the information due to the structure and starting state are fixed, and the length

terms, $I^*(L)$, grow equally, so it is the `Decision` terms we should examine. For FSM A, $J = 4$ for each observation, so the information increases by $12 = 6 \log_2 J$ every repetition. For FSM B, $J = 1$ at two of the states, which require no explicit decision information The information therefor increases by $2 = 2 \log_2 2$ every repetition—one bit each time the state with two exiting arcs is visited. For FSM C, every state has only one exiting arc, so the route contains no information. Plotting the total information measures for these three FSMs as a function of n in Figure 4.2, we see that each has a domain in which it is the preferred estimate. A is chosen for inputs of less than 11 characters, and C is chosen for data larger than 60 characters. In between, FSM B is chosen, out of these three, but we have not shown that some other FSM might not be a better estimate for some values of N.

We can generalize this observation and show that for data generated by *cyclic* FSMs, in which the arcs form a cycle with no branching, the MI estimate is asymptotically consistent. By consistent, we mean that the estimate will be the Nerode minimal member of the class of FSMs generating the data. Note that we can not hope to distinguish between the actual FSM generating the data and others to which it is Nerode equivalent, as they generate exactly the same set of strings. It is reasonable then to give the Nerode minimal member of this set as a canonical form to represent the entire set.

To show that the MI estimate is consistent when the true FSM, X, is Nerode minimal and cyclic, we show that (1) for any other FSM, Y, $I(Z_n, X) < I(Z_n, Y)$ for all n after some n_Y, and (2) for only a finite number of Y does $I(Z_n, X) \geq I(Z_n, Y)$ ever hold. From (1) we see that if the estimator reaches a limit, it is the consistent estimate. From (2), and the observation that $I(Z_n, Y) - I(Z_n, X)$

is a nondecreasing function in n, we see that it must reach that limit. The input data, Z_n, consists of n concatenations of a string Z_1, which consists of the labels of X in order of transitions. Property (2) results from the fact that information in the Arc terms are bounded above zero and additive. To see (1), note that $I(Z_n, X)$ grows at the slowest possible rate, $I^*(n) + C$, because $I(\texttt{Decision}) = 0$. So any Y which was preferred over X for all large n would also have to grow at this rate. Only a cyclic FSM, or one which has additional structure from which the route eventually enters a cyclic substructure, allows this low a rate. This cyclic FSM or substructure can not have fewer states and arcs than X, because X is Nerode minimal, so if $I(Z_n, Y) \leq I(Z_n, X)$ for all $n > N$, then $Y = X$.

This special property for strings generated by cyclic FSMs results from the fact that, asymptotically at least, almost all their information is in their length. A finite FSM is generating a unique infinite string. The repeating strings generated by cyclic FSMs are analogous in this sense to the notion of a nonrandom real in algorithmic information theory.

We have not implemented a search algorithm specifically for this class of structures, but see Section 4.4 for an algorithm for the multiple FSM problem. A plausible approach is to begin with either the simplest or most complex FSM for generating a string, i.e., Figure 4.1A or D, and transform it with state splitting or merging transformations. State splitting is illustrated in the transformation from Figure 1.1A to E, F, or G. These types of transformations are employed with good success by Van der Mude and Walker [1978]. Merging is illustrated by the inverses of these transformations, and in the passage from C to L, in which the two marked states of C are merged.

We note that finding the exact MI estimate may be difficult. Gold [1978] and Angluin [1978] show NP completeness results for the similar question of whether there exists a deterministic automaton with n states compatible with given input/output data.

4.3 Markov Sources

To extend the above ideas to Markov Sources, it is only necessary to incorporate the conditional routing probabilities in the description of the structure, and make use of them in measuring the description of a route. There are many ways to describe the sets of probabilities if we restrict them to rational values.

One method for describing rational values is to use a sequence of decimal or binary digits, interpreted as if preceded with a decimal point (or binary point). Another is to use a pair of integers, and interpret them as numerator and denominator of a ratio. If two integers are used, all rational values can be described, but irrationals are ineffable. If binary or decimal expansions are used, then only certain rationals are describable. For example, $\frac{1}{3}$ is not describable in a finite sentence interpreted as binary or decimal digits. It is not clear what criteria should enter into a choice of a language and information measure for the rationals. Note how the relative information contents of the different values $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{10}$ differ according to the choice. We will not propose a method here, as the one result of this section is independent of this choice and how we measure the information in rational numbers.

The most natural locus for the probability descriptions is in fourth element of each `Arc` term.

`Arc` → `Source Sink Label Probability`

This introduces a consistency problem of making the probabilities of the arcs with a common source sum to unity. Again, there are a number of ways this could be effected, and we will not discuss alternatives here.

Interpretation functions from Graph sentences to MSs are also straightforward to define, in analogy with the FSM case. Note here, that our countable grammar allows MSs with all possible FSM skeletons, but only a countably infinite number of probability distributions. The particular probabilities describable depend on the grammar and interpretation for Probability. We take no stand on this issue here, but simply let $\mathbf{M}$ denote the set Markov sources which can be described with whatever grammar and interpretation is chosen.

The effect of imposing a probability distribution for the exiting arcs of each node is that we can use it in defining an entropic information measure for the Decision terms. Letting $P(N_{k+1}|N_k)$ denote the probability that the $K+1^{\text{st}}$ state is described as N_{k+1} given that the k^{th} state is N_k, we measure

$$I(\texttt{Decision}) = 2^{-P(N_{k+1}|N_k)}$$

This leads to a form of optimality for the MS estimate. If the data, Z_n, is the first n characters of a string generated by an ergodic MS, $X \in \mathbf{M}$, the MI estimate of X will, with probability one, achieve the same asymptotic data compression rate as X. The asymptotically dominant term of $I(Z_n, X)$ is the first term of

$$I(\texttt{Decision}^*) = nH + I^*(n)$$

where H is the source entropy:

$$H = -\sum_i \pi_i \sum_j P(j|i) \log_2 P(j|i)$$

The summations are over the set of states, and the π_i terms are the steady-state probabilities of each state, which are guaranteed to exist by the ergodicity assumption. The expected information per `Decision` using X in the description of Z_n is therefore the source entropy in Shannon's sense, and with probability one can not be reduced with any other MS. Note that this does not guarantee the true MS is the MI estimate, (not even with the probability one caveat), but it is just as good for data compression. By Shannon's noiseless source coding theorem, no other description results in a significantly lower information growth, but other MSs can achieve equally good data compression.

Rudich [1985] and Rissanen [1986] show stronger consistency properties for a reduced set of sources in which the state of the source is a function of the previous state and output letter. These structure estimators can not give the correct structure for most FSMs however, as discussed above.

4.4 The Multiple FSM Problem

In the multiple FSM estimation problem, we model the data string, Z, as the interleaved output of n FSMs, where n is unknown, and part of the structure to estimate. The individual FSMs may have distinct, overlapping, or identical alphabets of labels. In addition to the n FSMs, an assignment function must be estimated which indicates which FSM generated each symbol of Z. The subsequences associated with each of the FSMs must correspond to a valid path through that FSM. This corresponds to a situation in which a set of FSMs operate independently, and their separate outputs are "shuffled" together, into a sequence

preserving the order of each individual FSM's outputs, but not necessarily following any probabilistic interleaving rule. There are no requirements concerning how observations from different sequences are ordered.

Given the FSM descriptions above, a description of a set of FSMs is naturally given by

$$\texttt{MultipleFSMs} \rightarrow \texttt{Graph}^*$$

To describe the data we can continue to describe `Decision` terms as above, but they must be paired with an indication of which FSM changes state. This can be an index into the set of FSMs, with information measured combinatorially.

Summing these terms gives

$$I(Z,\theta) = I^*(N) + n \log_2 N + \sum_{i=1}^{N} I(Z_i, \theta_i)$$

to be minimized over the sets of FSMs. Here N is the number of FSMs in the model, n is the length of the input sequence, and Z_i is the subsequence of the input assigned to the i^{th} FSM, θ_i.

Optimization of this criterion over sets of FSMs is an interesting problem. The local search techniques discussed in Chapter 3 suggest transformations in which arcs are inserted, deleted, or exchanged in the various FSMs. While we are fairly confident that such techniques could be made to work, we follow a slightly different method here. For trial purposes, we have implemented a simpler algorithm which allows insertion transformations only. An arc is inserted either between two existing states, or from an existing state to a newly created state. These transformations may happen to any of the FSMs in a model, or a new FSM can be created as a state with no arcs, before the transformation applies to

it. To guide the search, it operates incrementally, scanning the input data one symbol at a time, always keeping track of "the current state" for each FSM in each possible set. The only transformations which change structure are those in which an arc labeled with the next symbol of the data is inserted from the current state of an existing or newly created FSM. A second transformation just updates the current state pointer when an arc labeled with the next symbol of the data already exists out of the current state. These transformations guarantee the data can be generated by the structure.

We have implemented this in a "best first" manner. After each symbol is processed, only the best few estimates, in terms of an information criterion, are kept. These are then transformed according to the next symbol. The number of structures to keep at each iteration is a parameter of the algorithm. This technique can lead to poor local optima in the case where the path to the best structure was evaluated as relatively poor at an early stage of the data, and was one of the structures discarded at some iteration. If all possible transformations are considered, the algorithm will eventually find the optimum structure, but exponentially many would have to be examined. The particular information criterion guiding the search in our simple implementation is not the complete MI criterion above, but only the dominant term, which counts the total number of arcs in all the FSMs of the set. After all the data is processed, the best structure out of the final hypotheses, using the complete MI criterion, is selected as the estimate.

The algorithm also allows two types of special constraints to be optionally imposed. These are natural constraints on the class of FSM models discussed in the Appendix. The first constraint is to associate numerical values with each arc, and require the loop sum of these values around any path in the FSM to be zero,

analogous to Kirchhoff's voltage law in circuits. This is relevant in the special case where a variable takes a numeric value at each state, and the observations correspond to changes in the variable. The second constraint which can be imposed prohibits FSMs with more than a specified number of arcs of the same label. This limits the maximum complexity of each FSM, and forces paths to loop back through existing arcs rather than create long dangling chains of arcs. Because the first constraint has the opposite effect of preventing loops from forming except under special conditions, the combined effect of these two constraints, happily, is to seriously reduce the set of possible structures. No constraints are used in the examples of this chapter, but several examples are given in the Appendix.

Figures 4.3–5 show the output of this algorithm on three input strings constructed from different pairs of simple cyclic FSMs. The states drawn with doubled circles are the estimates of the current state at the end of the data. In each case, the input string was in fact constructed by simulating the set of structures shown, so the estimates are "correct." Note that in the examples of Figures 4.3 and 4.5, the symbol A is sometimes assigned to the first, and sometimes to the second of the two FSMs. In Figure 4.4, the alphabets of the two FSMs are disjoint, and reconstructing the two FSMs is equivalent to noticing that the pair A and B always alternate in the data, as does the pair C and D. This data string was then modified by changing the D's to A's, to form the data for Figure 4.5.

The program is implemented on an IBM PC in the Scheme language [Texas Instruments, 1985], and requires ten to fifteen minutes to generate each of the estimates shown. This could be much improved with additional programming effort.

BLANK PG.

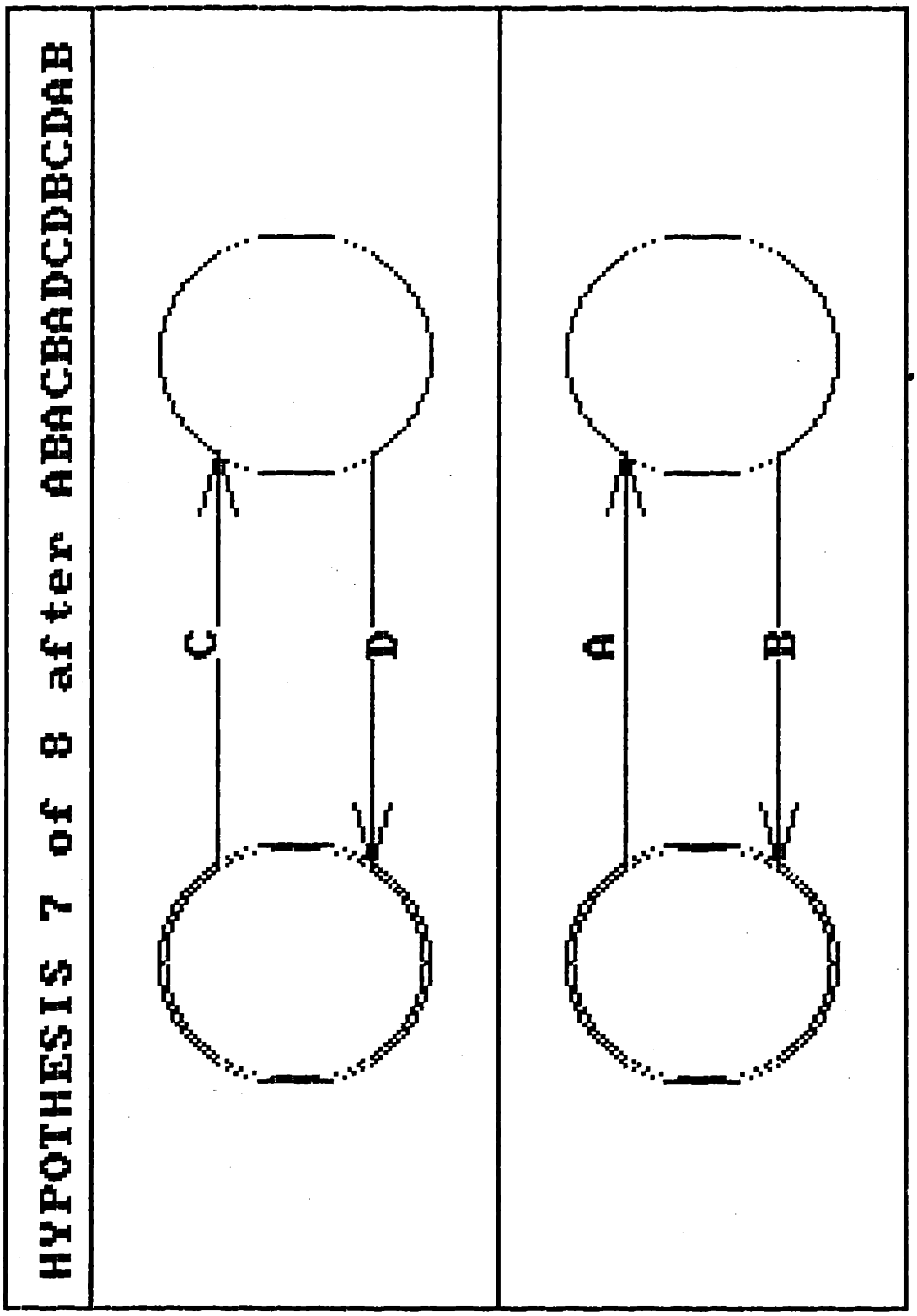


Figure 4.4 MI Estimate of Multiple FSMs for Data *ABACBADCDBCDAB*

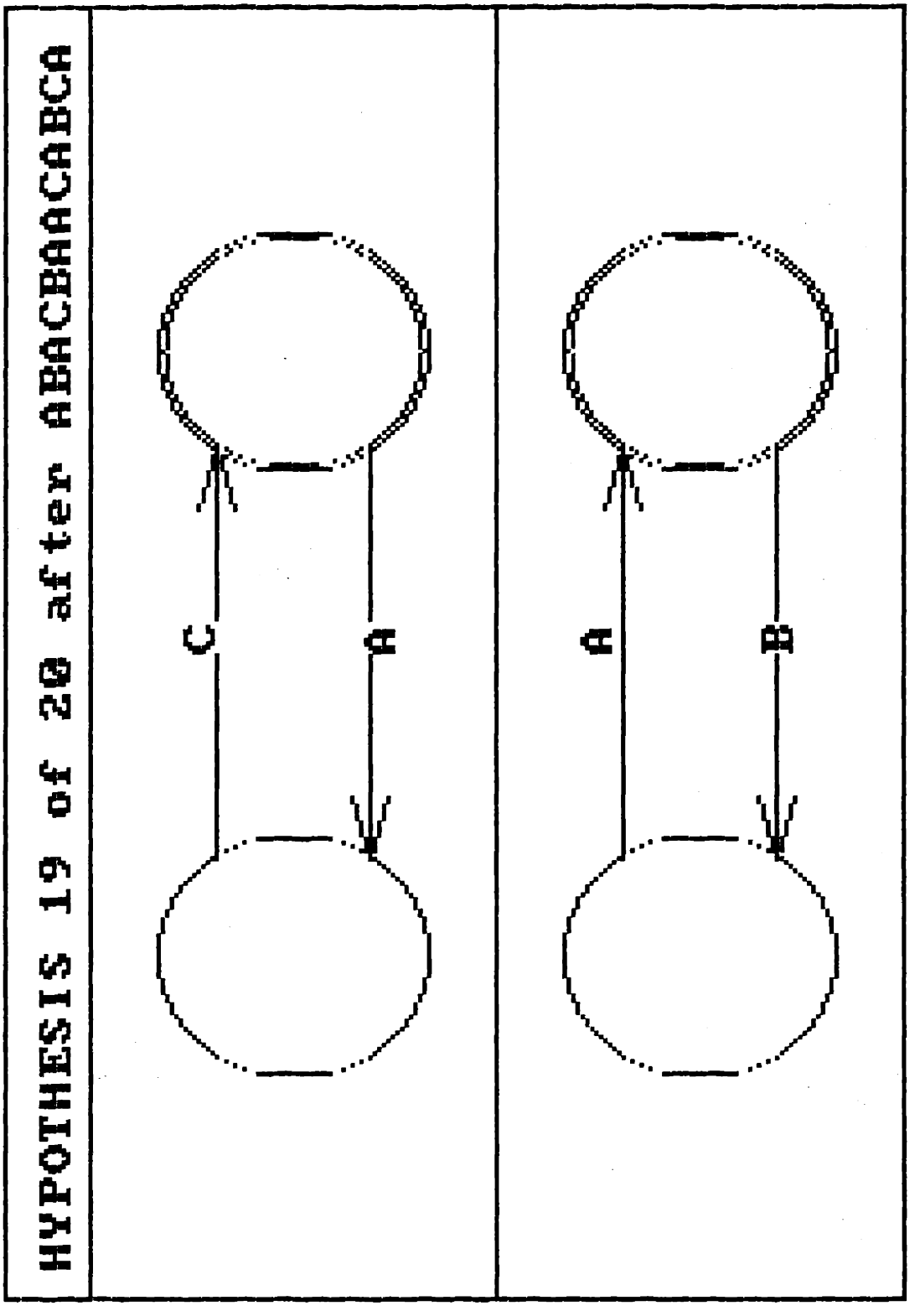


Figure 4.5 MI Estimate of Multiple FSMs for Data *ABACBAACABCA*

Our point in showing these results is *not* to advocate the details of this particular grammar, information measure, or optimization algorithm. However, we do feel that it demonstrates a new approach to FSM inference, which can be feasibly adapted to a wide range of problems. By balancing appropriately between the simplest one-state FSM which generates any string, and very complex FSMs which only generate the data, a reasonable estimate results. It is clear though, that we have barely scratched the surface of this problem.

Chapter 5

CLUSTER ANALYSIS

The following case study develops cluster analysis algorithms based on the minimum information criterion. Cluster analysis is a classical problem in which a "scatter plot" of samples is to be partitioned into groups which classify the samples according to a *data-dependent* criterion rather than a prespecified decision rule. The algorithm is designed to find natural groupings in the data, which in effect implies that it learns decision rules in an unsupervised manner. The simplicity/fit tradeoff arises when the number of classes is not known in advance. Many criteria and techniques exist in the literature which are suitable for clustering applications where the number of groups is prespecified (e.g. Anderberg [1973], Hartigan [1975]). However, if the number of groups is not known, the estimation problem becomes essentially one of structure, because a partition of the input is a set which must be estimated. We only consider this more difficult problem here.

It was for this problem that Wallace and Boulton proposed their minimum information criterion. In two papers [W&B 1968, B&W 1973], they develop information criteria for two specific clustering models, incorporating nonhierarchical and hierarchical cluster structure, respectively. The philosophy tacit behind these

methods can be directly generalized to the structure estimation framework we are using here. The example we give in this chapter differs from those of Wallace and Boulton in that we do not make any explicit probabilistic assumptions corresponding to their assumption of multivariate Gaussian within-cluster probability distributions. In addition, we are not concerned with the problem of discretizing real data; we assume the input has been appropriately quantized.

We will consider here the problem of forming rectangular clusters of a "scatter plot" in a discrete rectangular grid, as in Figure 5.1. In this 50×100 array, indicated by the hatched background, the marked points constitute the input to the estimator, and the rectangles indicate the estimated cluster structure. In this example, there are 29 points in the input, and the estimated structure contains 4 components, including a singleton cluster which contains only an isolated point. The numbers labeled *Bits* at the bottom of the figure indicate the reduction in the information measure as the algorithm makes a sequence of three local transformations, which are described in more detail below.

An informal statement of the problem of cluster analysis is that given a set of observations, the *data set*, described as points within an observation space, we wish to partition the set into groups such that the *natural associations* within groups are large compared to the associations between points of different groups. This reflects the desire to find "natural classes" in the data set, which is usually considered to be a set of "measurements" of individuals of a population mixture. In a vector or metric space, these associations can be quantified, and criteria can be formulated so that intra-group distances are small relative to inter-group distances.

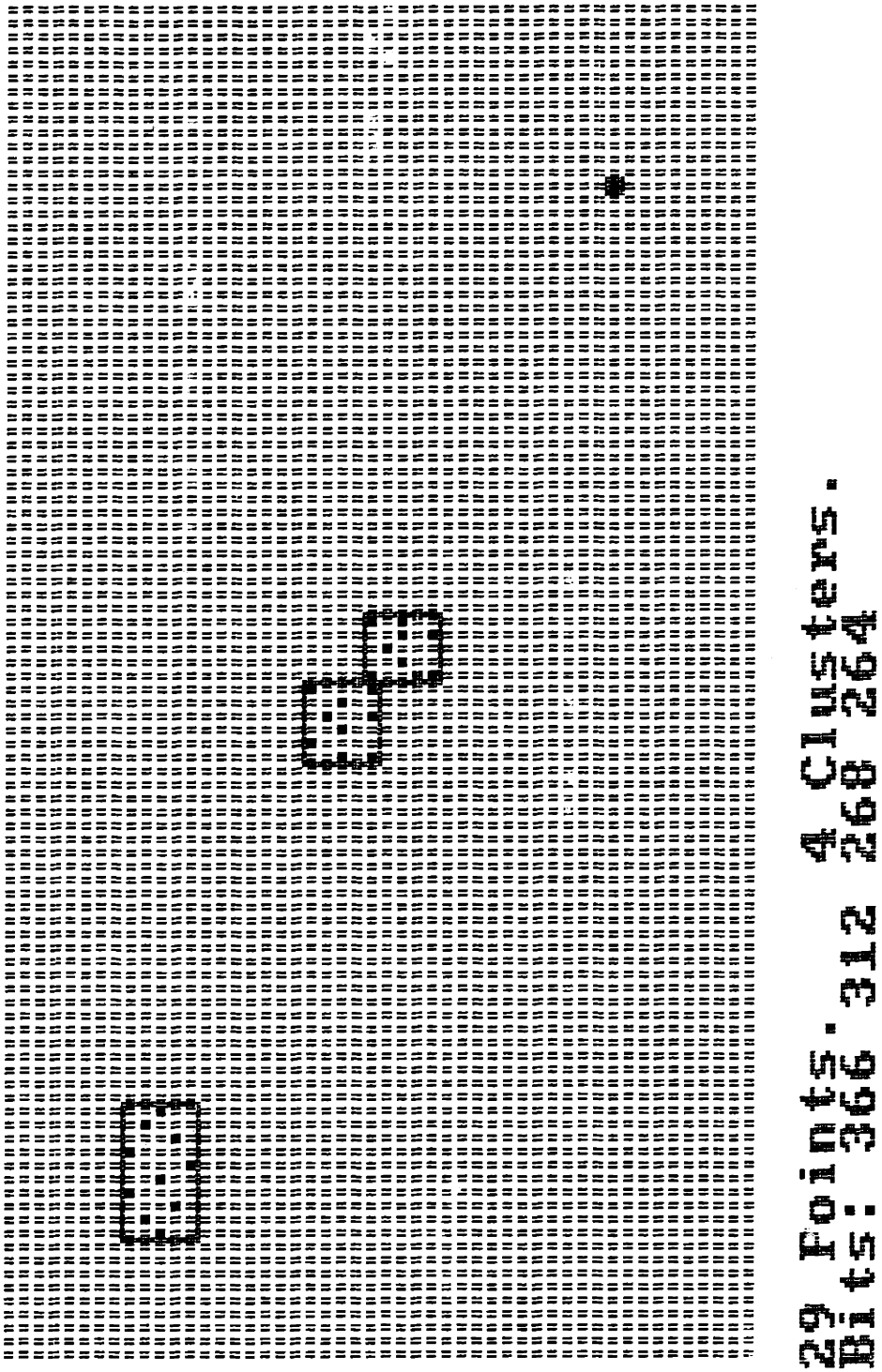


Figure 5.1 Cluster Analysis Example

In one application, Fisher [1936], the observation space is a vector space with components corresponding to the length and width of the petals and sepals of a flower. These measurements are recorded for a large number of individual flowers, and cluster analysis is invoked with the goal of automatically performing a taxonomic classification of the individuals into species. With an appropriately defined cluster analysis procedure, it is hoped that the resulting clusters will correspond to "natural kinds" in the biological domain. Note that the algorithm is not being called on to classify individuals into predefined species. The clustering algorithm produces definitions for the species. It suggests structure for the taxonomic tree.

The above example clarifies that cluster analysis is a particular case of pattern recognition, and shares many of its typical characteristics. In particular, the output of the algorithm can not be plotted as a point in a vector space of fixed dimension, as the number of clusters is not known in advance. In addition, for most pattern-recognition applications, there is no unique correct answer which can be used as an acid test when evaluating the action of various algorithms. Accordingly, we will not be able to say that the methodology below is correct or incorrect in any fundamental sense.

The real problem of cluster analysis is to define what we mean by groups or natural associations. This varies with the application, so we require a flexible framework. We claim that the MI framework provides an adaptable means for tailoring clustering criteria to problem domains in a manner which is be reasonable in a wide range of examples. Furthermore, we support our claims by the fact that the output of a simple implementation agrees well with the clusters which human subjects find in the data.

If natural association can be quantified with a metric in the observation space, the cluster analysis problem may be formulated as an optimization problem over the set of partitions of the data set. The difficulty with this approach is finding a reasonable criterion to optimize. It must evaluate how well a particular partitioning of a data set groups the data. In particular, it must make proper trade offs between inter-group distances, intra-group distances, the number of points in each cluster, and the number of clusters. Note for example that analyses at the two extremes of simplicity and fit are always available. An algorithm might declare that the data set consists of only a single large cluster, or of a large number of singleton clusters. In either case, the criterion that the inter-group distances are large relative to the intra-group distances might be justified as holding trivially.

Few explicit methods for selecting the number of clusters have been proposed in the literature. Duda and Hart [1973] give a hypothesis testing method based on the expected increase in fit when a Gaussian distribution is split into two equal portions along its principle component axis. Hart [1985] gives a similar generalized likelihood ratio test which does not assume the two portions are equal, however, it is tailored towards a specific application. Hartigan [1975] and many other authors rely on "knee of the curve" techniques to choose an appropriate number of clusters, using Euclidean or Mahalanobis measures of fit.

We propose that, for practical applications where the number of clusters is not specified in advance, the MI framework be used to develop and explore clustering criteria and algorithms. Information measures can be constructed as additive measures on "natural" data-description languages designed according to the types of clusters we accept as displaying natural associations in the data. We expect that doing so will lead the system designer to develop pattern recognition systems

that are appropriate for the particular clustering application. The Boulton and Wallace examples, along with the example below support this claim.

The MI approach does not provide a unique criterion for any particular cluster analysis application, but rather, provides a methodology for generating criteria which can be tailored to fit particular problems. The designer of the criterion first creates a language for jointly describing partitions and data sets, which is tailored to the types of clusters that are deemed to be acceptable. A concise way of doing this is for a statement to first describe the cluster structure, and then describe how the data set can be realized from it. The selection of a criterion is then largely reduced to the design of a language. However, many of the details of the language turn out to be unimportant, as only the information content of a statement affects the final criterion.

The resulting function, which identifies the optimal clustering as a function of the data set, is sometimes equivalent to some particular MAP estimator. From the details of its construction, one might extract the equivalent of an a priori distribution for partitions, but no Bayesian or probabilistic assumptions are inherent in the method.

In Section 5.1 below, we formalize a language for describing scatter plots in terms of clusters, and develop an information measure. The example assumes that rectangular clusterings are a natural form of association for the problem domain. Section 5.2 then presents an optimization algorithm which uses local transformations based on the syntax of the language. It approximately minimizes an approximation to the criterion. The algorithm is shown in Section 5.3 to produce excellent results when the data set can be grouped in terms of the rectangular model class. Finally, we discuss a probabilistic interpretation of the method, some of its weaknesses, and methods of extending or improving it in Section 5.4.

5.1 Description Language and Information Measure for Clusters

Probabilistic approaches are generally employed in clustering without enormous justification. Although it is often reasonable to model measurements of individual characteristics of members of a population as a sample from a probability distribution, the particular forms of the individual and joint distributions are usually selected arbitrarily, or for numerical or analytical convenience. We will need to make analogous assumptions concerning appropriate methods for describing measurements.

In the example given here, these assumptions are in the rectangular form being used to describe clusters. In data sets such as Figure 5.1, this class of models suffices to define natural groupings in the input. Examples are given in Section 5.3 which show the results of the method when this class of models is not appropriate. From a probabilistic point of view, these correspond to the assumption that within a class, different measurements are independent and uniformly distributed (with unknown mean and variance). Note that we are choosing a criterion based on the group as a whole, rather than the usual pairwise distance relations between points.

The input to the estimator consists of an N_X by N_Y array of pixels, each of which is either "ON" or "OFF". We wish to find a model of the data which describes the position of the ON pixels in terms of a set of N_C "natural classes", each of which has a rectangular form with a high density of "ON" pixels. Our model requires every "ON" pixel be assigned to some class. We phrase our formulation in the terms of a vision problem, with an array of pixels, in order to emphasize its status as a pattern recognition problem. It could be rephrased so that the input is a set of observations, the (x, y) pairs of the ON pixels. The main difference is that

this latter form allows observations to be repeated. The languages and algorithm below allow this without modification.

Because the data must be partitioned completely into clusters, a natural description will be composed of descriptions of the elements of the partition. Each of the cluster descriptions consists of its bounds within the grid. To this we attach its portion of the realization, a list of the points within the cluster.

```
      S  →  Cluster*
Cluster  →  Bounds Point*
 Bounds  →  X-value X-value Y-value Y-value
X-value  →  1 | 2 | ··· | N_X
Y-value  →  1 | 2 | ··· | N_Y
  Point  →  1 | 2 | ··· | < Area of Rectangle >
```

In describing a point within a cluster, we note that there are only as many possibilities as there are points in the rectangle, which we term the "area" of the rectangle. Therefore a point can be described by its ordinal position in an alphabetic enumeration of the pixels which fall within the cluster bounds. It will consequently take fewer bits to describe a general point within a small rectangle than within a large one. This is desirable as it encourages rectangles which fit snugly around the data. Two alternatives to this method of describing points are considered in sections 5.3 and 5.4 below.

Formally, a listing of ordinal values requires a syntactic constraint that each `Point` should not exceed the area of the rectangle described by the preceding `Bounds`. This type of dependency may be handled in many ways, e.g. a context-sensitive grammar. For our purposes, it is easiest to allow sentences which violate this constraint to remain in the language, but have the algorithm ignore them by assigning such a `Point` no point as its interpretation. As such a term adds formal information without changing the interpretation of the sentence, these sentences

are ignored by the estimator. The matter is irrelevant to the optimization below, as the design of the algorithm ensures that all sentences considered satisfy the constraint.

An additional constraint occurs within the expansion of Bounds. If the four components are interpreted as the left, right, top and bottom extremes of the rectangle, then our syntax allows boxes with a left side to the right of the right side, and/or a top below the bottom. These possibilities can be eliminated either syntactically or semantically, as above. We shall bypass this decision by explicitly defining the information measure for a Bounds term, rather than recursing down to its components. A rectangle is defined by a lower-left corner and an upper-right corner, both selected from among the pixels in the grid, and for each pair of pixels only one of the two orderings is consistent with our interpretation. The combinatorial notion of information gives

$$I(\texttt{Bounds}) = I_R = 2\log_2(N_X N_Y) - 1$$

where we use the notation I_R as mnemonic for the information to describe a rectangle. The information in Point* in the i^{th} cluster, using the I^* measure for the star, and a combinatorial measure for the positions, is

$$I(\texttt{Point}^*) = I^*(n_i) + n_i \log A_i$$

where n_i is the number of points, and A_i is the area of the i^{th} cluster. Adding these terms for the N_C clusters, and incorporating a I^* term for the Cluster* term gives

$$I(S) = I^*(N_C) + N_C I_R + \sum_{i=1}^{N_C} [I^*(n_i) + n_i \log_2 A_i]$$

which is to be minimized over the set of all statements which can describe the input.

5.2 Optimization Method

In principle, $I(S)$ can be minimized by evaluating each possible partition of the input and choosing the optimum. In practice this is not feasible, as there are $O(N^N)$ partitions of a set of cardinality N. Because of these combinatorial difficulties, we settle for a local minimum with respect to certain neighborhood transformations, instead of the global minimum.

Our algorithm begins with an initial description of the data set in terms of the simplest possible model: a single cluster. This initial cluster is chosen as the smallest rectangle in which all the given points belong. Because of our constraint that every point must be inside some rectangle, and because the information grows with the area of the rectangles, this will be the optimum rectangle if the MI estimate requires only a single cluster.

The algorithm proceeds by local steps which replace a `Cluster` clause with two clauses, thereby increasing the number of clusters by one. As the algorithm proceeds, only the statement with the smallest information measure is retained. The allowed transformations of a statement have the effect of splitting (partitioning) one of its clusters into two smaller clusters by considering all possible ways of reassigning the points to two smaller clusters.

These transformations are considered one cluster at a time, and for each cluster, either no split is found which reduces the information measure, or if one or more transformations reduce the measure, the best such split is adopted. The process terminates when no cluster can be split into two in a way that reduces $I(S)$. Conceptually, the search is isomorphic to a gradient descent through the partition-lattice of the input set, selecting the branch with steepest slope at each tier. However, only refinements of the partition are allowed—merges have not

been implemented—so the "neighborhood relations" are not actually symmetric in this space. This does not seem to be a problem in practice however; we have not been able to construct an example in which the algorithm reaches a local minimum which can be exited via a merge transformation. If we did find such cases, it would be simple to extend the algorithm to include merge transformations.

In our implementation of the algorithm, a pruning technique based on the geometry of the points within the rectangle can be employed to significantly reduce the number of partitions which need be considered for splitting each cluster. It does not affect the outcome of the algorithm as it guarantees the optimal two-way split is not pruned. If all ways of dividing a set with n members into two nonempty subsets were considered, $2^{n-1}-1$ possibilities would have to be enumerated, and the algorithm would require exponential time. However, most of these subsets correspond to two clusters which overlap significantly, and would therefore increase rather than decrease the information measure.

We can eliminate most of these superfluous partitions if we use the fact that in an optimal split, each side of each rectangle will have at least one pixel ON just inside each of the four borders. (This is because otherwise the border could be translated inward one unit to reduce the area of the rectangle without affecting cluster membership. As this would reduce the information measure without affecting the interpretation, it can not be possible in an optimal split.) We can therefore arrange four nested loops to loop through just those indices which satisfy this constraint, and create all possibilities for the first rectangle in $O(n^4)$ time. For each possibility of the first rectangle, the second rectangle is determined directly from the remaining points. They can be gathered and boxed in with the smallest

rectangle which contains them in $O(n)$ operations. Thus one split requires $O(n^5)$ operations, and the complete analysis of the input requires $O(n^6)$ operations.

While this is not as low an order polynomial as one might want, the leading coefficients are apparently low enough that the algorithm operates reasonably for data sets under 40 points. The complete analysis, coded in Turbo Pascal [Borland 1984], requires up to five minutes on an IBM PC. For larger data sets, a number of heuristics could be employed to reduce the search time significantly. This would also be essential in higher dimensions, as the time required for a direct extension of this algorithm would increase by a factor of $O(N^2)$ for each dimension.

To evaluate the information savings of a putative split, we need only examine the portion of the data set within the cluster under consideration. Letting S_k be the description of the data in terms of k clusters before a possible split, and S_{k+1} be the description of the data if the split were accepted, the algorithm splits iff $I(S_{k+1}) < I(S_k)$. The terms in this expression concerning the other clusters all cancel. Our implementation ignores the small $\log_2 \frac{k+1}{k}$ term, which results from the two I^* terms that almost completely cancel, and we elsewhere approximate $I^*(x)$ as $\log_2(cx)$. The test to determine whether or not the two smaller clusters are preferable to a single cluster then becomes

$$\text{Split} \quad \text{iff} \qquad \log_2 \frac{A_1^{n_1} A_2^{n_2}}{A_c^{n_c}} + \log_2 \frac{c n_1 n_2}{n_c} < -I_R$$

Here n_1, n_2 and n_c are the number of points in the two subclusters and the original combined cluster respectively, (so $n_c = n_1 + n_2$), and the correspondingly subscripted A terms are the respective areas.

5.3 Results

The implementation of the algorithm is quite successful at finding rectangular clusters. It can separate nearby clusters, find overlapping clusters, and even find high density clusters completely embedded within other (low density) clusters. This type of performance is typical for the algorithm. Figure 5.2 shows another clustering result, somewhat more complex than the example of Figure 5.1.

The sequence of numbers labeled *Bits* at the bottoms of the figures indicate the decrease in $I(S)$ as the partition is refined. The first value indicates what $I(S)$ would be if only a single cluster were used to describe the points. The final value is $I(S)$ for the partition shown. (The sequence of intermediate partitions corresponding to the intermediate values can not be reconstructed from the figures.)

Note that when a pixel appears inside of the geometric domain of more than one cluster, the algorithm above automatically assigns its membership to the subsuming cluster which has the smallest area. The pixels in the common area thereby contribute the least to the overall information measure.

The results in Figures 5.1 and 5.2 agree exactly with most observer's subjective clusterings, with one possible exception concerning the "background noise" which is discussed below. This agreement is typical of situations in which the subjective groupings are reasonably rectangular, with aspect ratios not too far from unity. However, the next two figures show that in other situations, the estimated structures have a peculiar quality.

When the perceived clusters are poorly described as rectangles, the algorithm fares less well. In Figure 5.3 we see the output of the algorithm when a long diagonal line of n pixels is presented as input. Here, the algorithm tries to describe this nonrectangular group using its rectangular vocabulary, with somewhat surprising

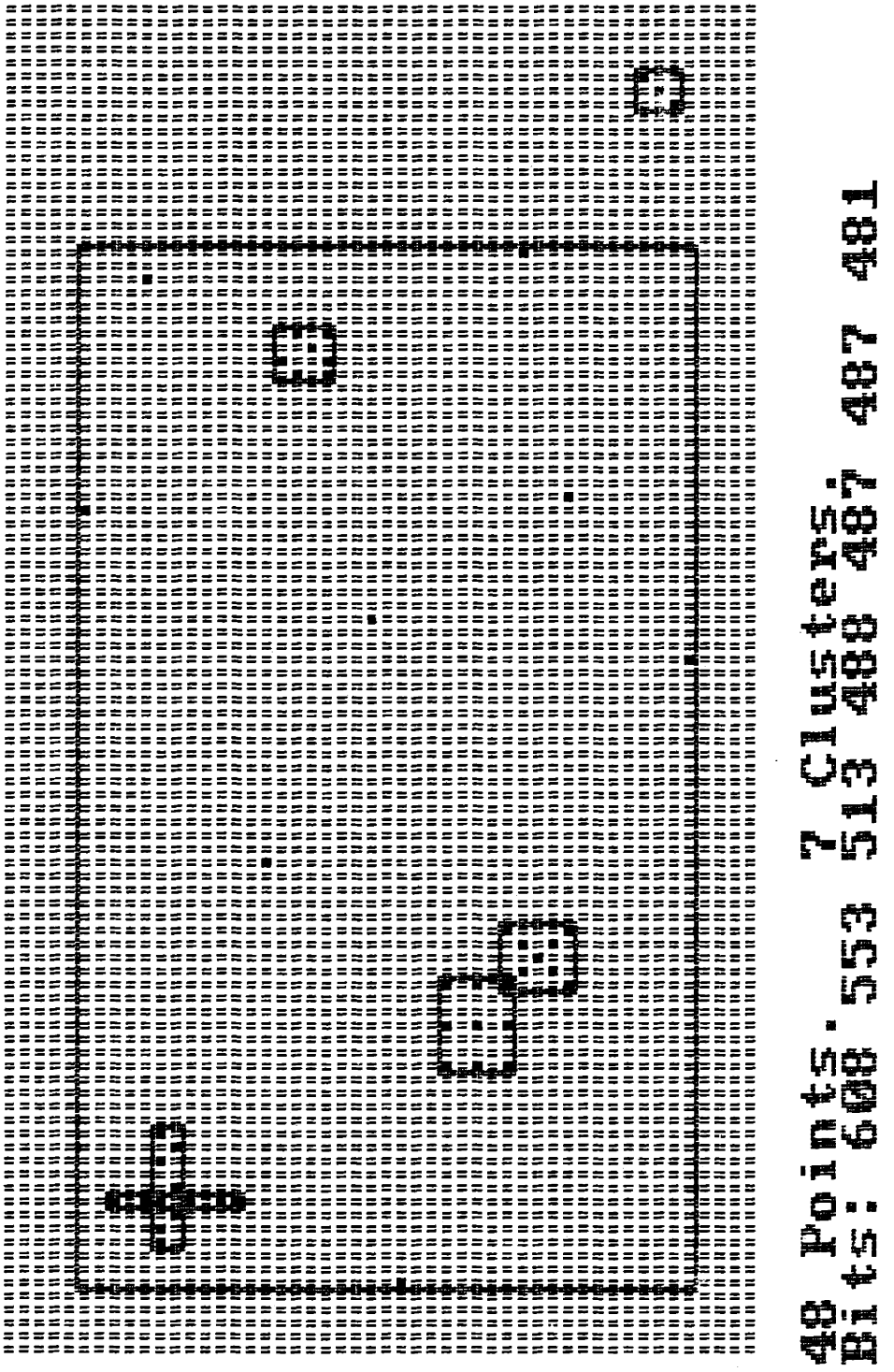


Figure 5.2 Second Clustering Example

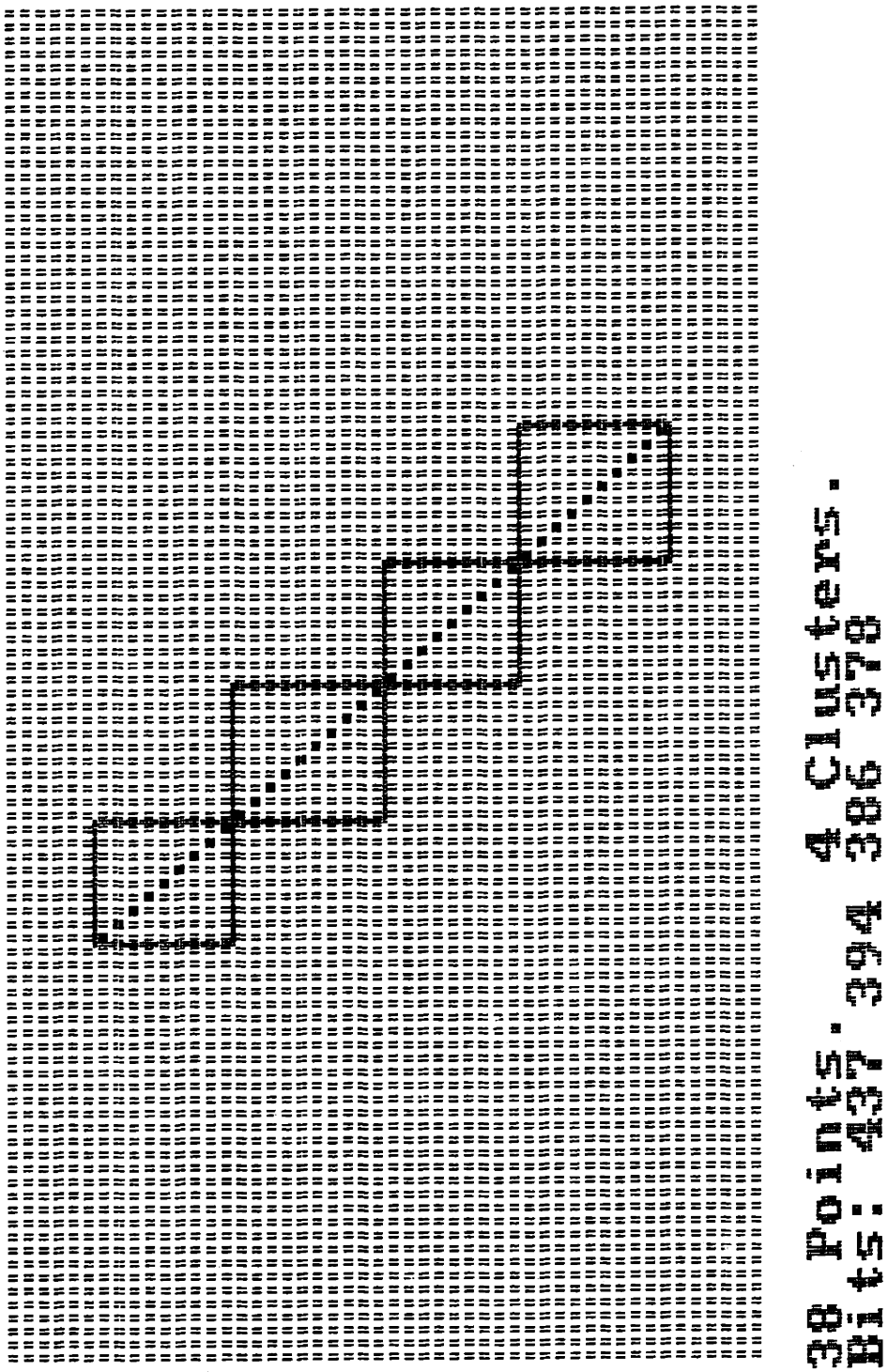


Figure 5.3 Poor Fit Between Rectangular Clusters and Diagonal Groupings

results. As one large cluster would be wasteful of area, and n unit rectangles is wasteful of complexity, the estimated structure contains a sequence of rectangles of intermediate size.

The optimal size is easily determined analytically in a simple example. Consider the case of a square $N \times N$ input array consisting of a line of N pixels ON along a main diagonal. From geometric considerations, the MI estimate of its structure using the models above will consist of K subsquares exactly covering the diagonal in the manner of Figure 5.3, with $1 \leq K \leq N$. By symmetry arguments, we expect the K squares to be as nearly equal in size as the integer constraints allow. Letting S_K be a sentence that describes the input in terms of the K groups, we wish to determine $\frac{N}{K}$, the size of each group. We do this by specifying the information measure above to this example as

$$I(S_K) = K[I_R + \frac{N}{K} 2 \log_2 \frac{N}{K}]$$

and differentiating with respect to K to get

$$\frac{\partial I(S_K)}{\partial K} = 4 \log_2 N - 1 - \frac{2N}{K \ln 2}$$

From this we determine the preferred size of each cluster:

$$\frac{N}{K} = \frac{2}{\ln 2} \log_2 N - \frac{1}{2 \ln 2}$$

It is curious that the preferred cluster size is logarithmic in the grid size. This is one of several scaling problems discussed in Section 5.4.

A second property of this clustering algorithm that may not agree with intuition is demonstrated in Figure 5.4. Here we see that the criterion is quite content to generate long narrow clusters which are not geometrically compact. They are

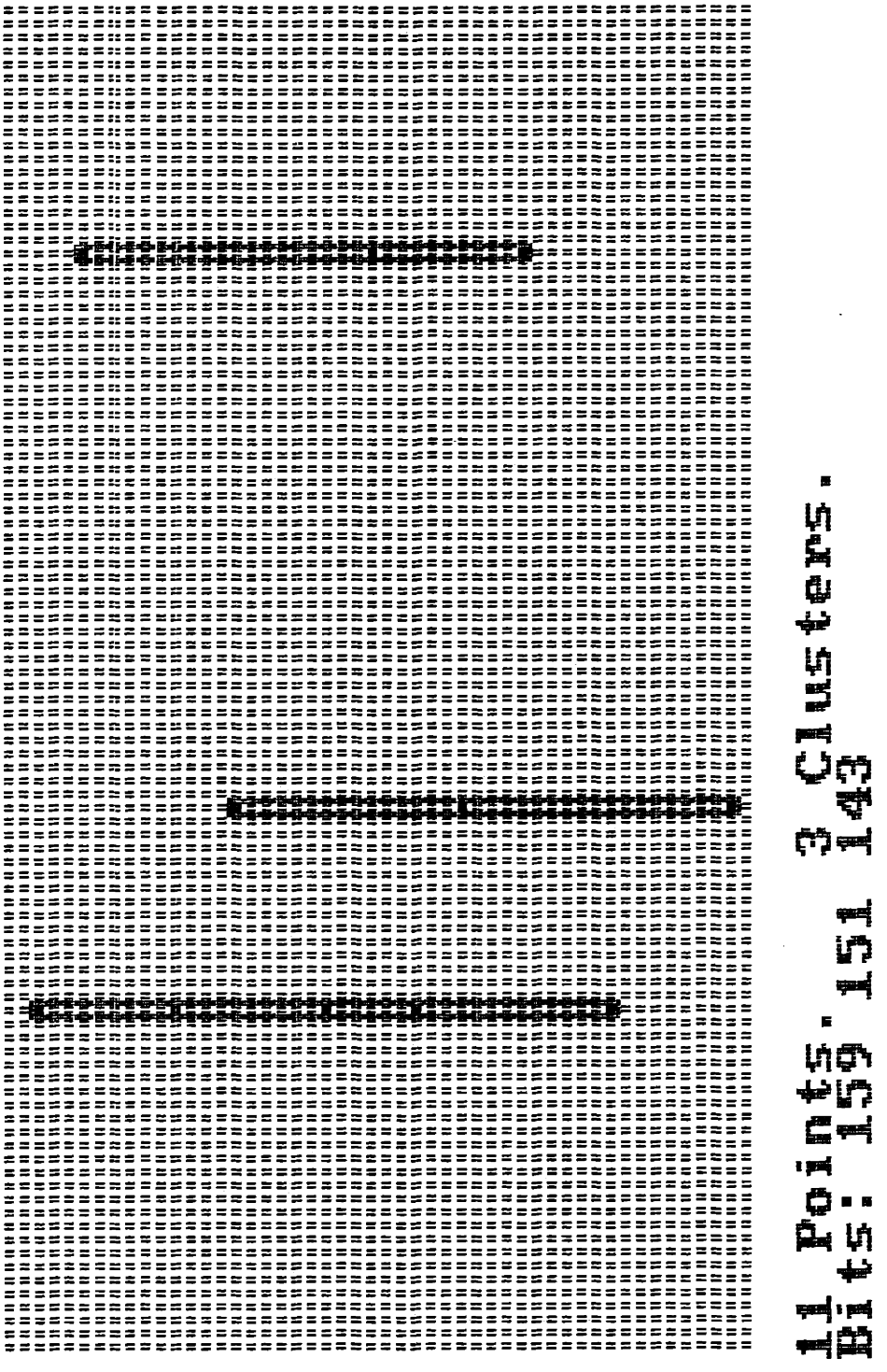


Figure 5.4 Long Rectangular Clusters

quite small in terms of the area measure we employ however, and in hindsight are to be expected. If we do not wish the estimator to produce such results, there are several ways to proceed. One way to accomplish this is with an outright constraint in the grammar which simply prohibits clusters beyond a certain aspect ratio. Another is to describe rectangles, not by their bounds, but by their centers, and major and minor "radii". An entropic information measure on the radii could be designed which is biased against large values, or large differences between the two values.

A final aspect of the results, which may be disturbing, concerns the large diffuse clusters of "background noise" such as that visible in Figure 5.2. It is not clear what the "intuitively correct" clustering should be in this situation. The constraint requires every point to be in a cluster, but a sparse set of points does not appear to form a cluster. Are these points naturally associated? The alternative of putting the individual "noise" pixels each into their own 1×1 rectangle certainly is appealing—this is how the singleton pixel in Figure 5.1 is described. The effect of the MI criterion is to use a diffuse cluster to describe pixels in a region of relatively low density, as in Figure 5.2.

It is interesting to see how a simple modification of the description language can be used to eliminate these diffuse "clusters" if we deem them unacceptable. The language above requires a description of each ON pixel associated with each rectangle. If we want or expect relatively dense clusters to result from our estimator, it seems more natural to describe the OFF pixels. After all, one might argue, they are the unusual situation in a dense cluster, and so it is their information we should measure. Implementing this involves only a trivial modification to the optimization algorithm. As we still require each ON pixel to be in some rectangle,

only the cost function for the pixels within a putative rectangle changes. The remainder of the algorithm is unchanged. Implementing this gives mixed results, shown in Figure 5.5. Dense clusters are now preferred, and the diffuse region of Figure 5.2 is completely divided into the singleton rectangles. The denser clusters remain unchanged, but other sparse clusters have also been divided into smaller regions.

5.4 Discussion

We have selected one particular notion of natural association, and explored how it can be formalized and proceduralized into a cluster analysis algorithm. It bears emphasizing that this technique can be directly adapted to many other clustering criteria. The technique would be geometrically more complex, but conceptually unchanged, if cluster regions were different shapes, e.g., circles or ellipses, or were constrained to certain minimum or maximum sizes or aspect ratios. Higher dimensional variants are straightforward extensions, and entropic information measures can be employed when relevant. The methods of Wallace and Boulton are special cases of these variants.

Although we have only considered the problem in which the number of clusters is unknown, there are many ways structural issues may arise even when the number of clusters is known. There may be structural uncertainties to be estimated at finer levels (e.g., the structure of the points within each cluster) or coarser levels (e.g., the structure of the relation between the clusters). For example, even if one specifies the number of groups, an estimator can choose if each should be rectangular or some other form.

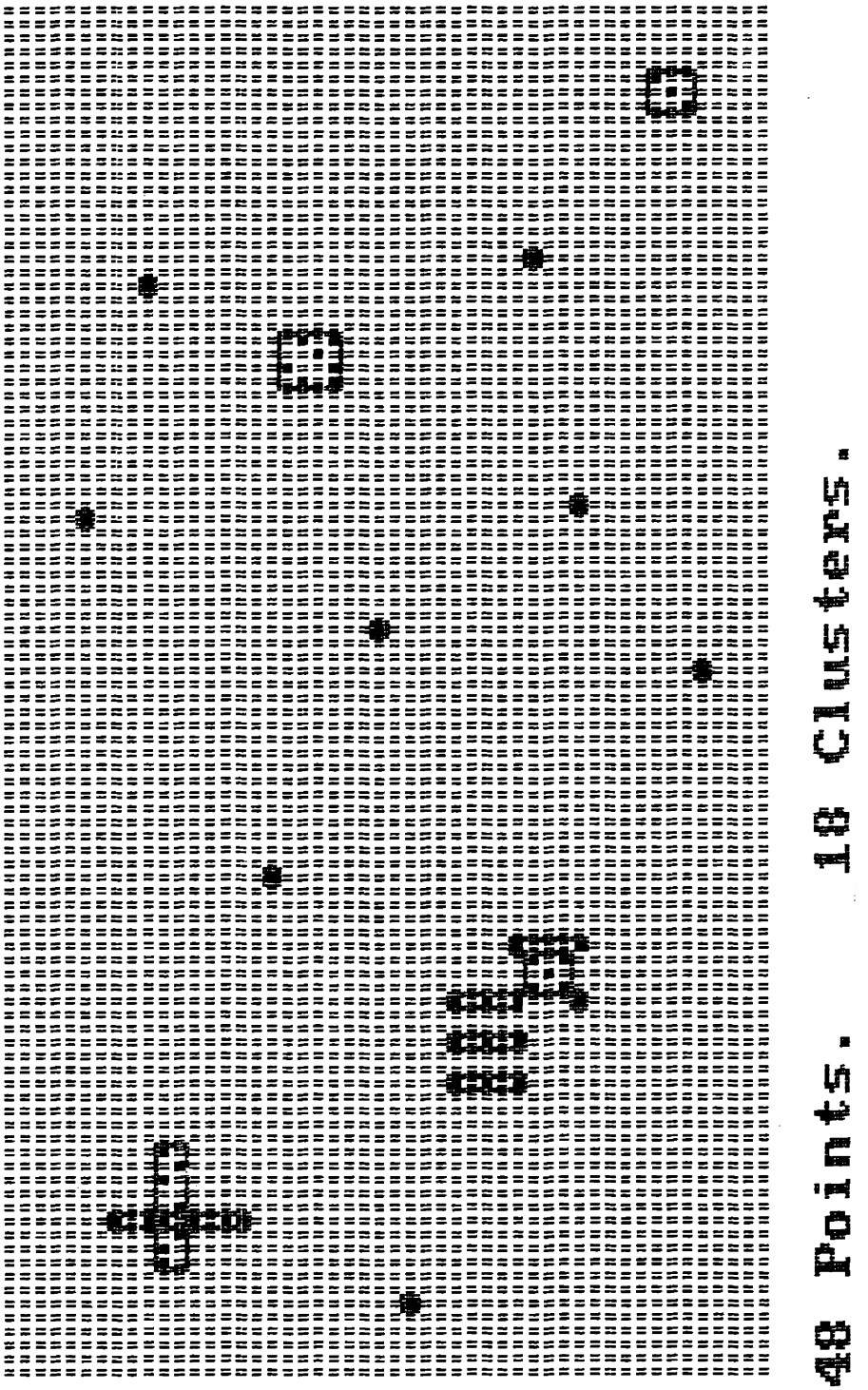


Figure 5.5 Clustering Resulting from Describing Pixels OFF, Rather than Pixels ON (c.f. Fig. 5.2).

Hierarchical relations between the groups are often of interest. The clusters produced by our algorithm can be viewed as the leaves of a binary tree formed by the sequence of splitting transformations. We have not shown the tree, but it is defined by the intermediate descriptions during the operation of the algorithm. If one were interested in the entire tree rather than just its leaves, one would design a description language for trees (e.g., the *S expressions* of LISP), and allow "subclusters" to be described concisely by making use of the description of the "parent" cluster. An approach to hierarchical classification of this form is developed in Boulton and Wallace [1975], who mention its application towards decision trees. Quinlan and Rivest [1987] present another example in which the structures of decision trees are estimated using an MI approach. It can be interpreted as a form of clustering.

One aspect of this cluster analysis method for which we have no entirely satisfactory solution is the issue of scale invariance, or rather, lack thereof. If the scale of the observation space is changed, the MI estimate of the structure may change. We might change the scale of the observation space in two ways: by keeping the distances between the ON pixels constant, or expanding them in proportion to the space. (We ignore contraction scalings as they introduce certain distractions when two ON pixels converge upon the same coordinates.)

In the first case, the data set is fixed, but we expand the linear dimensions of the space by a factor E, so that the grammar allows

$$\texttt{X-value} \rightarrow 1 \mid 2 \mid \cdots \mid EN_X$$
$$\texttt{Y-value} \rightarrow 1 \mid 2 \mid \cdots \mid EN_Y$$

The information in the description of the data set given the rectangles remains unchanged for any given clustering, because this depends only on the areas of

the rectangles. However, I_R increases by $4 \log_2 E$, as there are more rectangles in the grid. As the same partition is now measured to have a greater complexity, a coarser partition may result. In terms of the decision rule in Section 5.2, the threshold for splitting, on the right side of the inequality, becomes more negative, and only a subset of the splits previously allowed are still accepted. For large enough E, the data set will be analyzed as a single group.

This strikes us as reasonable. It is analogous to standing at a distance from an image and being uninterested in details which are apparent upon closer inspection. It also emphasizes the hierarchical nature of the results. What appears at a distance to be a splotch reveals itself as an intricate set of structures when we focus our attention on it. However, these properties are not entirely satisfactory. One could argue that the question of whether or not clusters are present in a small region should be independent of the size of the universe.

The second form of scaling, in which we scale the coordinates of the ON points in proportion to the space, results in a similar effect. As points recede from each other in this expanding universe, the algorithm is again less willing to split, for similar reasons. I_R again increases by $4 \log_2 E$, but now the areas of the rectangles increase by a factor of E^2. However, the effects of this increase on the parent and daughter rectangles cancel, and the decision rule is the same whether we scale the points with the space, or keep them fixed. For large enough E, the MI criterion gives a single diffuse cluster.

The "correct" response to scaling in this case is less clear. From one point of view, we would like more, smaller, clusters. This expansion increases the distance between pairs of points, and therefore reduces their association. Groups which were previously clustered should then be separate. (This property can be achieved

if we describe the OFF points rather than the ON points, and speaks well for that method.) From another point of view, we would like the clustering to remain unchanged, as the proportional distances within the space are unchanged. We have not been able to work out a description and information technique with this property.

Note that, in contrast to scaling, the criterion (and the algorithm) has the correct invariance properties with respect to translations, rotations, and reflections that map the data into other data sets within the same grid. If the data is rotated 90° or 180°, is reflected through a horizontal or vertical line, and/or is translated, the resulting clusters are transformed accordingly.

We have presented our problem in a form much like a vision problem, in which each pixel is either ON or OFF. To see how easily this formulation is changed into a plausible vision problem, consider a language in which we define "clusters" to be linear arrangements of points, which we interpret as a line of pixels in the grid, possibly diagonal. Such an arrangement can be described with a clause conjoining descriptions of the two endpoints. By this means, the problem is transformed into a plausible line detection problem, suitable for finding dark lines on light backgrounds in somewhat idealized situations. A more realistic vision problem is carried out in Chapter 7.

The ON/OFF formulation for the input is really designed to find the range of variation of a group, and is not concerned with density changes within this range. This will eventually lead to unsatisfactory results if the data is generated as samples from probability distributions with unbounded extent. Asymptotically, each cluster would fill the entire input array, and no discrimination could result. We are, in effect, assuming that only a fixed amount of data is available (as in a vision

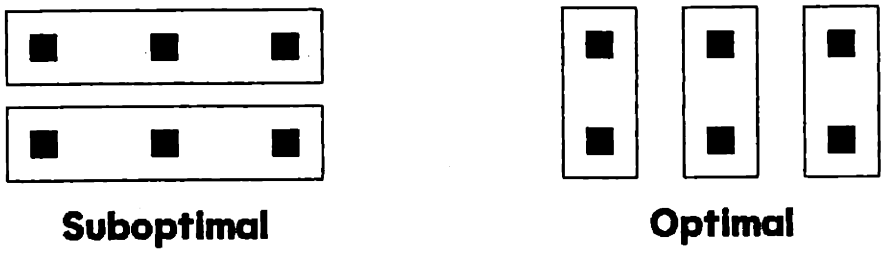


Figure 5.6 Suboptimal Partition due to Binary Splitting Transformation, and Optimal Partition which would Result if Ternary Splits were Allowed.

problem), and would use other languages were the situation otherwise. The main difference is that if the application allowed each point to be observed a variable number of times, we would design the language accordingly, and not require the point coordinates to be repeated for each observation. A more appropriate format is to describe the coordinates only once, followed by a count of the number of observations.

A number of questions concerning the optimization technique presented above must be addressed, beginning with the set of transformations used. It is easy to construct situations in which the algorithm becomes trapped in a local optimum. Figure 5.6 shows a simple situation in which an initial binary division leads to a suboptimal result compared to the global optimum. The global optimum is reached if an initial ternary split is permitted. One might want to augment the two-way splitting transformations with multiple-way splits and various types of merges, to eliminate certain local pitfalls. There are also many possible extensions into readjustment transformations which combine splits and merges. We justify our simple choice by its performance.

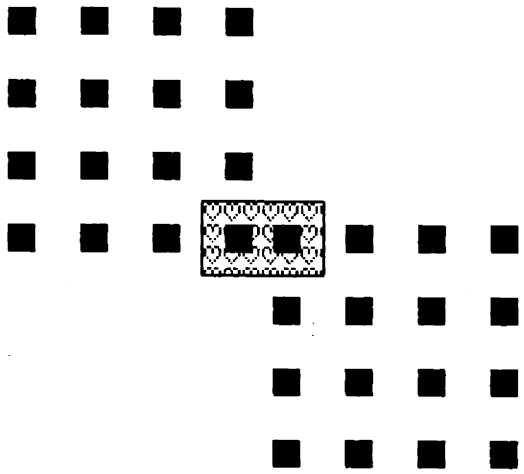


Figure 5.7 Erroneous Association which would Result if Optimization Relied on Agglutinative Transformations, Using Finest Partition as Initial State.

An alternative optimization method, which also requires only a single transformation, would be to start with the finest partition—singleton clusters—and merge clusters into a larger cluster whenever the information measure is reduced. If this could be made to work, it would be a computational boon, as all possible merges can be considered in $O(N^2)$ time, compared to the $O(N^5)$ time explained above for splits. This agglutinative approach has a great potential for error however, as it is easily misled by local properties of the data set. For example, Figure 5.7 shows a situation in which elements of two distinct rectangular clusters would be erroneously merged because outlying points of the two clusters happen to lie close to each other. The local transformation is oblivious to structure which is determined by the global context. Because configurations of this nature are not uncommon, we selected the top-down, divisive approach which takes advantage of large-scale statistical averaging. Note that Figure 5.1 includes two clusters related as in Figure 5.7, and demonstrates that the algorithm performs properly in this respect.

The fact that this algorithm easily resolves these adjacent clusters, which causes problems for many other algorithms, is noteworthy. This configuration is included in a list of problematic configurations which confuse many clustering algorithms, in Patrick [1972, p. 358]. The crossed clusters, overlapping clusters, and embedded clusters of Figure 5.2 are also included in this list. We therefore endorse the MI approach to clustering as outperforming many other approaches.

However, the time required by this particular implementation, five minutes for 40 points, combined with its growth rate, $O(N^6)$, condemn it to laboratory applications only. For many "field applications" where cluster analysis is employed, rapid analysis of large data sets is required. This time results from the fairly exhaustive set of transformations considered. Many heuristics seem plausible for accelerating the optimization by reducing the set of splits considered for each cluster. We will not consider them here however, as we are more interested in demonstrating the general utility of the MI criterion than in the fine details of this particular example. As an alternative approach, we suggest that a multigrid optimization method, roughly along the lines of Section 7.2.1, seems quite plausible here, although we have not worked out its details.

A final note about the optimization of this criterion is that it would be interesting to compare the local optima of Figures 5.1 and 5.2 with the global optima, if they differ. After all, one might argue, it is possible that the MI criterion would give very poor results if actually optimized, and what we see and approve results from failings in the optimization! As we have no general way to find the global optimum in such a large space, we must learn to live with this possibility.

Our final set of comments concern the relation between this method and probabilistic approaches. The immediate Bayesian reaction to the MI criterion is

that it is isomorphic to some MAP estimator. The information in the description of the Bounds terms increases linearly with the number of rectangles, independent of their position. The estimator is therefore isomorphic to some MAP estimator with an *a priori* distribution for rectangles which is exponential in their number and uniform in their placement. The two ways for describing the pixels within a cluster, listing the ON pixels or the OFF ones, then correspond to two different conditional probability distributions, $P(z|\theta)$.

A third method of describing the pixels, and an associated information measure, are suggested by this probabilistic point of view. Rather than list the positions of points of only one type, we could describe every point in the rectangle, with a sequence of 1's and 0's.

$$\texttt{Points} \rightarrow \{0|1\}^{\texttt{Area}}$$

From a coding point of view, this results in a more compact representation for all but the smallest rectangles. The information in the sequence would then be measured as proportional to the length of the sequence, which is the area of the rectangle. The constant of proportionality would be the sample entropy if an entropic measure is used. We are uncomfortable with this entropic measure however, as it favors highly sparse rectangles over slightly sparse ones, which is not a desirable property in clusters.

One further probabilistic comparison is of interest, concerning the decision rule at the end of Section 5.2. Curiously, this rule has a very similar form to a likelihood-test rule derived using the assumption of a mixture of two Gaussian distributions. For this model, one [Hart 1985] can derive the rule

$$\text{Split} \quad \text{iff} \quad n_1 \log|\Sigma_1| + n_2 \log|\Sigma_2| - n_c \log|\Sigma_c| < \gamma$$

where γ is a constant threshold, and the $|\Sigma|$s are the determinants of the covariance matrices of the parent and two daughter distributions. The analogy arises from the fact that the determinate of the covariance is a measure of the *area* of the portion of the distribution within a constant-probability ellipse. There is a difference however in that our decision rule also has second-order terms resulting from the $\log^*$ terms in $I(S)$, which measures the increase in structure. This is analogous to Rissanen's [1978] addition of second order terms to Akaike's [1974] information criterion.

Chapter 6

WAVEFORM SEGMENTATION

This chapter considers an apparently simple, yet subtly difficult problem of waveform segmentation, from the field of signal processing. The essence of the problem exists in a variety of applications, but its structural characteristics are not generally pointed out. The general problem is to partition a time period into contiguous segments which are modelled as uniform with respect to some given properties. The input data is a sequence of observations, $z(t)$, as a function of quantized time, and the output of the estimator is a partition of the time axis into contiguous segments, along with a description of the properties of each segment. Traditional spectral approaches to signal analysis are of little help, as they do not approach the fundamental issue of complexity.

The most difficult part of this problem may be realizing that it is an ill-posed structural estimation problem. The tradeoff between simplicity and good fit comes about because at the two extremes, the analysis could be that there is only a single long segment in the partition, or that there are as many segments as observation times. The simplest structure, that of a single segment, will generally have the poorest fit to the data, as all of the variation in the data must be described as

within-segment variation. The most complex structure, in which each observation constitutes a separate segment, generally has the best fit to the data, as there is no remaining variation to describe if the one observation can simply be described as a constant. An MI estimator can provide the additional structure necessary to balance these two extremes, so as to derive an appropriate degree of structural complexity relative to the class of models and the input data.

The case study presented here uses the simplest possible within-segment model: each segment is constant at some z value, and z is a scalar quantity. The example also assumes there is no between-segment "dynamics"—once a segmentation is chosen, the models within each segment may be selected independently of the neighboring segments. It would not be difficult to modify the method however, to allow for more complex models in each segment, such as vectors in which each component is a polynomial of fixed degree. One method for doing this is discussed in Section 6.4. Inter-segment constraints, such as the continuity of derivatives commonly implemented with cubic splines, could be incorporated with more difficulty.

More interesting segmentation problems would involve more complex within-segment functions, such as models appropriate for phonemes in speech processing, instrumental notes in music analysis, or electrocardiograms and other medical waveforms. These extensions are discussed in Chapter 8. A brief survey of other waveform segmentation techniques can be found in Pavlidis [1977].

6.1 Languages and Information for Segmented Models

The details of this problem come from the application described in the appendix. We are given a discretized sampled input waveform which is well-modelled

as piecewise constant, with additive Gaussian noise of unknown variance. We have no probabilistic models concerning the number, duration, or levels of the constant segments. The estimator must determine the boundary points between these intervals, the constant value during each period, and the variance of the noise.

In order to use a description-based method on inputs which are to be analyzed with a segmented model, we need a formal language which can describe the number and position of the segments, the model function within each segment, and the difference between the input data and the model function value. We assume the input is given as a function of T uniformly spaced sampling times.

$$z(t) \qquad \text{for } t = 1 \ldots T$$

For convenience, we further assume that the z values are integers in the range from 1 to N. This is an appropriate model for computer applications in which data has been digitized by an analog-to-digital converter. The choice of N will introduce scaling issues similar to those discussed for the clustering problem in Section 5.4.

A waveform segmented into M periods will be described as a set of functions, $\{f_i\}$, for $1 \leq i \leq M$, and starting points, $\{t_i\}$, which satisfy

$$\begin{aligned} t_1 &= 1 \\ t_i &< t_{i+1} \qquad \text{for } i = 1 \ldots M-1 \\ t_M &\leq T \end{aligned}$$

For convenience in the definitions, we let t_{M+1} be defined as $T + 1$. The model function within the i^{th} segment is the function f_i, so

$$\hat{z}(t) = f_i(t), \quad \text{when } t_i \leq t < t_{i+1}$$

is the complete estimate of the noiseless data. The estimation error is modelled as noise:

$$\epsilon(t) = z(t) - \hat{z}(t)$$

Our formal language describes the segment starting times, $\{t_i\}$, with a clause, $s(t)$; the individual functions, f_i, with clauses, $s(f_i)$; and the estimation error with a clause, $s(\epsilon)$. The root productions define $s(\theta)$ to be the concatenation of $s(t)$ and the $s(f_i)$

$$s(z,\theta) \rightarrow s(\theta)\ s(z|\theta)$$

$$s(\theta) \rightarrow s(t)\ s(f)^*$$

$$s(z|\theta) \rightarrow s(\sigma)\ s(\epsilon)$$

The natural method of describing the set $\{t_i\}$ is with an enumeration of its elements:

$$s(t) \rightarrow s(t_1)s(t_2)\cdots s(t_M)$$

Given the ordering constraints between the segment starting times, it is inconvenient to assign a quantity of information to each separate starting-time description, as they would have to be contextually defined. A combinatorial measure of the information in all M times as a whole is

$$I(s(t)) = \log_2(T) + \log_2 \binom{T-1}{M-1}$$

because $t_1 = 1$ and the other $M-1$ starting times can be chosen as a group from the $T-1$ time samples. The $\log_2(T)$ term, which is a combinatorial measure of the information in the description of M, may be ignored, as it is independent of the $\{t_i\}$ and $\{f_i\}$, and only appears once in the final MI criterion. It therefore does not affect the estimate.

For large M and T, the factorials in the second term may be approximated, if computationally desirable, with the dominant terms of Stirling's formula, to give

$$I(s(t)) \approx (T-1)\log_2(T-1)) - (M-1)\log_2(M-1) - (T-M)\log_2(T-M)$$

More drastically, for the common situation where $M \ll T$, the approximation

$$I(s(t)) \approx M \log_2 T - \log_2(M!)$$

simplifies the optimization.

Note that the information measure above penalizes complex models for the information in the description of the segment borders in addition to the information in each segment function, which will be described below. This is more of a penalty in complex models than a less natural, but plausible, alternative which might be reasonable if T is fixed for the application. In this latter situation, one might allow that each of the

$$2^{T-1} = \sum_{M=1}^{T} \binom{T-1}{M-1}$$

segmentations are equally complex, in which case the description of the segmentation does not affect the estimate. Complex models would still be penalized however in the description of the $\{f_i\}$, which grows with N, but the segmentation itself need not be penalized. In what follows, we use the former measure of information rather than this latter one.

The information in $s(f_i)$ will depend on the class of functions allowed for each segment. We consider the simplest example, in which each function is a constant, so only this constant value need be specified.

$$s(f) \rightarrow 1|2|\ldots|N$$

We allow the function to take on any value that the input can, so a combinatorial notion of information gives

$$I(s(f_i)) = \log_2 N$$

for each segment and

$$I(s(f_1)\ s(f_2)\ldots s(f_M)) = M \log_2 N.$$

One method of extending the description and information measure to polynomial functions is discussed in Section 6.4.

To describe the fit between the segmentation and the input data, we assume a probabilistic additive noise model in which the error at each time sample is independent. In the simulations below, an additive Gaussian noise model with unknown variance is assumed. This allows a simple least-squares computation of the constant levels, and information in the fit, for any given time segmentation. This form of noise model is reasonable for the context in which this problem arose, which is described in the appendix. We are not concerned with the details of the description of the variance, σ^2, since we will measure its information as constant, so it will not directly affect the MI criterion. Its effect will be indirect, through the information in $s(\epsilon(t))$. The error is therefore described with the production rule

$$s(\epsilon) \rightarrow s(\sigma)s(\epsilon_1)\ s(\epsilon_2)\ldots s(\epsilon_T)$$

With this model, an entropic information measure for the description of the error at each time sample is

$$I(s(\epsilon_t)) \approx -\log_2(\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\epsilon_t^2}{2\sigma^2}})$$

where we use the value of the normal density as a rectangular approximation of its integral over a one z-unit interval. This approximation is valid for $\sigma \gg 1$, which is the interesting case, for if the noise level were very low, the solution would be

trivial. Edge effects, due to the truncation of the tails of error distribution at $z = 1$ and $z = N$ are reasonably ignored, as long as we can assume the true models and the input data are well described with integers from 1 to N. Summing the above information measure over the entire time period gives

$$I(s(\epsilon)) \approx T \log_2 \sigma + \frac{\log_2 e}{2\sigma^2} \sum_{t=1}^{T} \epsilon_t^2 + T \log_2 \sqrt{2\pi}$$

in which we can drop the last term, as it does not affect the minimization.

The MI criterion then says to choose the model, $\hat{\theta}$, which minimizes

$$I(s(z, \hat{\theta})) = \log_2 \binom{T-1}{M-1} + M \log_2 N + T \log_2 \sigma + \frac{\log_2 e}{2\sigma^2} \sum_{t=1}^{T} \epsilon_t^2$$

6.2 Optimization Methods

Optimization of this criterion will be approximate. There are 2^{T-1} partitions, so an exhaustive search is intractable. We rely instead on local search techniques as outlined in Section 3.4. Descriptions of partitions are transformed into coarser ones when the information measure is reduced, starting with the finest partition as an initial state.

The minimization of the criterion is organized about two parameters, σ and the set $\{t_i\}$. The set of possible σ is considered to range from 1 to 50 in increments of 1, which are searched exhaustively. The exact range and step size are not crucial to the method, as will be seen below. The point of an exhaustive search is that it is simple to implement and guaranteed to work. It is generally the best technique when there are a small number of states in the space to search. Here, the fifty values are a reasonable space, because this search can be intimately intertwined

with the segmentation optimization in a manner which allows both searches to occur in parallel. We show below that we do not need to optimize σ separately for each value of $\{t_i\}$, or vice versa.

To see how this occurs, assume first that the optimal value for σ is known. The structure $\{t_i\}$ is then chosen with a one-pass steepest-descent method starting with the finest partition as initial state, described as T segments, with $t_i = i$. The local transformations simply merge a pair of adjacent segments into one combined segment. There are $M-1$ such transformations on a description with M segments. The time structure of such a merger is simple: the new segment spans from the starting point of the earlier segment until the ending point of the later segment. This is effected syntactically by simply deleting a node $s(t_i)$ (other than $s(t_1)$, which is fixed as 1). To determine the $\hat{z}$-value for the new segment, we use a well-known property of the Gaussian noise structure: minimizing the expression for $I(s(z,\theta))$ with a fixed segmentation results in a least-squares estimate of the $\hat{z}$-values within each segment. As there are no weightings in the squared-error summation, this is the simple average of the z values. It is implemented recursively by forming the $\hat{z}$ value for a new segment as the average of the $\hat{z}$ values of the two merging segments, weighted by their lengths.

The one-pass steepest-descent algorithm simply considers each of the $M-1$ possible mergers of adjacent segments and performs whichever reduces the information criterion the most, assuming the information does decrease with some merger. It is straightforward to calculate the increase in squared error, $\Delta\epsilon$, and the decrease in model complexity when two segments starting at t_i and t_{i+1} merge. By additively updating the length of each segment as $l_i = t_{i+1} - t_i$, and the sum

of the z values for each segment as $s_i = \sum_{t=t_i}^{t_{i+1}-1} z(t)$, every time two segments merge, the test becomes

$$\text{merge iff } \Delta I = \frac{\log_2 e}{2\sigma^2}\Delta\epsilon - \log_2(N\frac{T-M+1}{M-1}) < 0$$

where

$$\Delta\epsilon = \frac{s_i^2}{l_i} + \frac{s_{i+1}^2}{l_{i+1}} - \frac{(s_i + s_{i+1})^2}{l_i + l_{i+1}}$$

This is repeated with whichever segment most reduces the total information cost until no merger can reduce the information with the particular σ chosen.

At this point we can examine how the optimization over different values of σ interacts with segmentation. From the form of ΔI above, it is clear that any merger which takes place at a given value of σ would also take place if σ were increased. This makes intuitive sense; at higher noise levels, we have less reason to suppose that a given change in z values is really a valid segment border. Thus, an algorithm could scan through a set of increasing values of σ, and for each select the segmentation which is optimal by the above local search technique, always starting with the finest partition. However, all of the merges at each σ value would be repeated needlessly at the next larger value of σ. It is more efficient to start at a small value of σ, find the best partition, then increase σ, and further aggregate the segments until the segmentation is again locally optimal. These operations repeat alternately until the maximum value of σ is reached. From another point of view, the optimal segmentation at each σ value is being used as a starting point for the local search at the next larger value of σ.

At some point along the way, the information will be minimized, and the algorithm stores this segmentation as the MI estimate. Note that this algorithm does

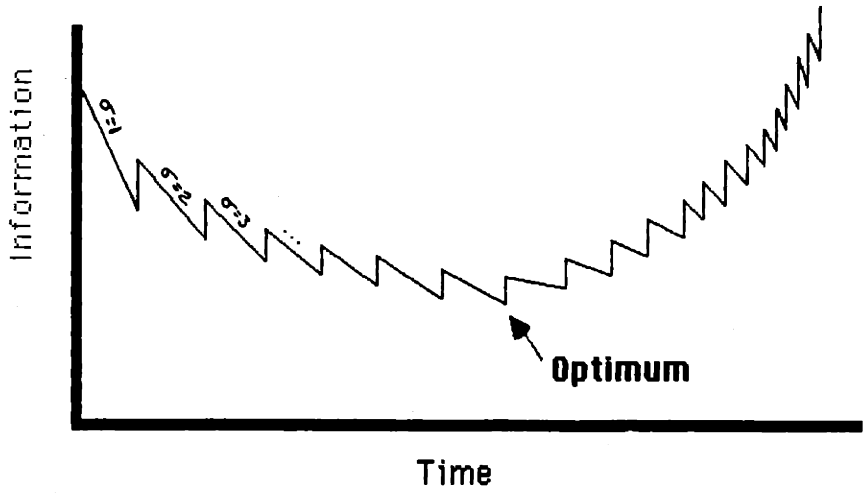


Figure 6.1 Changes in information as M decreases and σ increases.

not decrease the information measure with every operation. When σ is increased, the term $T \log_2 \sigma$ in the definition of $I(s(z,\theta))$ increases. Therefore, it is necessary to verify that the algorithm does terminate at some point. That this is so is indicated in Figure 6.1 which shows how the total information in the description of the data varies over the course of the algorithm. The downward steps at constant σ levels correspond to mergers which reduce the total information, while the upward steps occur when σ is increased. The exhaustive search for σ allows some constant number, C, of upward steps, depending on the resolution desired, and there are at most $T-1$ downward steps before the analysis returns a single segment. The "outer loop" of the algorithm therefore requires at most $T+C-1$ operations. The "inner loop" of the algorithm finds the segment which best merges with the following segment. As there are $M-1$ values of ΔI to compare, and this is always less than T, the search is complete in time $O(T^2)$. The local optimum is indicated schematically with an asterisk on Figure 6.1.

6.3 Results

Figures 6.2–5 show typical results of an implementation of the above optimization scheme with $N = 50$ and $T = 100$. In each of these figures, the points indicate the input data and the horizontal lines depict the output of the estimator. For the first three figures, the data was generated using the same "true" segmentation, but with different pseudo-random Gaussian noise levels. The "true" model, used to generate the data, was very close to the estimated model shown in Figure 6.2. That estimate is correct vertically, and the segment borders appear within one pixel horizontally of the "correct" locations. For variety, Figure 6.5 uses a different "true" segmentation. Notice in the first three of these figures how the three leftmost segments, which are distinguishable at the lower noise levels, become merged into a single segment at the highest of the three noise levels, Figure 6.4. The values of $\hat{s}$, printed at the lower right of each figure, are the estimated standard deviations, in units of vertical pixels. These estimates are correct in the figures shown. The types of results shown are typical for the algorithm.

The algorithm was coded in Turbo Pascal [Borland 1984], and runs on the 50 by 100 arrays shown in approximately 30 seconds on an IBM PC. It was not specifically coded for speed and could certainly be improved by a factor of 5 to 10 with minimal effort.

6.4 Discussion

It is insightful to look at the above segmentation algorithm as a form of nonlinear filtering. Both the input and output are one dimensional functions in the same discretized space. Compared to linear filtering techniques, such as

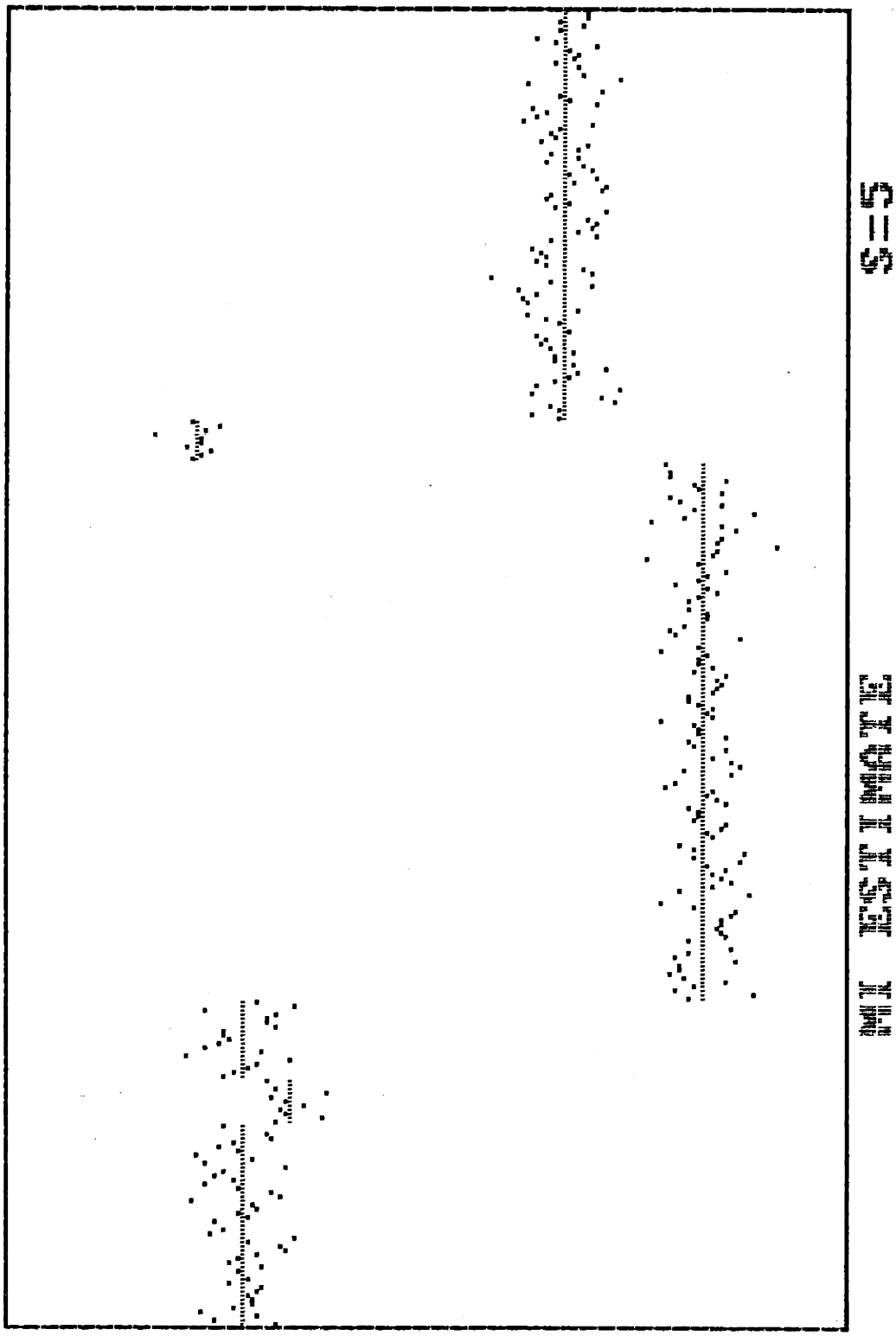


Figure 6.2 Results of Algorithm, First Data-set, $\sigma = 5$

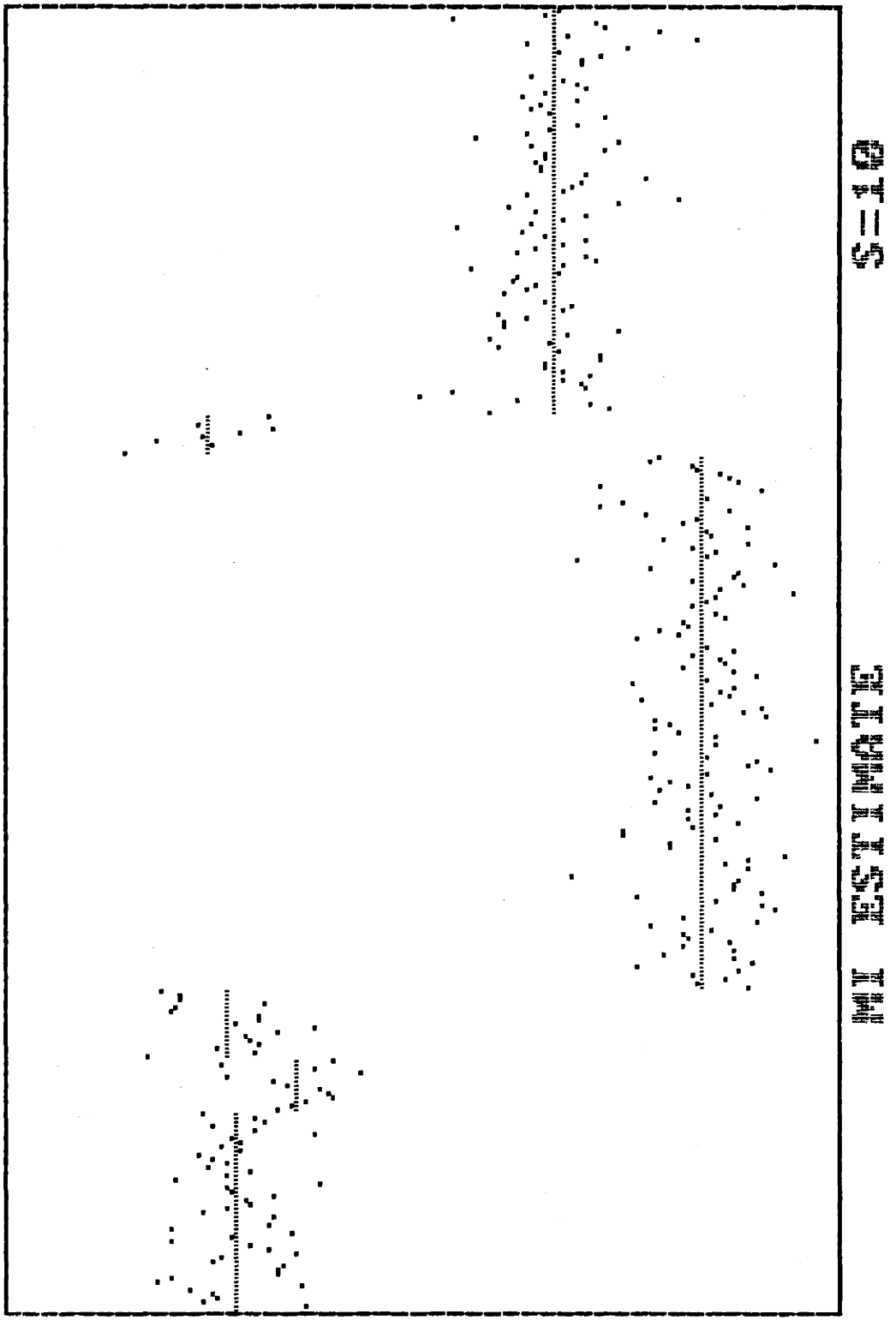


Figure 6.3 Results of Algorithm, First Data-set, $\sigma = 10$

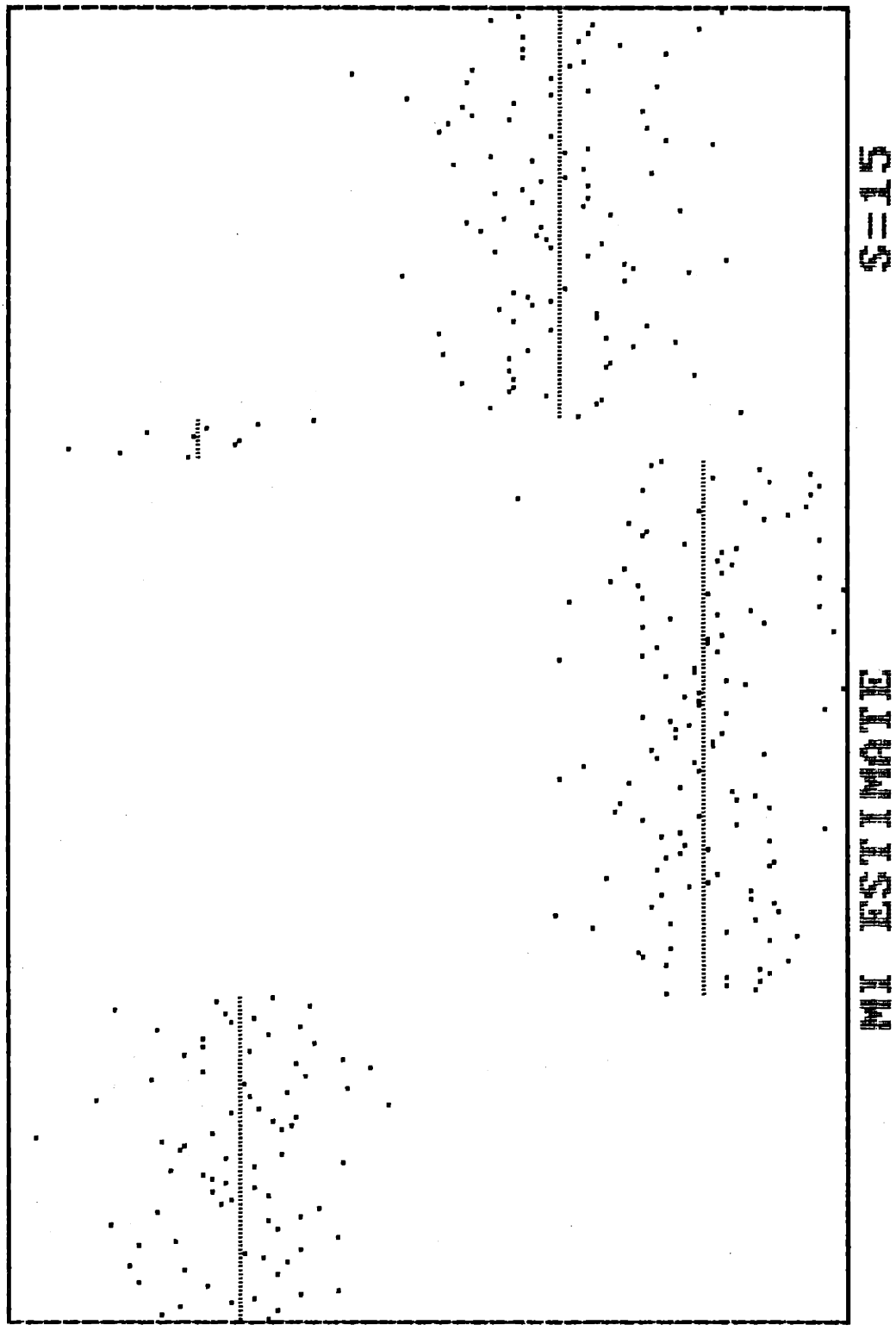


Figure 6.4 Results of Algorithm, First Data-set, $\sigma = 15$

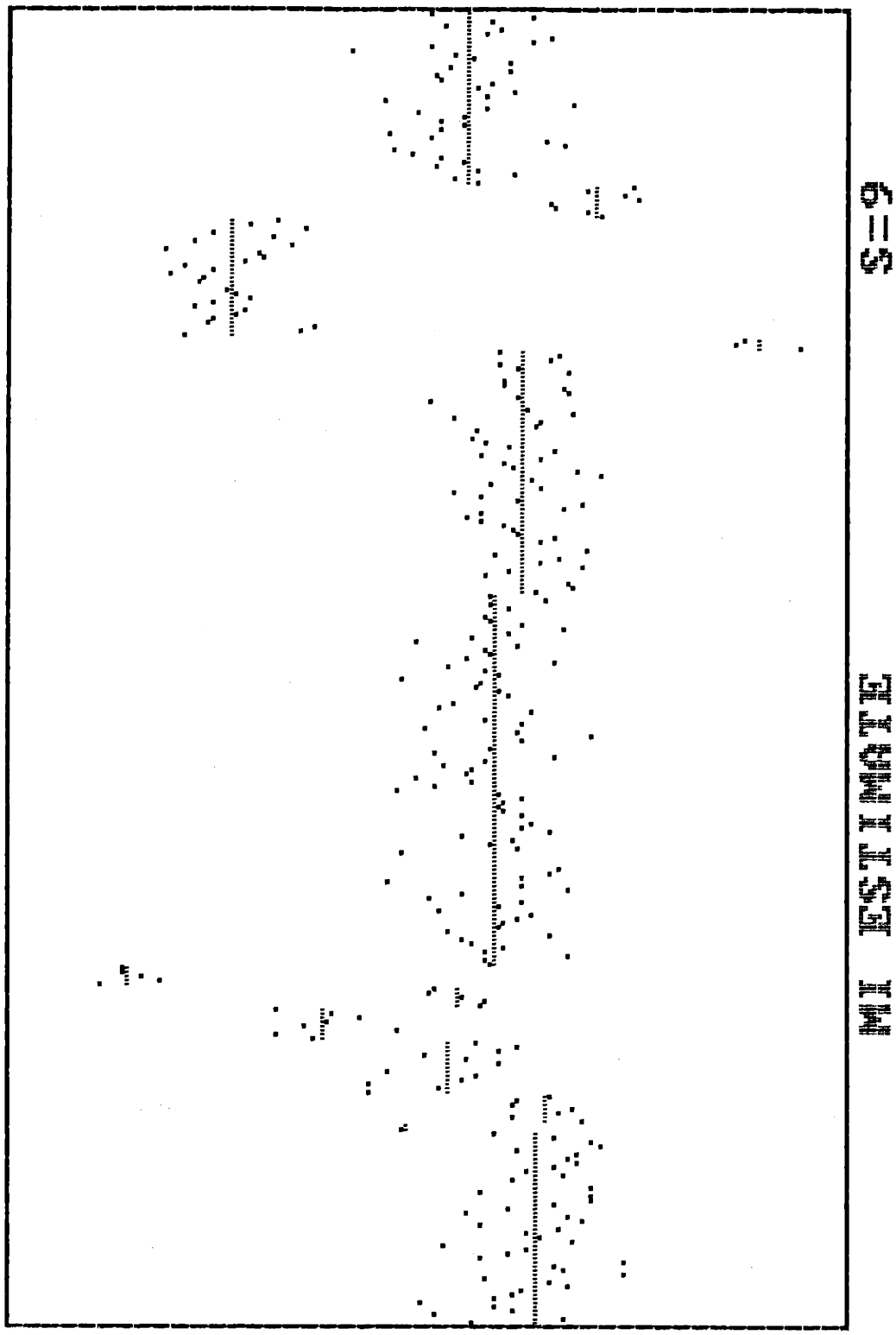


Figure 6.5 Results of Algorithm, Second Data-set, $\sigma = 10$

Wiener filtering or Kalman filtering, the method makes very different types of assumptions, and the resulting filter has very different properties. Instead of the statistical assumptions required for linear filtering, we are assuming that our *a priori* knowledge indicates this segmented class of models is appropriate. When this class of models is valid, as in the application described in the appendix, estimators based on them should perform far superior to linear filters.

The resulting nonlinear filters have very different characteristic properties from linear filters. As is typical of nonlinear filters, the results in the above section show "graceless degradation" as the noise level is increased. The filter operates quite well, with little or no error, up to a certain critical noise threshold, at which point the estimate suddenly diverges drastically from the true structure. A linear filter could be designed which shows at least some dip in the top left of Figure 6.4 in the region where the nonlinear filter completely misses the change in level. On the other hand, no linear filter could perform as well as the nonlinear filter in figures 6.2 and 6.3. In these cases, the nonlinear filter simultaneously eliminates noise and sharpens the edges of the signal at the segment boundaries. As these characteristics both involve the high frequency components of the signal, the former attenuating them and the latter amplifying them, no linear filter can have these properties.

It is also worth pointing out that the above computation method is naturally parallelizable. With an appropriate linear arrangement of processors, the cost, or savings, in information by merging a segment with its neighbor to the right can be computed simultaneously for each segment. The merger of maximum savings can be located in $\log_2 M$ time, and the maximum total time for up to T segment mergers is reduced to $O(T \log_2 T)$.

Extension of the technique to vector constants is straightforward if the individual components can vary independently. We merely describe each component individually, so the information in $I(f_i)$ is multiplied by the dimension of z.

However, more complex families of functions present some conceptual difficulties if they are not naturally associated with the discrete z values. For example, it would be interesting to apply this method to curve-fitting of polynomials. It is difficult to describe these however, without encountering problems of truncating real numbers or assigning complexities to the rationals. One approach, which naturally generalizes the constant function case, is to describe k^{th} order polynomials with a sequence of $k+1$ values for $\hat{z}$, distributed evenly across the segment. For example, a first order polynomial on the i^{th} segment, $t_i \leq t < t_{i+1}$, is a line, which can be described by specifying its two endpoints, $\hat{z}(t_i)$ and $\hat{z}(t_{i+1}-1)$. Then $f_i(t)$ can be interpolated linearly between these two values. A rounding convention, e.g. to round off $\hat{z}$-values to the nearest integer, must be assumed in the interpretation.

More generally, a k^{th} order polynomial can be specified with $k+1$ $\hat{z}$-values, $\hat{z}^{(0)}, \hat{z}^{(1)}, \ldots, \hat{z}^{(k)}$, as the unique k^{th} order polynomial, f_i, such that

$$\hat{z}^{(j)} = f_i(t_i^{(j)})$$

where the time values, $t_i^{(0)}, t_i^{(1)}, \ldots, t_i^{(k)}$, are spread maximally and evenly across the segment, e.g., with

$$t_i^{(j)} = t_i + \frac{j}{k}(t_{i+1} - 1 - t_i)$$

Because the interpolated polynomial can exceed the range $1 \leq f_i(t) \leq N$, we could incorporate an additional convention in the interpretation: values of $\hat{z}$ greater than N or less than 1 are truncated to these extreme values.

Although it is not altogether clear what set of N^{k+1} polynomials can be described in this manner for k intermediate between 0 and the length of the segment, they do form a reasonable set at the two extremes of this range. The zero$^{\text{th}}$ order polynomials are constants, described by a single z-value, the first order polynomials are described by their two endpoints, and the most complex polynomials are described by listing each of the z values the polynomial passes through in the segment. In this last case there are exactly $N^{t_{i+1}-t_i}$ such functions, one for each set of possible input values for the segment, so the fit can always be made exact. We hasten to point out that we have not implemented this class of functions, or seriously examined optimization methods for the structural portion of the problem. As discussed in Chapter 8, we leave this as a problem for future research.

Finally, we should mention that the above problems may be solved with many different optimization techniques other than the simple approach implemented here. Rather than this one-pass merge-only algorithm, a combination of split and merge transformations, starting from a number of random segmentations, is more likely to avoid local maxima. It is also possible to develop recursive approximations to the criterion which work in real time, but with some delay, for applications such as in the Appendix. Other techniques, such as dynamic programming, are also feasible.

The MI framework gives a new perspective to a large class of important segmentation problems. Two obvious ones are the segmentation of continuous speech and music. Another is the detection of times at which changes occur in the parameters, or order, of a dynamic system, due to operator adjustments or component failures. Medical waveforms, such as electrocardiograms, provide other interesting segmentation problems. In chapter 8 we suggest these as areas for future research.

Chapter 7

IMAGE PROCESSING

As a final application of the MI framework, we propose a new approach to machine vision problems, based on the notion of "image complexity." We take a simple problem from the field of image processing, and pose it in terms of structure estimation. The problem here is to reconstruct simple binary images by "eliminating noise". By *binary image*, we are restricting ourselves to images which are describable as a rectangular array of $\{0,1\}$ values. For example, the image in Figure 7.1 is a 128×128 array of pixels, each of which is either ON or OFF, with no intermediate grey levels. In this model, noise has the effect of inverting a bit, from 0 to 1, or from 1 to 0, i.e. it is additive, modulo 2. Given a noisy image, such as Figure 7.1, we wish to estimate the "structure of the image." We will present an estimator which produces Figure 7.2 as an estimate of the structure of a noiseless image for this particular input. Possible applications of this, and similar problems, may be found in the field of industrial robotics, in situations where a

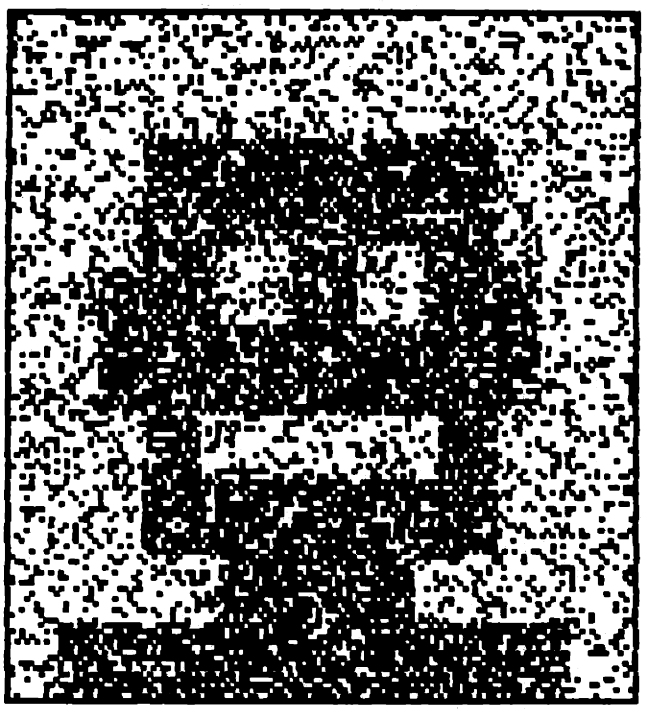


Figure 7.1 Noisy Binary Input Image to Process

noiseless binary image is sufficient information for parts location. A survey of other image segmentation techniques can be found in Pavlidis [1977].

In order to reconstruct images in this way, the MI framework first requires that a language be designed for describing images, and that we have some prior notion of simplicity and complexity for images. For illustrative purposes, we take a very simple model of image structure which is appropriate only if our *a priori* knowledge is to expect "boxey" rectangular images such as Figure 7.2, in which all boundary lines are orthogonal to the array borders. As discussed below in section 7.5, more versatile image models can be incorporated into the same framework. A measure of the fit between an input, such as Figure 7.1, and a model, such as Figure 7.2, is easily obtained by counting the number of pixels of the image which agree with the model. This will be modified somewhat below to incorporate a probabilistic noise model.

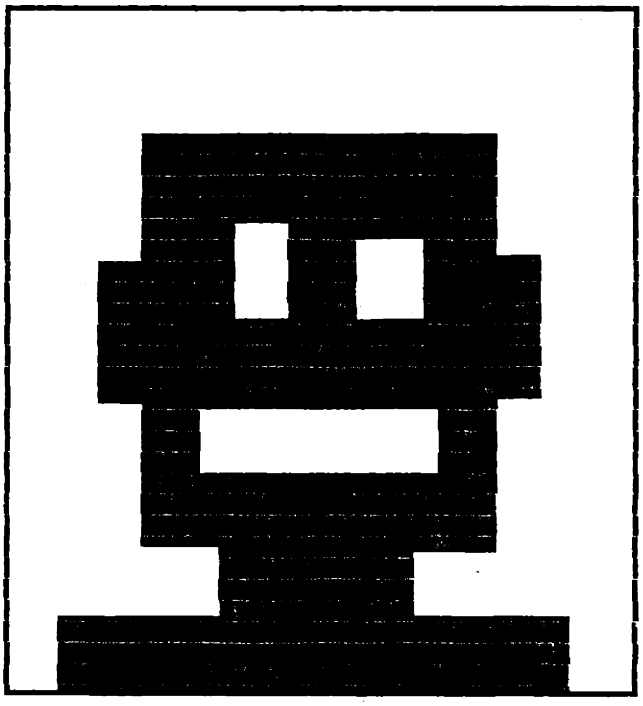


Figure 7.2 Reconstructed Image

It is insightful to think of this problem as a two-dimensional generalization of the one-dimensional segmentation problem of Chapter 6. The trade off between simple models and good fit comes about in an analogous manner. Here, the simplest model will be the null image—a blank image is easily described—and Figure 7.1 can be described as a null image in which all the pixels that are set ON are described as noise. At the other extreme, every pixel in the figure can be described as part of the image structure—Figure 7 may well be a noiseless image of many small random squares and rectangles—but it would be a very complex image to describe. The estimator which we will develop below chooses Figure 7.2 as the model of intermediate complexity between these extremes which, along with a description of the pixels where Figures 7.1 and 7.2 differ, gives the most concise description of Figure 7.1.

Compared to the one-dimensional segmentation of Chapter 6, this problem incorporates the simplification of models restricted to binary values, so there are

only two possible levels for each segment. The problem of estimating the level, given the structure, is therefore easier. But the segmentation problem is much more complex, as each region of the plane might have a boundary of any complexity. In the linear segmentation problem, only the number of segments and the positions of their endpoints had to be determined. Here we must determine the number of regions in the plane, the number of bordering segments for each region, and the positions of each border. The structural possibilities are far more complex.

We should note that this problem comes from Marroquin [1985], but here we use a very different class of models and method of solution. In Marroquin's approach, an image is modelled as a Markov Random Field (MRF), and a stochastic relaxation solution technique is employed. The underlying similarity lies in the fact that both classes of models incorporate some sort of local spatial interactions, but ours are geometric, based on properties of descriptions, while Marroquin's are probabilistic. Because our "boxey" model class has very different properties from the MRF model, the solutions display very different properties. A comparison of the two methods is made in Section 7.3.

7.1 Languages and Information for Binary Images

In a visual field discretized into N pixels, a binary image, i, can be represented as a point in a N-dimensional vector space where each component is restricted to the field $\{0,1\}$. Some ordering of the pixels, e.g., lexicographic on the X and Y image axes, can be specified to determine the vector indexing. Similarly, an estimate

of an image, $\hat{i}$, and the noise, n, which corrupts this image can also be represented as points in this space. With the additive noise model described above, we have

$$i = \hat{i} + n \quad \text{with } i, \hat{i}, n \in \{0,1\}^N$$

where the addition is modulo 2.

We can describe such images with different forms of image description languages, each designed around different types of image structure models. The most common model describes an image as a sequence of N pixel descriptions, where a pixel is described with the terms "0" or "1" to denote its being off or on. This form of pixel by pixel description, or "bitmap", is formalized by the grammar

$$BitMap \rightarrow P_1\ P_2 \ldots P_N$$

$$P_i \rightarrow 0|1$$

Such a description can be interpreted as an N-vector in the obvious way. A combinatorial notion of information gives

$$I(P_i) = 1$$

$$I(BitMap) = N$$

If we interpret an image as *noise*, then different languages and information measures are called for, depending on the structure of the noise. Given our noise structure, in which pixels are randomly inverted independently of each other, and with unknown probability, p, we augment the syntax above with a term $\hat{p}$ to describe p, and employ an entropic measure of information for the pixels.

$$noise \rightarrow \hat{p}\ BitMap$$

$$I(\hat{p}) = C$$

$$I(P_i) = \begin{cases} \log_2 \hat{p}, & \text{if } P_i = 1; \\ \log_2(1-\hat{p}), & \text{if } P_i = 0. \end{cases}$$

The choice of syntax for describing $\hat{p}$ will not affect anything below. One option is to use an integer between 0 and N, interpreted as the numerator over a denominator of N. Because the information, $I(\hat{p})$, is assumed constant, and $\hat{p}$ appears exactly once in the overall description, we may ignore its contribution to the total information. Then, by Gibbs' theorem, the value of $\hat{p}$ which minimizes $I(noise)$ for any given bitmap is the sample probability,

$$\hat{p} = \frac{N_1}{N}$$

where N_1 is the total number of times that "1" appears in the bitmap.

Measuring the information in $\hat{p}$ as constant is reasonable because only an extremely narrow and high *a priori* probability distribution for p would have any affect on our estimate of the noise. For a reasonably sized image, such as the 2^{14} pixels in any of the figures in this chapter, the description of the bitmap within the noise term is expected to contain $2^{14}H(p)$ bits of information, where H is the standard entropy function.

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p)$$

Then $I(noise) = NH(\hat{p}) + C$. A coefficient of 2^{14} will cause this term to outweigh the information in the description of $\hat{p}$, unless p is known in advance with far more accuracy than is reasonable for a noise model.

For noise models with correlation, e.g. diagonal stripes or the short horizontal bars of television "snow", other description languages and measures would, be more appropriate.

To describe noiseless images, we need to employ a notion of simple images which is relevant to our particular vision application. In a simple industrial robotics application, for example, this might consist of *templates*, translations and rotations of views of an inventory of components to be located. In the most general context, one would want to describe images in terms of "the furniture of everyday life," such as tables, chairs, people, faces, etc. We wish here to use a picture description language of intermediate versatility between these extremes, but far more general than templates. For demonstration purposes, we employ a very simple notion of a *connected rectangular image* (*cri*), which is bounded by a simple closed curve of horizontal and vertical segments. (We discuss below the effect of relaxing this simple-closed curve condition.) Figure 7.2 contains four *cri*'s, corresponding to the outline and three "holes" in the picture.

To describe a *cri*, we could list the coordinates of the vertices in a cyclic order around its perimeter with

$$\begin{aligned} cri &\rightarrow V_1 V_2 \ldots V_k \\ V_i &\rightarrow (X, Y) \\ X &\rightarrow 1|2|\ldots|128 \\ Y &\rightarrow 1|2|\ldots|128 \end{aligned}$$

and measure information combinatorially as

$$I(X) = I(Y) = \log_2 128 = 7$$

for our 128×128 images. However, given our constraint that the segments are either horizontal or vertical, the X or Y value (alternately) repeats from the previous vertex, and half of the description is redundant. Accordingly, we can either adjust the language to eliminate the redundancy with

$$cri \rightarrow X\ Y\ X\ Y \ldots X\ Y$$

or change the information measure to $I(V) = 7$, without recursing down to the X and Y terms. In either case, a connected rectangular image with k vertices would have $I(cri) = 7k$. More generally, this will be

$$I(cri) \approx \frac{\log_2 N}{2} k$$

Note k must be even because horizontal and vertical segments alternate in the cycle.

At a finer level of analysis, we may wish to eliminate the redundancy allowed by the language in the k cyclic permutations of the sequence for listing the vertices. We could for example require the first vertex listed to be the first according to some exogenous ordering, such as the lexicographic ordering. We may also account for the V^* term with an information term such as $I^*(k)$, as discussed in Section 3.2. But as these two factors respectively subtract and add terms of order $\log k$, which largely cancel, we are content to measure $I(cri)$ as given above. We also ignore the fraction of descriptions corresponding to images which are not simple closed curves; this becomes significant with large k, and suggests an interesting, but difficult, counting problem.

We interpret the connected rectangular image description as an image vector in which the pixels interior to the perimeter are ON and those exterior to it are OFF. The notion of "interior" is well defined here whether or not we restrict descriptions to those where the edges form a simple closed curve.

The next step is to combine a set of m connected rectangular images into a *general rectangular image (gri)*, with

$$gri \rightarrow cri_1\ cri_2 \ldots cri_m$$

which we interpret as the sum modulo 2 of the individual images. For reasons discussed below, we do not wish to allow the case in which some of the connected rectangular components *partially* overlap. In measuring $I(gri)$, we again ignore the $I^*(m)$, and the redundancy in the $m!$ possible orderings.

Finally, our complete description of an image combines a general rectangular image and a noise image.

$$\begin{aligned} s(z,\theta) &\rightarrow s(\theta)\ s(z|\theta) \\ s(\theta) &\rightarrow gri \\ s(z|\theta) &\rightarrow noise \end{aligned}$$

The interpretation of the image is again the sum modulo 2 of the interpretation of the components. The final MI criterion is

$$I(s(z,\theta)) = \frac{\log_2 N}{2} K + NH(\hat{p}) + C$$

where K is the total number of vertices, summing k over the connected images, and $\hat{p}$ is the fraction of pixels set ON in the description of the noise, i.e. the sample probability. The computational convenience of a complexity measure that is a function of K rather than the m values of k will be seen below. It allows an algorithm which operates in the image plane, rather than syntactically, and is the real reason for ignoring the logarithmic terms above.

Note that our language incorporates a slight asymmetry between ON pixels and OFF pixels. A completely null image is described by the null sentence, but an input of all 1's requires a four-vertex description of the entire array. (This asymmetry appears in the optimization algorithm below as a boundary value condition on the estimated structure. We fix a border of 0's around the array storing the estimated structure.) The slight preference which results for black images on a white background, as opposed to the reverse, could be eliminated, if desired, with a syntax for a *gri* that includes one bit of information for inverting an image.

7.2 Optimization Methods

The problem of minimizing $I(s(z,\theta))$ for this problem is formidable. In terms of bit inversion in the image plane, the search space is a hypercube of 2^{14} dimensions. There are $2^{2^{14}}$ distinct images which can be described as a general 128×128 rectangular image, and could therefore be the output of our estimator. Exhaustive search is not an option. In syntactic terms, the space is even larger, as each image has an infinite number of distinct descriptions. Furthermore, in addition to determining the structure of the general rectangular image, we also need to determine the noise parameter, $\hat{p}$.

There are many ways this criterion might be optimized, at least approximately. Our concern here is not to find the best optimization method, so much as to demonstrate that the MI criterion itself is a viable way to approach vision and pattern recognition problems in general. Accordingly, we have developed a simple, fast algorithm for approximately optimizing the criterion which, we feel, demonstrates the feasibility of the method. For actual machine vision applications, more attention would have to be paid to the optimization techniques, in order to avoid some of the local minima which befuddle our simple algorithm. These problems and possible improvements to the algorithm are discussed below.

We again rely upon local search techniques, but here we embed them in a slightly more sophisticated *multigrid* superstructure. The multigrid approach involves successive approximations to an estimate using successively finer levels of resolution. In the image processing literature, this superstructure is often referred to as a *quartic picture tree*, or *pyramid*, and is used in a variety of image processing algorithms. This approach will be seen to allow a very simple set of transformations to quickly lead to a local minimum which avoids many possible local pitfalls.

It does have some trouble at the levels of high resolution however. We therefore augment our first set of transformations with a different class of transformations which make adjustments at the finest level of resolution. The multigrid technique is presented in Section 7.2.1, and the follow-up "slide" transformations are given in Section 7.2.2.

Although these sections are very specific to rectangular images, and involve a considerable amount of implementation details, the reader should not lose sight of the bigger picture. Both sets of transformations are centered on local operations which are suggested naturally by the clause structure of the grammar for describing images. As outlined in Chapter 3, the transformations can be described as adding, deleting, splitting, or merging the clauses which describe connected rectangular images. They are performed whenever the information measure in the complete sentence is reduced. The algorithm terminates when no further local improvements are possible.

7.2.1 MULTIGRID ALGORITHM

The multigrid portion of the algorithm involves repeating the same generalized transformations at increasingly fine levels of resolution. The set of transformations we employ can be stated either in terms of syntactic transformations which modify descriptions, or directly as operations on pixels in the "image plane". An interplay between these two modes of analysis is developed and exploited, to find a fast algorithm in the image plane that maintains the properties we specified in the description space.

Let R be the resolution index, which will vary here from coarsest, 7, down to finest, 0. More generally, for a square array of N pixels, where the number of pixels

on a side is some power of 2, R varies from $\frac{\log_2 N}{2}$ to 0. For each resolution value, we allow only certain restricted classes of descriptions. Specifically, we constrain the set of X and Y values which constitute the vertices in the description of a rectangular image to be multiples of 2^R. At the finest level, the coordinates are unconstrained, being multiples of 2^0, i.e. arbitrary integers. At each level of resolution, the algorithm searches for a locally optimum estimate, using the best estimate from the previous, coarser, level of resolution as the initial value.

We index each pixel at the finest level with (x, y) coordinates ranging from 1 to $\sqrt{N}$. The fundamental transformation we employ is easiest to describe in the image space: we insert or delete a *unit square* at the current level of resolution. By "unit square" we refer to a block of pixels with (x, y) coordinates satisfying

$$i_x 2^R < x \leq (i_x + 1)2^R \text{ and } i_y 2^R < y \leq (i_y + 1)2^R \text{ for some } i_x, i_y$$

The i_x and i_y terms are the coordinates of the unit square in the grid at level R, and satisfy

$$0 \leq i_x, i_y < \frac{\sqrt{N}}{2^R}$$

Following the greedy approach discussed in Section 3.4, we wish to consider all possible transformations, and perform the one which results in the largest decrease in $I(s(z, \theta))$, repeating this until no transformation results in an information savings. At multigrid level R, there are $2^{-2R}N$ unit squares to consider inverting. For any inversion, there will be some change, ΔK, in the total number of vertices required in the image description. Geometric considerations show ΔK can take only five values, because rectangular images always contain an even number of

vertices, and inverting a single rectangle can change the number of vertices by no more than ± 4. Therefore,

$$\Delta K \in \{-4, -2, 0, 2, 4\}$$

There must also be an inversion within the corresponding unit square of the noise description: the 0 and 1 descriptions of the pixels within the unit square must change to 1 and 0 respectively. This is required in order that the sum modulo 2 of the various contributions to each pixel remain invariant. In other words, when we change $s(\theta)$ we must adjust $s(z|\theta)$ appropriately, so that the complete sentence remains a description of z. From the form of the MI criterion at the end of Section 7.1, we know the effect of inverting a unit square is to increase $I(gri)$ by $(\log_2 \sqrt{N})\Delta K$, and change $I(noise)$ by a term which depends on the number of $\{0,1\}$ values of the pixels within the unit square, and the value of $\hat{p}$. We first address the determination of ΔK.

The effect of these unit-square inversion transformations on the descriptive complexity, $I(gri)$, depends on the local context of the inverted square. Nine of the 256 possible local contexts in which a unit square might be inserted (set to 1) are indicated pictorially in Figure 7.3. Inserting or deleting a unit square has different affects on ΔK depending on the eight surrounding unit squares. It is independent of the remainder of the image however, because whether each corner is or is not a vertex in the most concise description of the *gri* depends only on the four squares meeting at the corner, and only the four corners of the square which inverts may change their vertex status.

Each row of Figure 7.3 shows a separate example. For each, in passing from the left figure to the right, the central unit square of a 3×3 group of unit squares

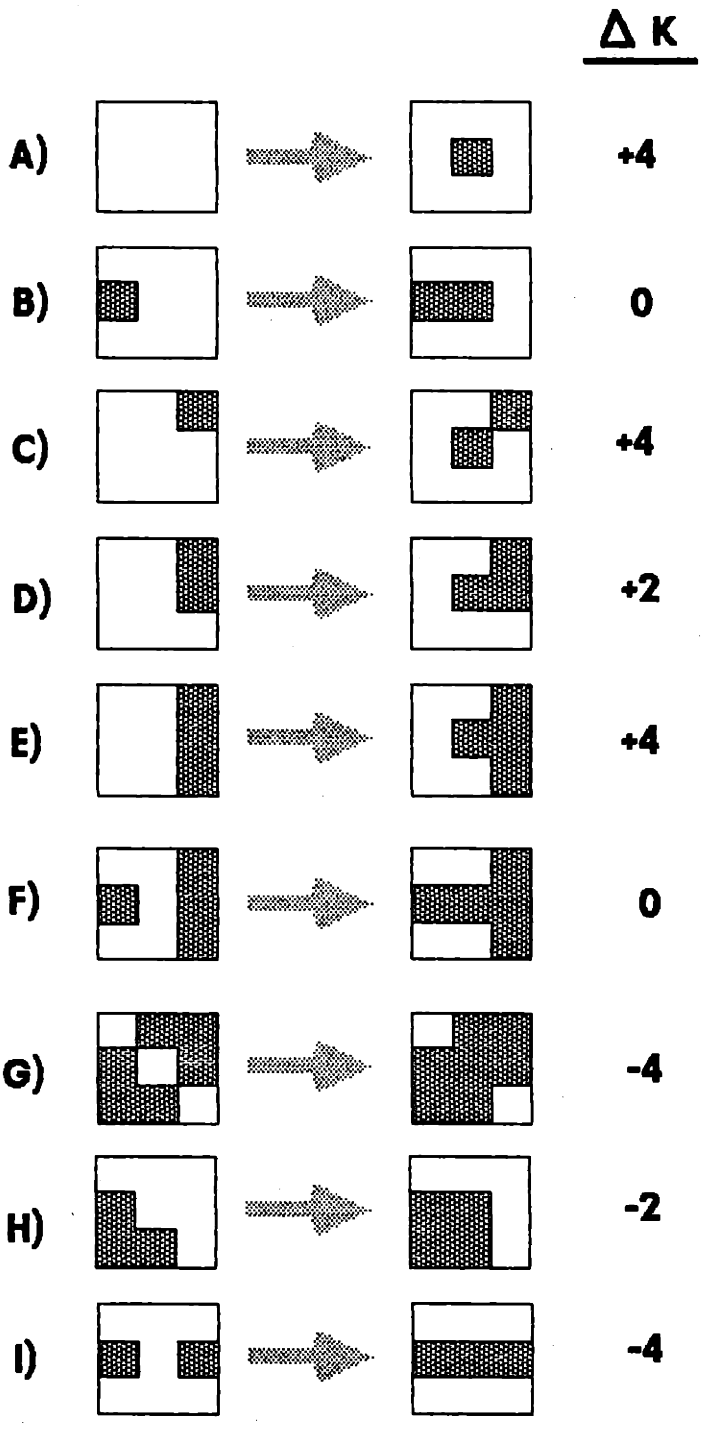


Figure 7.3 Local Image Transformations, and Effects on Complexity

is inserted (set to 1). The net increase in K, the total number of vertices in the entire figure, is indicated in the column ΔK. The transformations are reversible, so transformations in which unit squares are removed (set to 0) correspond to passing from a figure on the right to a figure on the left. The associated change in K will then be the negative of the quantity printed.

The types of transformations shown in Figure 7.3 suggest themselves naturally when one seeks ways of perturbing a rectangular image into a similar image. The particular transformations we have implemented are restricted to squares on grid boundaries however, while one would generally want the affine rectangular transformations. This leads to a class of local search failures discussed below. Corresponding to these unit square inversions are various syntactic transformations of the description of the image which involve adding, deleting, merging or splitting terms describing connected rectangular images. Respective to the rows A–I, the syntactic operations are to:

(A,C) Insert a new *cri* term with four vertices in the *gri* list.
(B) Increment or decrement a single X_i term by 2^R.
(D) Insert an X term and a Y term (adjacently) into a *cri* list.
(E) Insert four terms, $X\ Y\ X\ Y$, into a *cri* list.
(F,G,I) Various forms of splicing two *cri* lists together or dividing one into two.
(H) Deleting an XY pair from a *cri* list.

Admittedly, these and other operations are unified and more concisely described in terms of image plane operations than in terms of syntactic descriptions. However, it is important to note that they can all be formalized as local syntactic transformations, and we require that the change in complexity of the description be determined before a transformation is effected, in order to ensure the MI criterion always decreases. The syntactic statements of the transformations also have the

advantage of generalizing immediately to the affine rectangular boxes, where the image-plane versions would be more complex. The unification of the transformations by means of unit boxes is only allowed because we can develop a measure of change in descriptive complexity based on local image properties.

For example, the top row of the figure indicates that inserting a unit square which is not adjacent to any other picture elements increases the total number of vertices by 4. Conversely, deleting an isolated square eliminates four vertices. The second row indicates that sliding a vertical edge to the left or right only adjusts the lengths of segments, and does not affect image complexity, and so $\Delta K = 0$.

Row C illustrates a rather subjective judgement of image complexity. The restriction mentioned above, that the edges of a connected rectangular figure form a simple closed curve, was implemented so that this configuration is just as complex as one in which the central square does not share a corner vertex with the region in the upper right. In other words, we give no benefit, in terms of simplicity, to diagonal adjacencies. If the perimeter of a connected rectangular image were allowed to intersect itself, two rectangles with a single common vertex could be described as one object with six vertices, rather than two objects totalling eight.

Similarly, the restriction that two connected rectangular images not partially intersect is implemented because we feel the left image of row G is more complex than the right image, and should be measured as two notched objects with six vertices total, rather than two overlapping squares totalling only two vertices. We therefore require that no image borders, whether in the same object or two distinct objects, may cross. Complete images may be embedded in other images however, as in Figure 7.2.

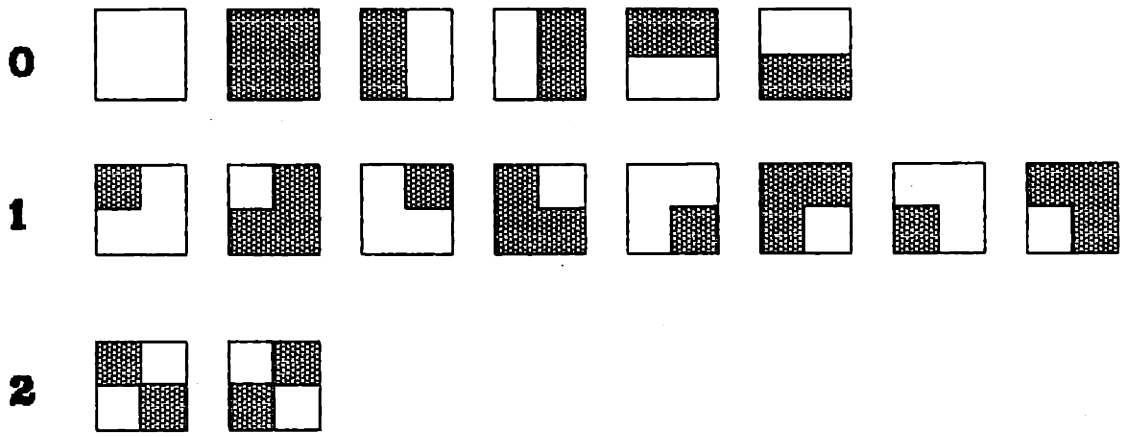


Figure 7.4 Sixteen Combinations of 4 Squares Being ON or OFF, and Corresponding Number of Vertices Required to Describe the Local Portion of the Image

For computational purposes, we need an algorithm which gives us the value of ΔK as a function of the eight surrounding squares. One way to develop this algorithm is to begin with the observation that a grid point either is, or is not, a vertex of an image as a function of the four squares which meet at the grid point. Figure 7.4 indicates the sixteen possible arrangements in which four squares surrounding a grid point may be independently ON or OFF. The corresponding number indicates if the central grid point must be described once, twice, or not at all, in the shortest description of the image. The eight arrangements with an odd number of squares ON each require the center grid point to be part of the description of the image, because a horizontal and vertical segment meet there. The two "checkerboard" arrangements require *two* descriptions of the vertex, because

of our constraint that borders not cross. The remaining six configurations require no description of the central vertex.

(Note incidentally that the algorithm would be simper without the border-crossing constraint, as then the vertex count would simply be the parity of the number of squares set. Casual experimental comparison of these two complexity measures did not show a systematic significant difference in the final estimate. All the results presented here use the constraint.)

The patterns of Figure 7.4 are easily recognized, so an algorithm can straightforwardly determine the number of vertices in a 3×3 group by adding up this quantity for each of the four 2×2 groups within the 3×3 group. By doing this once for a given 3×3 arrangement, and again for the same arrangement but with the central square inverted, the difference between these two quantities gives the ΔK value of Figure 7.3. These values can be computed once and stored for the 256 possible contexts of a unit square.

We now address the question of how the noise term changes when a unit square is inverted, and the problem of estimating $\hat{p}$. As mentioned above, the 0's and 1's in that portion of the bitmap which describes pixels in the inverted unit square must invert to 1's and 0's respectively. This ensures that the interpretation of the description remains unchanged. Thus, the number of noise bits described with a "1" in this region changes from its current value, n_1, to $A_R - n_1$, where $A_R = 2^{2R}$ is the area, in pixels, of the unit square at resolution R. The increase in "1" bits in the bitmap is then $\Delta N_1 = A_R - 2n_1$. The exact increase in $I(noise)$, taking into account the optimal change in $\hat{p}$ due to the change in N_1 is then

$$\begin{aligned}\Delta I(noise) =&(N_1 + \Delta N_1)\log_2(\hat{p} + \Delta\hat{p}) + (N - N_1 - \Delta N_1)\log_2(1 - \hat{p} - \Delta\hat{p}) \\ &- [N_1 \log_2 \hat{p} + (N - N_1)\log_2(1 - \hat{p})]\end{aligned}$$

where $\Delta\hat{p} = \frac{\Delta N_1}{N}$. Ideally, this would be computed for each inversion, and $\hat{p}$ would be updated. Rather than compute this each time however, it is reasonable to linearize it by holding $\hat{p}$ constant. The increase in $I(noise)$ then becomes

$$\Delta I(noise) \approx (2n_1 - A_R) \log_2 \frac{\hat{p}}{1-\hat{p}}$$

This is a quite innocuous approximation because the area of a unit box is typically much smaller than N, so $\Delta\hat{p}$ for any inversion is quite small.

The algorithm then operates as if p were a known constant during each multigrid level, using the value of $\hat{p}$ estimated from the previous level. In practice this converges rapidly to its correct value. This could be modified if desired, by repeating the analysis at the same level with a new $\hat{p}$, alternately estimating the structure and estimating $\hat{p}$ in the manner of the EM algorithm [Dempster 1977, Feder 1987], until they converged. However, we expect this would run slower and produce identical results, because the coarser levels exist essentially to provide an initial description to modify at the finest level, where previous errors can generally be corrected.

Combining the changes in information due to the complexity, ΔK, and the fit, $\Delta I(noise)$, we get the following decision rule:

$$\text{Invert unit square if } \gamma = (\log_2 \sqrt{N})\Delta K + (2n_1 - A_R) \log_2 \tfrac{\hat{p}}{1-\hat{p}} < 0$$

which we interpret in the usual "greedy" manner, always making the transformation of maximum improvement. This is easily implemented by storing a table of the decision variable, γ, for each unit square. This is searched for the most negative entry. Note that after a transformation takes place, at most nine entries in this table must be updated, the inverted square and its eight neighbors, because

the structural change affects the 3×3 neighborhood which determines their ΔK term. (This locality is the real computational benefit of linearizing the $\Delta I(noise)$ term. In the nonlinear form, the value of γ for every unit box would have to be updated slightly after every transformation.)

This multigrid optimization, although quite effective, utilizes local transformations, and can not be expected to find the global optimum. After implementing it and watching it succumb to a certain type of problem, a second class of local transformation suggested itself.

7.2.2 SLIDING TRANSFORMATIONS

The problem with the multigrid method is that at the finest level or two of resolution, the number of pixels in a unit square is quite low. The maximum savings in noise information is then less than the information required to describe two new vertices. Accordingly, the decision rule allows no structural additions. (The size at which this happens depends on $\hat{p}$. The algorithm can detect this and terminate the multigrid operations without making any computations at the finest levels.) It often happens though, that if a line of pixels in a row or column, or a rectangular group of unit squares, were all inverted together, the noise savings in this larger region would pay for the complexity.

A general fix for this problem is to consider all possible rectangular groupings of unit squares, and try to invert an entire group when no single square can. But at the finest resolution there are $\frac{N^2}{2}$ such rectangles, and we do not wish to spend the time to try all possibilities. Since the most common error we noticed after the multigrid operations was a segment which should be translated one or two pixels to its side, a reduced class of rectangles suggested itself. As indicated in Figure 7.5,

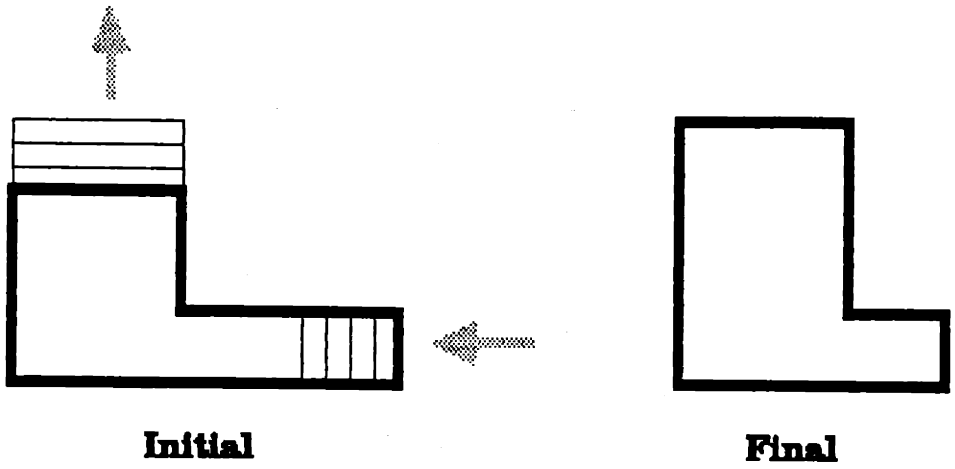


Figure 7.5 Effect of Sliding Transformations

we consider inverting a rectangle formed by translating an edge segment one unit its side. By iterating the operations, extended "slides" are possible, as long as each step reduces information.

Syntactically, this operation simply increments or decrements a single X or Y term in a *cri* description, but does not add complexity. Accordingly, the operation reduces the information measure whenever the line of pixels under consideration contains at least one more noise pixel than non-noise pixel (assuming, of course, that $\hat{p} \neq 0.5$).

As with the implementation of the multigrid algorithm, our implementation of this transformation is in the image plane rather than in the space of descriptions, but it is conceptually clearer in descriptive terms. Our implementation first finds all vertical boundaries, tries sliding them left or right, and then tries to slide horizontal edges up or down. These operations are repeated until no transformation is found which improves the information criterion. (In the image plane, a pixel by pixel scan is undertaken. A segment border is detected by the presence

of two neighboring pixels at which the estimate differs. The edge is followed, and the number of noise pixels along either side of the border is tabulated. If more than half are noise, the edge is translated.)

7.3 Results

Figures 7.6–8 show the results of the above optimization algorithm on a sequence of similar images, with progressively increasing noise levels. In each case, the upper image is the input, z, and the lower image is the the estimate, $\hat{\theta}$. These three inputs are all constructed from the same simple rectangular figure, corrupted by adding different pseudorandom noise images, with the indicated density of pixels independently set. We have therefore constructed an input in accordance with the noise model assumed above. The underlying figure from which the images were constructed happens to be identical with the lower image in Figure 7.6; the estimate here is exactly correct.

The types of performance shown in these figures is typical. For relatively low noise levels, ($0 < p < 0.25$), the estimate is generally exactly correct if the "true" image is truly boxey. At a higher noise level, such as in Figure 7.7, where p is 0.3, the estimator loses some of the finer details of the figure, but finds the larger structures. Increasing the noise level to 0.35 in Figure 7.8, large features are typically missed. The algorithm returns just the torso when $p = 0.40$. (At $p = 0.5$ the input is totally noise, independent of the underlying figure, and the algorithm returns a null image.)

The analogy between this two-dimensional segmentation problem and the one-dimensional problem of Chapter 6 is evident. These properties are typical of nonlinear filters, and it is again insightful to think of the estimator in these

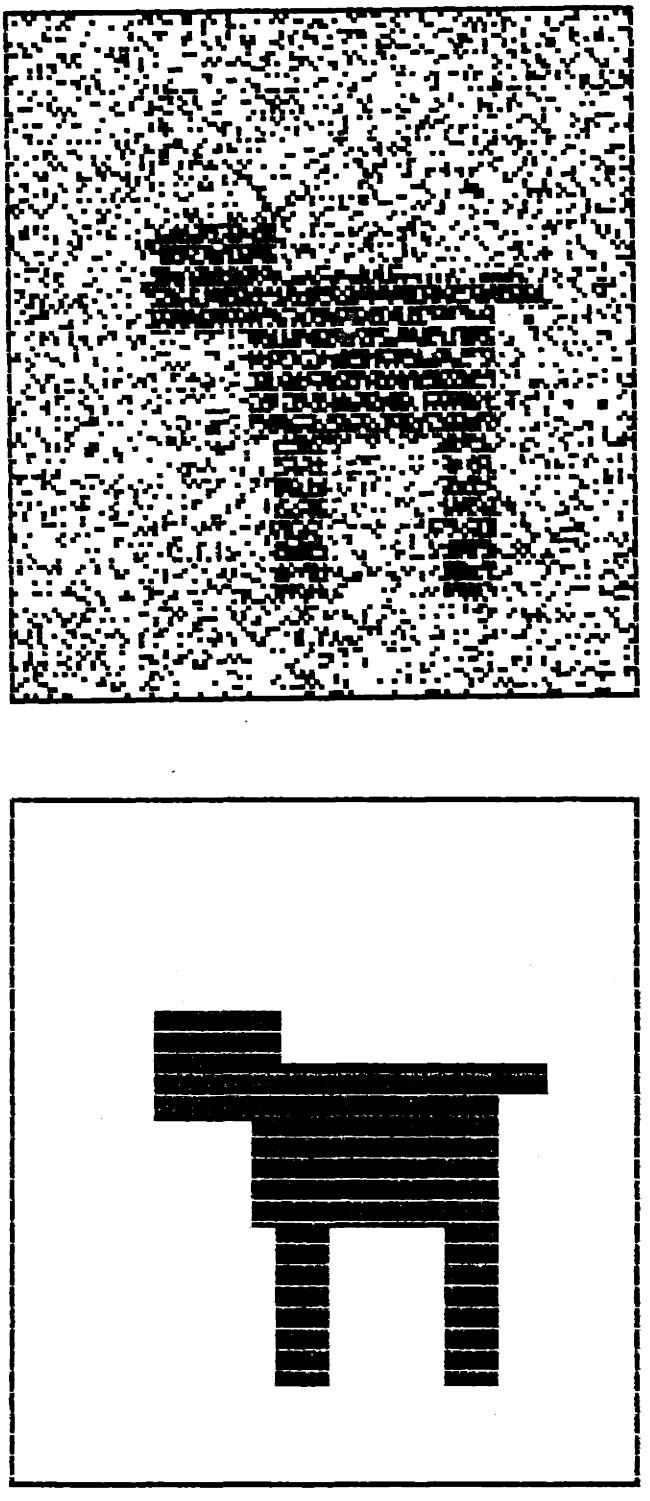


Figure 7.6 Input and Estimated Structure, $p = 0.25$

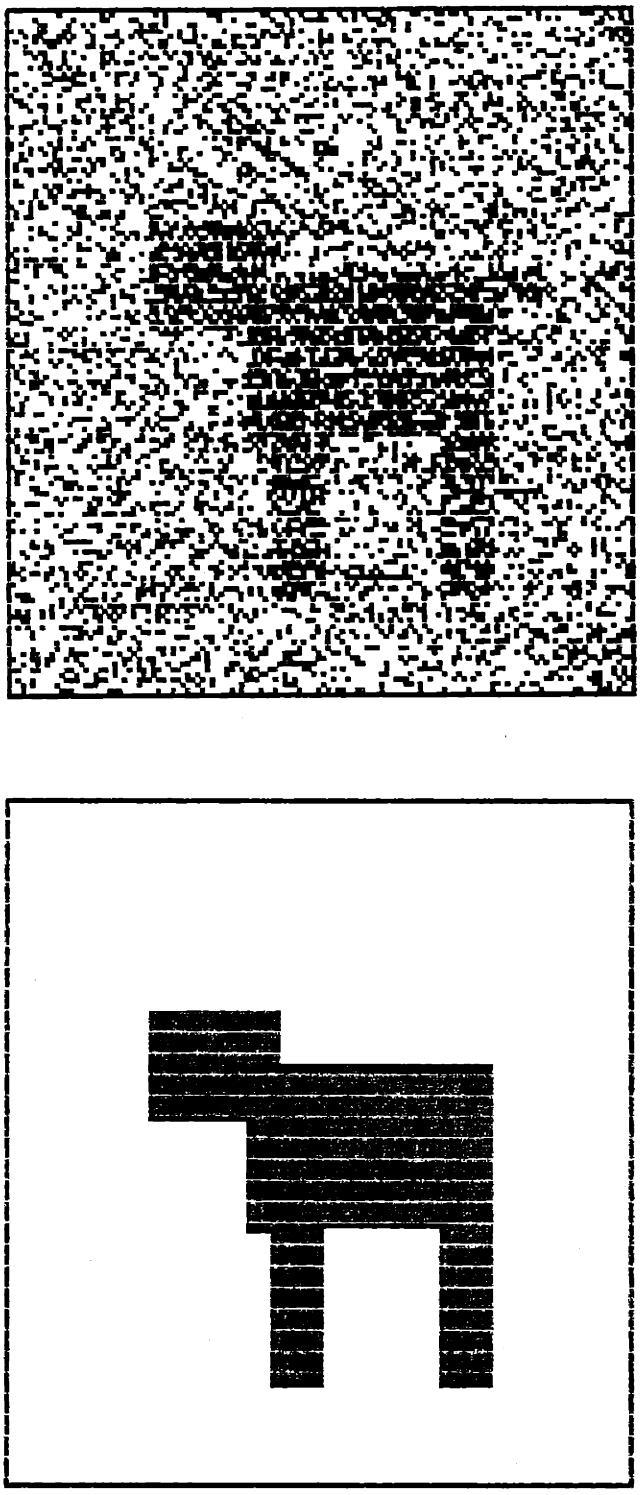


Figure 7.7 Input and Estimated Structure, $p = 0.30$

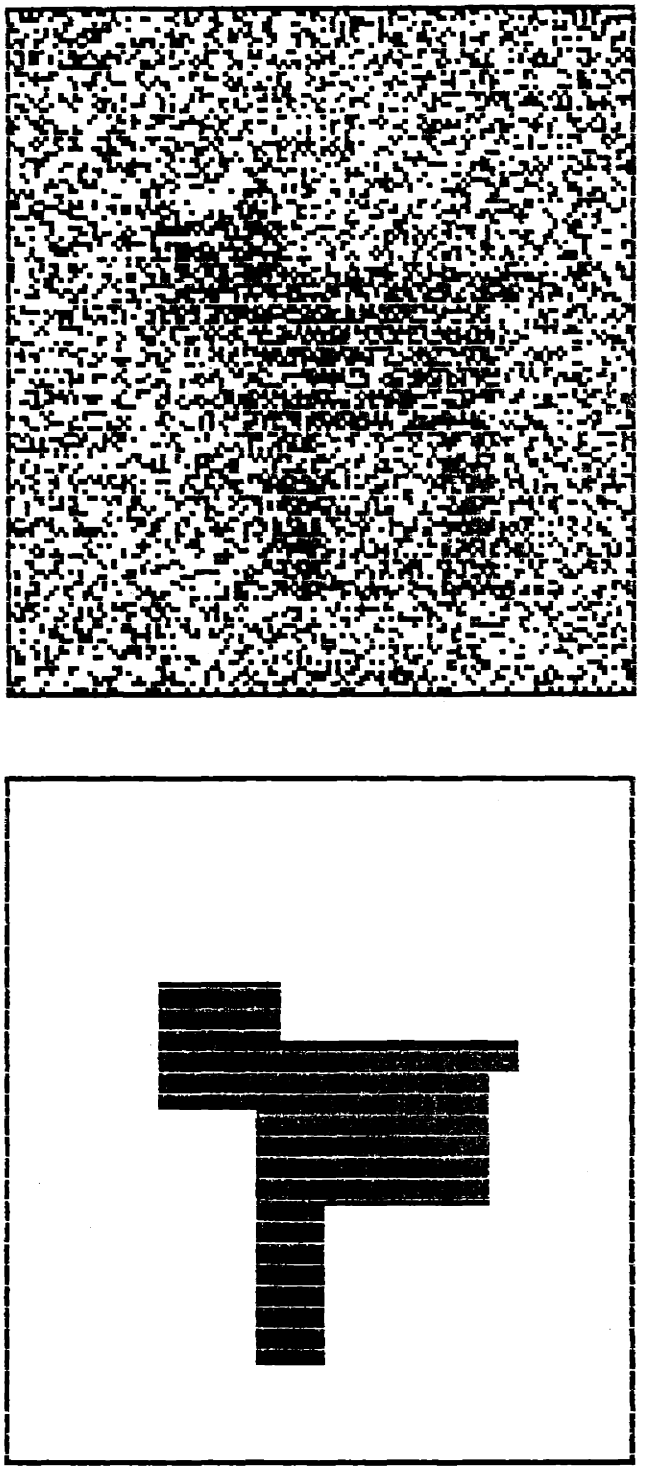


Figure 7.8 Input and Estimated Structure, $p = 0.35$

Figure 7.9 Image Restoration Based on Markov Random Field Model, from Marroquin [1985].

terms. The estimator filters the input image in a nonlinear way to arrive at the output image. A nonlinear filter has the advantage that it can simultaneously reduce noise and sharpen edges, which no linear filter can do. It has the disadvantage however, that above some critical noise levels it begins to make serious, unacceptable, estimation errors.

To compare our approach with a more linear approach, consider Figure 7.9, from Marroquin [1985]. From left to right, the figure shows a synthetic figure similar to ours, a noise-corrupted input (with $p = 0.35$), and a maximum likelihood estimate using a first-order Markov random field model. The details of Marroquin's method do not concern us here, but the general properties of his estimator do. It is not truly linear, as no method can be, with only binary values for output. However, a linear approach, such as a simple spatial filter, passed through a threshold comparator, would have similar characteristics. A linear technique is expected to provide at least a rough approximation to large features, even in high noise, and would give some indication of the leg amputated in Figure 7.8. On the other hand, a linear method will always give the type of textured edges seen in Figure 7.9, rather than the sharp edges of Figures 7.6–8.

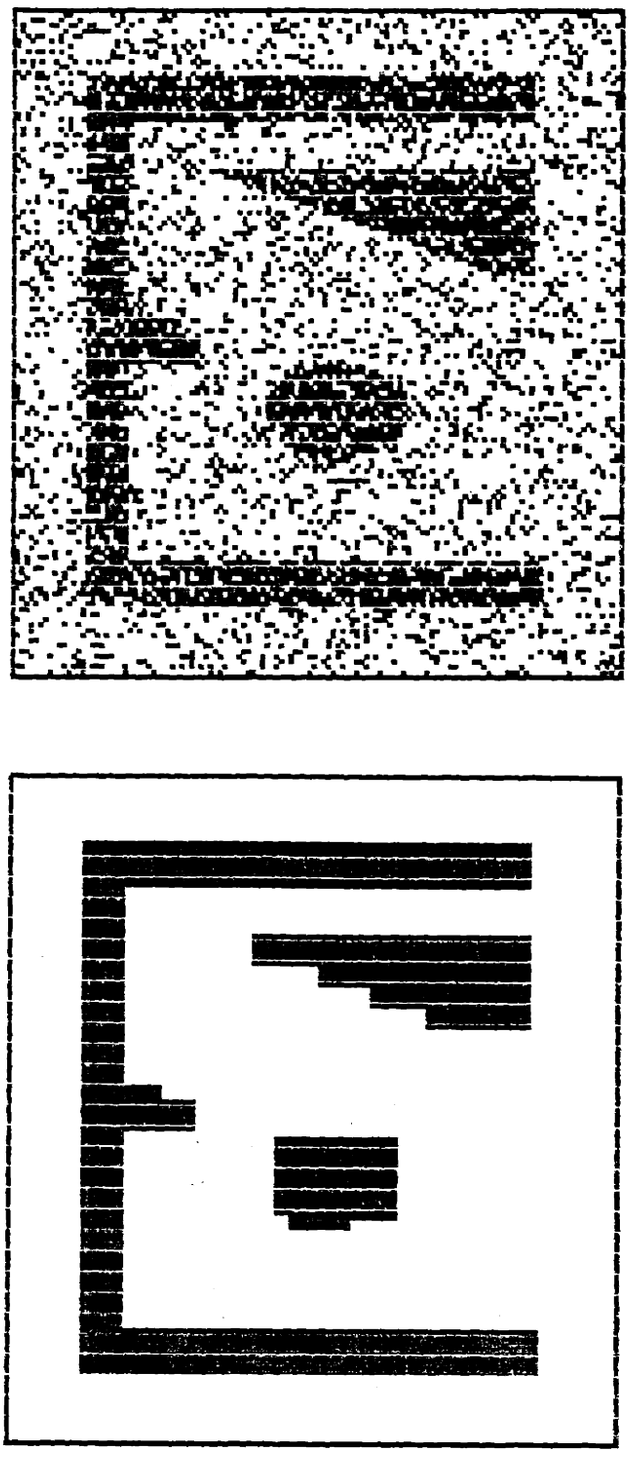


Figure 7.10 Image with Nonrectangular Components, and Estimate, $p = 0.2$

If the noise levels are not too high, and the rectangular model is appropriate to the image processing task, then this descriptive MI approach is clearly superior, because it incorporates the correct *a priori* information. For other classes of images, a different description language would be required. For example, in Figure 7.10 an image containing rectangular, triangular, and elliptical substructures is estimated with the rectangular algorithm. Not surprisingly, rectangular approximations to the nonrectangular components result. A more general image description language would incorporate terms for describing diagonals and curves.

The algorithm is coded in Turbo Pascal for an IBM PC, and runs in less than a minute for a 128×128 image. The program was developed with a higher priority given to the programmer's convenience than speed. A small effort would probably speed it up by a factor of ten. (For example, the determination of ΔK as a function of the 3×3 neighborhood is computed anew for every inversion considered. Much time would be saved if the values had been precomputed and stored in a table of the 256 possible contexts. Additional improvement would result if Turbo's 64K-byte memory capacity limitation were sufficient to allow various images to be byte-mapped rather than bit-mapped.)

7.4 Probabilistic Interpretation

There are two ways for a strict Bayesian to interpret this example probabilistically. One could look at the language and information measure as specifying an *a priori* distribution for all images, or one could interpret the resulting decision rule as specifying an *a priori* distribution for each unit square as a function of the eight surrounding squares. In the first case, the probability of an image could be

interpreted to be the normalization of the function, 2^{-K}, of the complexity of the image.

It is more interesting to examine the multigrid algorithm as if it were a likelihood ratio test technique for selecting between structures with known conditional probabilities. Rewriting the decision rule of Section 7.2.1 (for the case $\hat{p} < 0.5$) gives

$$\text{Declare square "inside" if } n_1 > \frac{A_R}{2} + \frac{\log\sqrt{N}}{2\log\frac{1-\hat{p}}{\hat{p}}}\Delta K$$

This makes intuitive sense, because $\frac{A_R}{2}$ is the decision boundary if the image does not change in complexity, e.g. for a "slide" transformation, and higher values of n_1 are required in proportion to the increase in complexity, ΔK. At higher noise levels, a larger number of bits are required in the fit, to justify the complexity.

Under the assumption that the unit square in question is entirely outside of the true image, our noise model makes n_1 the sum of A_R Bernoulli trials, i.e. a Bernoulli random variable with parameter p and mean pA_R. Under the assumption that the square is inside the image, the parameter becomes $1-p$. The likelihood ratio test that chooses between these two hypotheses, under the *a priori* assumption that a box is inside the image with probability P_{in} and outside the image with probability $1 - P_{in}$, is

$$\text{Declare square "inside" iff } n_1 > \frac{A_R}{2} + \frac{\log\frac{1-P_{in}}{P_{in}}}{2\log\frac{1-p}{p}}$$

By equating corresponding terms of these two decision rules, we obtain a relation between ΔK and P_{in}. The two approaches make the same decisions when

$$P_{in} = \frac{1}{1+\sqrt{N}^{\Delta K}}$$

Corresponding to the five possible values of ΔK, we get the following rather strong table of relations for our implementation in a $2^7 \times 2^7$ array.

ΔK	P_{in}
4	$\approx 2^{-28}$
2	$\approx 2^{-14}$
0	0.5
-2	$\approx 1 - 2^{-14}$
-4	$\approx 1 - 2^{-28}$

One could, if one were a strict Bayesian, interpret our method as one in which we merely selected these values as *a priori* conditional probabilities for a unit square being inside the true image, conditioned on knowledge of the eight surrounding squares. For each of the example contexts in Figure 7.3, we are making decisions as if the context told us the *a priori* probability of the center square being in the image. However, we view this as a rather roundabout interpretation of the results. We certainly did not design an estimator with any such probabilistic model in mind.

It is interesting to note that this gives us a Markov random field interpretation of our model. Each pixel has a conditional probability distribution determined by the values of the pixels in its neighborhood. It is not clear however, if this probability distribution satisfies the consistency conditions for a MRF. If so, this suggests that other methods, such as stochastic relaxation, may be useful for finding the MI estimate.

7.5 Extensions

The descriptive model and optimization algorithm above might be useful for severe data compression applications. However, they are intended merely to

demonstrate the concepts and practicality of structural estimation of images, and not as an end in itself. There are many ways to improve the optimization of the given criterion, and even more ways to develop more versatile criteria.

The optimization method would benefit from additional transformations which allowed arbitrary rectangles to invert, rather than only unit squares at the current multigrid level. For example, the missing appendage of Figure 7.8, which is clearly visible to the eye in the input, is missed because of weaknesses in the combinatorial optimization, rather than the MI criterion. If the correct rectangle were available as a transformation, the criterion would say to invert it. Many possible ways of increasing the quality of the available transformations without searching through the complete set of rectangles come to mind. There is no point in expounding on these here however, as the rectangular class of images is only of limited application.

For completeness, we should note that other optimization methods are always worth exploring in difficult combinatorial problems. Stochastic relaxation techniques would sometimes allow transformations in which information increases in the hopes of climbing out of certain local minima. Rather than always starting with the null image, multiple initial descriptions based on random rectangles, linear filters, or various heuristics could also lead to better minima. (In an application in which video images are to be compressed or analyzed, the estimated structure of the previous frame would be another good initial description.)

The locality of the transformations and decision rules presented above make them highly parallelizable. For example, each unit square at a given resolution level can be analyzed in parallel by a separate processor in a rectangular grid of processors. Another form of parallelization can be implemented by arranging a

cycle of processors corresponding to the cycle of edges or vertices defining a connected rectangular image. Many of the transformations above can be implemented by having each processor responsible for just those transformations which change its local geometry. Increased complexity would be accompanied by the splicing of additional processors into the loop.

Another interesting direction for future work is to develop other picture description languages, of greater eloquence, yet which also lead to efficient algorithms. Objects with curved borders seem, at this point, to be a difficult extension as they are generally computationally burdensome. A more promising step would be to design a language for arbitrary polygons. They are easy to describe in terms of pixel vertices, and the relations between the descriptions and the image space may not be too troublesome. (Of course, "jaggies" and other problems due to the discrete geometry of pixel arrays may be something of a nuisance.) The set of transformations could center on the operations of inserting, deleting, and translating vertices.

A different direction for extending the binary rectangular image class is to explore models with different grey levels. In addition to piecewise constant models, these allow piecewise linear models, in which image intensity is expected to vary smoothly across each segment.

Finally, we note that the MI criterion appears to be a very natural approach to the problem of extracting stereo depth. If segments of one image are similar to translated segments of another image, the joint description in terms of regions, translations, and residual differences provides a very concise description of the two images together.

Chapter 8

CONCLUSIONS

In this final chapter we review and discuss the approach to estimation advocated here. The summary in Section 8.1 emphasizes the logical structure of the framework, and the nature and function of the different parameters of estimation, when seen from this point of view. In Section 8.2 we discuss the method and interpret it from a range of vantage points. Finally, in Section 8.3 we suggest possibilities for extending the framework and mention several applications which seem amenable to the technique.

8.1 Summary

The goal of this work is to further our understanding of the basic issues and relations involved in the estimation of structure, while remaining cognizant of the need for practical methods to apply to real problems. We have formulated a class of structure estimation problems which are not handled satisfactorily by classical estimation means. Many problems in many fields can be formulated in these terms, and often new insights and new methods of solution appear.

To solve these problems, we propose that a formal description system be designed for each application which makes explicit the structural nature of the space of unknowns, and the relations between the members of this space. To the maximum extent possible, this formal language should maintain a structural relationship between the construction of individual sentences and the relevant aspects of the structures in the space. A hierarchical organization is both natural and powerful. It allows complexity measure for objects in this space to be formed in an additive manner. At any point in the hierarchical structure, the information measure then can be used to compare substructures in terms of a combination of simplicity, probability, and nonrandomness which is relevant for the application.

Following this hierarchical format, it is natural to describe the input to an estimator in terms of possible structures and residual difference between a structure and the data. An additive information measure then automatically results in a tradeoff between simplicity and fit in the final estimate. The MI criterion expresses what we desire in an estimate. This is clearly separated from the algorithmic question of how to find its optimum value. In general, this will be a difficult computational problem, and different methods of optimization are appropriate in different applications.

Our general intent is to make as explicit as possible the nature and effects of the choices required in any estimation procedure. From a mathematical point of view, these choices will be arbitrary. But, we believe that if description systems and information measures are chosen in accordance with "human taste and usage" then the performance of the resulting estimator will be deemed acceptable. Note that this philosophy contrasts strongly with many others, e.g., Rissanen [1986, p. 1085] who seeks "a foundation for statistical reasoning which is as free from

arbitrary choices as we can make it." We feel these arbitrary choices can not be eliminated, and instead we must seek to understand them.

We have applied this methodology to a range of problems and presented the results of simple simulations. The variety of problems illustrate the adaptability of the method: finite and infinite domains, probabilistic and nonprobabilistic domains, numeric and non-numeric domains can all be treated. In the FSM and MS examples, the inputs and outputs are unbounded in the sense that the observation might be of any length and the output might be of any complexity; in the other examples, the input and output are selected from finite, but large sets. Probabilistic assumptions are made in the realization models for MSs, segmentation, and image processing, but not FSMs or cluster analysis. The clustering, segmentation, and MS problems involve numeric terms, while the FSM and vision problems are basically non-numeric. For completeness, the details of these implementations have been described, but our primary intent in the examples is to demonstrate the feasibility, ease, and directness of the overall method, not to recommend particular algorithms.

We have not included any mathematical or logical demonstration of the validity of the method, because such a demonstration is not possible in the general context assumed here. Given that an infinite number of structures are generally compatible with the input data, a criterion which goes beyond the data must be appealed to. The step from data to structure is a form of nondemonstrative inference, and can only be given plausibility arguments.

This thesis contains five different types of arguments for the description-based MI approach to structure estimation. None of these, when examined in isolation, carry enormous weight, as the overall method can not be validated. Together,

however, they may be convincing enough to encourage others to pose problems in terms of the framework, and further explore the MI approach.

1. The method is an *insightful* and *natural* approach to many problems. Although such terms have only subjective meaning, in the final analysis this argument may be the most convincing. The different functions of description and information measuring are clearly separated; the tension between complexity and fit is clarified; and the MI criterion is presented as distinct from particular algorithms which optimize or approximate it.
2. The MI criterion reduces to many familiar special cases, such as MAP estimators, ML estimators, and Ockham's razor, which have been proposed for estimating structure. There is a sense in which we can say the MI estimator *interpolates* between all of these other methods.
3. The *versatility* of the method is phenomenal. A formal language framework allows MI estimates to be quickly and easily constructed for applications in very diverse fields. Many different kinds of observations and estimated structures can be integrated in a single estimator. The examples from the fields of grammatical inference, pattern recognition, signal processing, and machine vision are all seen to contain a common thread of structure estimation, which is addressed in a uniform manner.
4. The method appears to be the only way to accomplish certain results. For classes of structures in which no probability distribution is relevant, such as the image processing example of Chapter 7, a straightforward application of the MI criterion immediately yields results which can be obtained in no other framework.
5. The *case studies* should convince the reader of the worth of the method. It is important to note that they were not invented specifically to exercise the method. To the contrary, all the examples, except the image processing one, forced themselves on the author's attention while grappling with the application described in the Appendix. The image processing example was taken verbatim from Marroquin [1985].

In summary, we feel the description-based minimum information approach to structure estimation is an insightful and powerful technique for solving a wide variety of problems in diverse fields. It clarifies the opposition between simplicity and fit, and accommodates either probabilistic or nonprobabilistic models of structure distribution.

8.2 Discussion

The formal-description-based MI approach to structure estimation may be interpreted from many points of view. We feel it is a viable framework for estimation which can be used to generate new estimators quickly and easily, as well as to insightfully organize and classify the many special estimation techniques of Chapter 2. It exposes the relations between these different methods as well as the relations between the different components within a single method. However, we are well aware that the general approach seems to have no provable properties, so we wish to understand why the method seems to work as well as it does in the examples of Chapters 4–7. It is fruitful to consider the framework from a range of perspectives.

8.2.1 PRAGMATIC VIEWPOINT

From a purely pragmatic point of view, we can argue that there is no notion of a *true* answer to many structure estimation problems. Estimators produce structures which may be useful for certain purposes, such as data compression with finite-state Markov models, but the models need not be interpreted as true models of the world in any sense. From this point of view, we have a particular application we wish to automate, and we need some framework in which to proceed. An algorithm is developed in the framework, and tested on sample data. If it performs satisfactorily it is adopted for the application. The criterion by which one decides to adopt or not adopt the algorithm for new data is not always clear, especially when there is little understanding of the range of possible inputs and the underlying mechanisms generating the data. In this context, an MI estimator is not formally justified, but simply fills a need. The criterion is viewed as "an engineering solution" to a practical problem.

8.2.2 PSYCHOLOGICAL VIEWPOINT

From a psychological point of view, we can interpret an MI estimator as a way of understanding input data, that is analogous to human pattern recognition. One can plausibly argue that the balance between simplicity and fit which is explicit in an MI estimator is analogous to parallel tendencies which humans use when finding patterns in sensory input. Given two models which are perceived as having comparable complexity, one prefers the one which is perceived as fitting the data more closely; given two models which are perceived as fitting the data equally well, one prefers the one which is perceived as simpler. Furthermore, the success of the MI cluster analysis and vision algorithms in chapters 5 and 7 can be interpreted as support for the view that MI estimation is a good mathematical model of human pattern recognition in these applications.

However, as psychology is poorly understood, there is little we can comment, except to point out the analogy and leave it to the reader's introspection to evaluate its worth. Lest the MI model be too quickly discarded in some reader's minds due to the combinatorial optimization involved, it should be observed that the mind does incorporate a powerful ability to find reasonable solutions to many difficult combinatorial problems. Often, difficult problems can be acceptably "solved" by having a human visually examine a suitable two-dimensional graphic representation.

Many pattern recognition algorithms can only be evaluated relative to a human pattern recognition norm. We can not argue that the cluster analysis algorithm of Chapter 5 or the vision algorithm of Chapter 7 are "correct" in any absolute sense. We lack a formal principle for evaluating the estimator and comparing it to other putative estimators. Indeed, if we had such a formal principle,

we could use it to generate an estimation criterion rather than propose that the MI principle be used to generate a criterion. Lacking a principle, we evaluate the algorithm by seeing if it produces "reasonable" results compared to the patterns humans see in the data. In this respect, the cluster and vision algorithms fare well.

In light of the above, it appears that a reasonable avenue for proceeding would be to first endorse MI estimation as *a model of human pattern recognition*, and then to use this endorsement as a principle for selecting the MI criterion in particular computer applications. If it is felt that the MI principle is a valid model of human pattern recognition, then it is reasonable to employ the MI criterion in designing estimators. The principle of designing a computer algorithm in accordance with a psychological model is easily justified. In this regard we note that a structural approach can accommodate Julez's observation [1969, p. 580] that "visual perception occurs in hierarchical levels of increasing complexity." We are diffident about the first step however, as psychological modeling is outside the range of our expertise. Empirical investigation is suggested, to develop complexity measures which order the structure space in the way humans do.

8.2.3 LINGUISTIC VIEWPOINT

From a linguistic point of view, there is another parallel between MI estimation and human problem solving. A little studied, but essential aspect of human natural-language production is the ability to form concise descriptive expressions. A very simple model of sentence production involves two steps: first one has a thought, and then a sentence is chosen out of an infinite number of possibilities which expresses the thought. For example, a formal semantic model (e.g., similar

to Montague [1970, 1973]) can be given in which the same "thought" (an expression in first-order predicate calculus) is expressed by the following four sentences:

1. *I see the big red thing.*
2. *I see the thing that is both big and red.*
3. *It is the thing which is red and which is big that I see.*
4. *What I here and now see is the thing which is big and not small and that is either round or not round and which has the property of being red.*

Although there is obviously more to the art of rhetoric than merely choosing the most concise statement of a thought, a human ability clearly exists for finding relatively concise descriptions of thoughts from the infinite set of possibilities offered by natural language. Arguably, this could be used to support the claim that MI estimation is a justifiable model for human pattern recognition. It demonstrates a human ability which could be expressed formally as the combinatorial optimization problem of minimizing length within a set of sentences that share a common interpretation.

It also bears pointing out that the tendency towards concise exposition is not only a property of language production, but also a formidable force in language evolution. This is significant because true synchronic tendencies usually correspond to some diachronic tendency. As novel objects and structural relations become more common, language accommodates methods for providing more concise descriptions of them. Automobiles became *cars*, televisions became *TVs*, and personal computers became *PCs* as they appeared more frequently in descriptions. Conversely, if an MI estimator is judged to fare poorly, a natural method of repair is to correct the language or information measure to more accurately reflect the frequencies at which different substructures and relations are to appear.

A final comment from the linguistic point of view is that many linguists feel that the PSG formalism is an excellent model of natural language, Chomskian arguments to the contrary notwithstanding. If this is so, then the fields of application of the method are effectively unlimited, as descriptive systems may be constructed for any subject we can discuss. (Note incidentally, that any superficial similarity between the transformations on sentences described in Chapter 4 and Chomsky's transformational syntax for modelling natural language is spurious.)

8.2.4 PHILOSOPHICAL VIEWPOINT

There is an enormous body of literature, spanning over two thousand years, dealing with problems of induction, language, symbols, and representation, from a philosophical point of view. As we could not begin to do justice to this corpus, we restrict our comments to what we consider to be the two most relevant points. An MI estimator can be considered to produce a *theory* (e.g., a general structure such as a FSM) from particulars (i.e. its input data), and so must contend with the problems of induction which plague all theories of theories.

The classic problem of induction originates with Hume [1748] who points out that there is never a logical argument from particular observed evidence to general conclusions. His classic example, *that the sun will rise tomorrow*, is never certain, no matter how much evidence we have that the sun always rises. Future data might contradict a generality, whatever the past data may be. Analogously, we can not expect to find a logical argument that an MI estimator, or any other estimator based on any other criterion, generates the true structure for which it was designed. Accordingly, we are not daunted by critics who complain that the MI estimate may not be "correct." We can not hope to show validity in any absolute sense.

A second problem which inductive theories must contend with is Goodman's [1954] *new riddle of induction*, sometime referred to as "the Goodman paradox". Goodman argues that inductive arguments are implicit in many contexts, including the use of everyday language, and that no inductive technique can select a unique answer on logical grounds alone. He provides a simple example to demonstrate that an infinite number of contradictory conclusions can all be reached from the same data, using the same principles of induction. The term *grue* is coined, and defined to mean *green if observed before some future time t, and blue after t*. With this definition, any evidence we may have that an object is green is equally good evidence that it is grue. Why then do we not expect objects which appear green now to change color at time t? After all, we have excellent evidence supporting the proposition that the objects are grue.

Somehow, a principle is needed which is prejudiced against the property *grue* vis à vis *green*. The description-based MI approach can provide this prejudice if properties such as *green, blue* and *grue* are not ontologically equivalent. Instead, we follow the intuitive notion that *grue* is described in terms of *green* and *blue*, and measured in a way that reflects its greater complexity and information. Given that the properties *green* and *grue* fit the existing data equally well, the MI principle requires that the simpler, i.e. *green*, be selected.

Following the two horns of the *Nature/Nurture* controversy, the locus of this prejudice can be placed in either of two camps. Fodor [1975] argues that a particular formal language of thought exists innately as a property of the human mind, and the complexity we feel in the terms *bleen* and *grue* are due to their relatively complex descriptions in this language. For someone in Fodor's camp the essential problem is to discover the vocabulary and constructs of this language of

thought, and the description-based MI framework could be incorporated as a tool to formalize this program.

Goodman himself argues to the contrary, in a manner similar to the later Wittgenstein, that our practices, rather than human nature, determine what is an acceptable induction. Goodman feels that general principles of inference are justified by their conformance with particular inferences that are deemed acceptable, and particular inferences are justified by conformance with the general principles. In the closure of the circle, both theory and practice are justified in Goodman's view. Here the MI framework could be incorporated by providing an explicit theory of inference, something lacking in Goodman's exposition.

Goodman, incidentally, is also credited with suggesting that one should try to find "a criterion combining an optimum of simplicity and compatibility" for induction [Kemeny, 1953, p. 408].

8.2.5 BAYESIAN VIEWPOINT

From a Bayesian point of view, all estimation and rational decision making involves an *a priori* probability distribution, which is often subconscious, or implicit, in other forms of assumptions. The view promulgated here, that probability distributions are generally not available, and not even meaningful, in structure estimation, would be dismissed as ingenuous. The Bayesian argues that in designing a description language and information measure one is merely "coding" the prior into an obscure, but mathematically equivalent, form. The fact that the MI estimator is of the same mathematical form as a MAP estimator is used to justify the claim that all estimation is best viewed from a Bayesian perspective. From the form of the MI criterion in an application, the Bayesian will eagerly extract the

implicit *a priori* distribution "in order to help us better understand our assumptions."

In response, we argue that three approaches have been taken in the definition of "probability", and that for many problems, none of the three are compatible with reasonable Bayesian notions of *a priori* distribution:

(1) *Sample frequency*, or the limit of sample frequencies, within the context of repeated sampling from an ensemble, provides a naive *empirical notion* of probability. It is generally agreed that this class of definitions does not to lead to satisfactory results. Fine [1973], among others, discusses the flaws of this approach, such as the fact that unlikely events are perfectly compatible with probability theory, so probability distributions and their sample distributions need never agree. This is so even in the limit, as this could be a probability-zero universe.

(2) A *subjective notion* of expectations, given incomplete information, is advocated by De Finetti [1970] and others as the true meaning of "probability". The subjective approach offers nothing to scientists or mathematicians, as it can, at best, produce a theory isomorphic to that of Kolmogorov [1933], but with much more complex definitions, requiring decision-making entities, objects of value, and explicit bets. However, a general theory of probability must apply in models of the universe which lack these entities, objects and bets.

(3) The accepted mathematical approach to probability theory requires only the formal properties of *additive measures on sets* as axiomatized by Kolmogorov [1933]. In this framework, probability measures are simply functions from sample spaces to real numbers that satisfy his six axioms. The use of such probabilistic models in applications is justified either pragmatically, by previous successes, or intellectually, by the insight which they provide when carrying out an analysis.

As the first two notions above are seriously flawed, *a fortiori* they are inadequate notions of *a priori* probability distribution, and only the Kolmogorov formulation remains. Axiomatically defined *a priori* distributions are commonly employed in Bayesian estimation with enormous justification by both of the criteria above: pragmatic and intellectual. We only argue that other cases exist in

which no *a priori* distribution can be given which is justified by either of these criteria. The case of FSMs in Chapter 4 and rectangular clusters in Chapter 5 are arguably such. In approaching these problems, there was no available probability distribution over the set of FSMs or clusterings. Furthermore, we reaped no insight in examining the induced distribution implied by the natural description languages and information measures. In addition, we see no reason why general information measures should correspond to normalizable probability distributions.

Accordingly, the Bayesian framework is of little benefit in approaching these problems. The advantage of a formal description approach with hierarchically structured descriptions and additive information measures is that it requires one to make preference decisions only at certain points in the structure. These are automatically "scaled up" to the complete structures for which one does not necessarily have probabilistic notions. This point seems especially clear in the case of the image processing example of Chapter 7. One does not design an image processor according to prior notions of the probability that the camera will be aimed at a cat versus a dog. Image complexity is a useful notion, however.

On the other hand, because the information measure we induce on the set of all structures often corresponds to some probability distribution, we can take advantage of it if we wish. The methodology of designing a description language and information measure can be used to produce a function which satisfies the axioms of a probability distribution if it is required for other purposes and none is available otherwise.

8.2.6 CLASSICAL ESTIMATION THEORY VIEWPOINT

Compared to classical estimation theory, the proposed framework is more powerful in that it does not require that the estimated structure be identified with a point in a vector space, but is less powerful in that it deals only with countable sets of structures and so can not incorporate real numbers. For problems which are well modelled in the vector space framework, classical techniques should be used. They take advantage of the additional algebraic structure available, and do not require that the issue of complexity be addressed. Furthermore, for certain problems with differentiable families of probability distributions, classical criteria result in computationally tractable optimization problems.

An extended class of problems exists in which real numbers are involved and structural variability is an issue. For this class of problems, neither the classical nor proposed framework is sufficient, and a combined framework is desired. It is not clear how this might be performed satisfactorily however. The essential problem is that in the entropic sense, a real number contains an infinite amount of information relative to an integer, or other object, selected from a countable set. How to combine objects of these two types within an additive framework of information is a difficult problem for future exploration.

In relation to classical estimation theory, this framework suffers from a noticeable lack of provable properties, such as asymptotic consistency results. While such results are often interesting when available, we feel it is more important at this stage to understand the more fundamental issues concerning structure estimation. We have tried to present a practical framework which can be fruitfully applied to real problems with small amounts of data.

8.2.7 ALGORITHMIC INFORMATION THEORY VIEWPOINT

Solomonoff [1964] gives a theory of induction which can be interpreted as incorporating MI estimates of Turing Machine programs which generate the input data. Although there is a great appeal to AIT owing to the universality and intertranslatability of Universal Turing Machines, there are also several problems with the method relative to our structural estimation framework. We see our framework as a method for dealing with the three major problems of AIT: (1) algorithmic information is not computable, (2) Turing Machine programs are rarely in the class of structures of interest, and (3) AIT is only meaningful asymptotically.

The first problem is that the minimization required required by AIT is not computable—no algorithm can determine the shortest length Turing Machine program which generates a given string. In practice, this is not quantitatively different from any of the estimators presented in Chapters 4–7. Due to combinatorial difficulties, we can not determine the minimum length estimate on current computers in our lifetimes. The only difference is that we can prove algorithmic information can not be computed, while we only suspect there is no practical way to determine most MI estimates. In either case, we are likely to use approximation techniques and settle for an approximate minimum. Note that local search techniques are not likely to find reasonable minima in the class of Turing Machine programs, due to the extreme discontinuity of the space. Solomonoff [1986] is persuing methods for more effective enumerations.

The second problem with an algorithmic measure of information is that we are usually not interested in estimates of structure in the form of Turing Machine algorithms. General algorithms are rarely of interest as aids in understanding data. If an algorithm is a useful model for data, it is always organized by the hierarchical

principles of "structured programming." But more generally, we usually wish to estimate a structure in a very specific class of formal objects. If we are interested in estimates in the class of finite-state models for example, as in Section 1.4, there must be some purpose for which the estimate is intended. That purpose might require the best finite-state model we can derive, even if some more general Turing Machine model allows a more concise description of the input. For example, if we are estimating the control mechanism of an electrical appliance so that we can "reverse engineer" a similar appliance, and our engineering expertise limits us to synthesizing finite-state controllers, we prefer a complex finite-state structure to a simple Turing Machine model.

The third problem of AIT is that it is meaningless for finite inputs, and only becomes useful if the length of the input grows asymptotically infinite. However, we need techniques to deal with many applications in which a good estimate must be made from a fixed, finite amount of data. The formal description approach is designed to do just this.

Minimum Information estimators can be seen as defining a variant of algorithmic information in which the set of UTMs is replaced with less versatile mappings which are more appropriate to our means and goals. Conversely, algorithmic information is the special case of our information measures that result when the interpretation function is the mapping induced by a particular UTM, and the language is the set of binary strings. Note, finally, that Solomonoff's [1964] comments on the validation of his induction scheme—that it is based primarily on intuitive appeal and several case studies—are also apropos to our framework.

8.3 Future Directions

There are many directions in which this research can proceed. On theoretical and conceptual fronts, it seems important that the framework be extended to include real numbers. Extensions to metric spaces would also be of interest. In terms of applications, there are a great many problems which can be posed as structure estimation, which seem ripe for this framework. Some obvious applications are listed below.

8.3.1 REAL NUMBERS AND SCALING

As mentioned above, it would be of interest to derive a general method for incorporating real numbers in the structure estimation process, both as inputs and as outputs. One approach used by Wallace and Boulton [1968] and Rissanen [1978, 1983], is to consider truncating real numbers to fixed levels of precision, thereby reducing the uncountably infinite set of reals to a countable set of approximations. They are able to differentiate with respect to the precision level in order to obtain an "optimal" level of precision for minimizing description length. This approach has a number of problems which need to be addressed in a more general investigation.

An obvious problem is that a single uniform level of truncation is an arbitrary choice of description and generally insufficient. Some terms and inputs may warrant more detailed precision than others, and a versatile language for declaring and using truncations might be worth designing. A more subtle problem comes about because the methods do not result in estimators which are invariant under scaling and translation of the input space. As there are an infinite number of truncated reals, they can not all be given equal-length binary codes. When the

data is translated or scaled, it falls into a portions of the reals in which the relative lengths of the codes differ, and so varying estimates result. Rissanen [1983] develops a criterion for ARMA models superior to his [1978] model, in that it is invariant under scaling. It is not invariant under translation, however.

In many applications where data is naturally described with real numbers, scale invariance is a desideratum. This is true of the clustering, segmentation, and vision examples. Because we separate the description of an object from its information measure, we can allow nonentropic information measures on real numbers. Incorporating this idea into estimators which are invariant under linear transformations is a topic for future investigation.

8.3.2 METRIC SPACES

Although spaces of trees, graphs, sets or other formal structures lack the general algebraic properties of vector spaces, natural distance measures often exist which satisfy the minimal requirements of a metric space. In a metric space, a distance measure, $d(x,y)$, is defined on pairs of objects, and satisfies three properties:

$$d(x,y) \geq 0, \quad \text{and } d(x,y) = 0 \text{ iff } x = y$$
$$d(x,y) = d(y,x)$$
$$d(x,y) \leq d(x,z) + d(z,y)$$

As an example, the distance between two directed labeled graphs can be measured as the minimum number of arcs and nodes which must be inserted, deleted, or re-labeled, in order to change one graph into the other. It is easy to see that this measure satisfies the three properties. To show the triangle inequality, (3), note that $d(x,y)$ must be less than $d(x,z)+d(z,y)$, because the insertions, deletions and relabelings required to convert x to y can always be made by first transforming x

to z, and then transforming z to y; this can not result in fewer steps than $d(x,y)$ which is defined to be the minimum.

Such metrics are sometimes natural measures of distance which can be used to measure the size of the difference between a true structure and an estimated structure. Estimators can be designed to minimize the expected value of this distance, and estimator performance can be gauged in those terms. A further avenue to explore is the derivation of general bounds on estimation error, analogous to the Cramer-Rao or Barankin bounds, but not requiring their assumption that the estimator be unbiased. (Because metrics are non-negative, the only estimator with zero bias in these terms is the ideal estimator which always estimates the correct structure. Such an estimator would not exist in any interesting structure estimation problem.)

8.3.3 POLYNOMIAL ORDER

A classic problem of estimation theory and practical statistics is how to select the appropriate order of a polynomial to fit to data. This is a special case of the polynomial segmentation case of Chapter 6 in which we restrict the model to one segment. As discussed there, it is natural to associate k^{th} order polynomials over an interval with $k+1$ ordered pairs in the space. When combined with a Gaussian probabilistic error model, this gives a model order criterion analogous to the Akaike Information Criterion or Rissanen's MDL criterion. Because of the widespread use of least-square polynomial fits, it would be interesting to pursue this example and apply it to real data.

8.3.4 SPEECH PROCESSING

A natural generalization of the segmentation problem of Chapter 6 is the problem of segmenting speech into phonetic or phonemic units. The piecewise constant model must be expanded to include the types of formant structures typical of natural language. We do not expect to recognize speech based on the *shape* of waveforms. Instead, the amplitude spectrum of the signal appears to contain the relevant information, and we are not suggesting that the design of a grammar for these structures is a trivial exercise. Rather than a single z-value for each sample time, the input might be the frequency and amplitude of the major spectral peaks at each time, or the complete discrete Fourier transform at that time. Although this approach is likely to be more complex than current ad hoc approaches to speech segmentation, it is also likely to be much more tolerant to background noise, if analogy from the one and two-dimensional segmentations of chapters 6 and 7 is a valid guide. With an appropriately flexible grammar, it may also result in greater speaker independence than other techniques. As noise and speaker independence are currently major impediments to speech processing, the MI approach seems worthy of further investigation.

8.3.5 MUSIC

Another field in which segmented models are natural is that of computerized music analysis. The problem of automatically generating a score given a recording of a musical piece is fundamentally one of generating a segmented description. When only a single instrument is involved, and the signal to noise ratio is high, the problem is relatively straightforward, and various ad hoc methods have been implemented. We expect that an MI approach might be fruitful for the analysis

of ensemble music pieces (e.g., duets, trios, etc.) and in situations with significant background noise. An estimate of the meter and time signature could provide an information measure which "encouraged" notes to appear on the appropriate division points within each measure. Definitions of themes would allow repetitions with variations to be concisely described. The interesting complication in the case of ensemble pieces is that a single partition of the time axis is insufficient; a separate segmentation is required for each instrument, but the segment borders would not be completely independent of each other, as different instruments often begin notes simultaneously.

8.3.6 DETECTING CHANGES IN DYNAMIC SYSTEMS

As a final example of a one-dimensional segmentation problem, consider the problem of detecting changes in the characteristics of a dynamic system being monitored. In order to detect possible failures in the plant or in the sensors, it should be possible to model the overall system as distinct systems over separate time segments. With the appropriate description languages and information measures, the MI description of the observations should be segmented into time periods in which the system characteristics remained constant.

8.3.7 MACHINE VISION

The sample image processing problem of Chapter 7 relies on a very simple class of models. Few applications are sufficiently well described with a rectangular vocabulary of images. Section 7.5 discussed several extensions of the method, such as incorporating general polygons (with edges not necessarily orthogonal to the grid) and simple arc types. The MI approach is natural for reducing the bandwidth

of video images for low baud-rate channels, in applications where simple image approximations are adequate.

At the other extreme, very detailed classes of models, tailored to specific types of image components, might be designed for specific applications. The analysis of cloud-chamber photographs could be performed by means of a grammar designed for various types of particle traces. Optical character recognition can be performed with a grammar for describing text characters. Although these problems have been approached with syntactic pattern recognition methods, the MI approach, because it measures the fit between the models and the data, should result in improved performance.

8.3.8 ARCHITECTURAL PROBLEMS

An interesting class of problems involving complex hierarchical structures concern architectural structures. Buildings are composed of levels and wings, which are composed of suites and rooms, which are bounded by walls, doors, windows, which are described in terms of surfaces, edges and points. Estimation problems over this space of structures involve "parsing" of data into architecturally meaningful units. For example, it would be useful if a computer aided design system could understand the architectural relations between surfaces and lines given by a user, so that commands such as *make the walls in this room green* could be executed without requiring the designer to explicitly define which surfaces are walls.

Another application would involve estimation of architectural style. It would be an interesting exercise to design an estimator which, given a data-base of solids, surfaces, and/or lines defining an architectural structure, outputs the style of

the structure, e.g., *Gothic*. Presumably, the MI approach to this problem would involve a formal language defining structures in a range of styles. A description in the appropriate style should be more concise than a description in terms of the other styles.

8.3.9 MACHINE LEARNING

The learning of structurally defined terms from examples, as in Winston [1975], is naturally formulated as a structure estimation problem. Winston gives methods by which simple structures of blocks, appropriately labeled *arch* or *not arch*, are used to form a description of the structural relations which characterize arches. In the MI framework, we generate an explicit criterion for selecting such a characterization of arches. We simply seek a joint description of all the examples which incorporates a definition of *arch*. With an appropriate description language, the single definition of *arch* will contain the maximum amount of common information in the descriptions of the particular arches. This allows redundancy in the descriptions of the examples to be reduced. The MI estimate of the definition of *arch* will only contain the minimum amount of "forbidden features" necessary to indicate that the *non-arches* are not arches.

Another approach to machine learning which seems well suited to the MI formulation is the *conceptual clustering* of Michalski [1983]. This is a general clustering technique based on propositional logic descriptions of data (rather than coordinate values). An MI approach analogous to that of Chapter 5, but using descriptions in the language of predicate logic rather than coordinate descriptions, appears to be a directly applicable criterion for this example. Certain transformations which Michalski proposes for modifying one logical description into another

(e.g. replacing a constant with a universally quantified variable) would also be natural in the local search techniques to optimize the MI criterion.

A third machine learning problem is to produce adaptive structure estimators. As we have presented the framework, the language and information measure incorporates only *a priori* information given by the system designer. In certain contexts, labelled training sequences are available after the system is designed, and an adaptive system is required. The language and/or information measure may be modified as a function of the training sequence. This is naturally accomodated in our framework with a higher-level structure estimator, which takes the training sequence as input and estimates a structure estimator to be used on future, unlabelled, data.

8.3.10 MAN/MACHINE INTERFACES

A final future application is that of effective man/machine interfaces. One example from this class was suggested for the architectural design system above, in which the system "understands" which surfaces are *walls* based upon structure estimation rather than explicit definitions. In general, many manually controlled computer operations can be seen to be part of a larger structure which in some sense characterizes the relations between the manual operations.

As a not so hypothetical example, a user might control a word processor, through a sequence of simple commands, to capitalize the word *chapter* in each of a dozen separate text files. If the interface to the word processor could detect the fact that the operator repeatedly opens a file, searches for the word, changes it, and closes the file, it could automate many of the steps for the user. Detecting the larger sequence of operator actions amounts, in this case, to estimating the

structure of a FSM, as in Chapter 4, in which the arc labels are word-processor commands given by the operator. If this structure were detected in the process of modifying the first several files, the system could continue the procedure in each of the other files with minimum explicit control. More generally, the structure of sequences of computer commands can be estimated, resulting in a form of automatic programming.

Appendix

The Nonintrusive Appliance Load Monitor

The notion of structural estimation, and several of the case studies in this thesis were suggested by a problem concerning residential electric appliance energy monitoring. The problem is develop a nonintrusive appliance load monitor which can be installed in the kilowatt-hour meter socket of a home to estimate the nature and energy consumption of the major appliances which constitute the load. This load monitor is to be a microprocessor controlled unit, with current and voltage sensors, programmed with an algorithm to analyze the total load and estimate its makeup. It is given no access to information internal to the residence. No internal circuits are separately monitored, and no appliance survey is carried out beforehand. The problem is clearly one of structure estimation, as the number of appliances must be determined. Furthermore, each appliance can be modelled as a finite-state machine of unknown structure which describes its operating states and electrical characteristics. The overall problem can in fact be broken down into several structure estimation subproblems.

A summary of the problem and an *ad hoc* solution is given in Hart [1985b]. A more detailed description can be found in Hart [1985a]. The most salient characteristic of this problem is that it is severely underdetermined. The appliance mix and their operating histories combine to determine the total load in a well defined way, but given only the total load, an infinite number of different appliance mixes and histories could explain it equally well. The complexity/fit tradeoff comes about because most estimates of the appliances and their operating periods will result in some residual error relative to the input data. This discrepancy can always be reduced by postulating a small appliance and fitting its operation to the residual.

As input to the estimator, we have current and voltage measurements for the two 120 V legs at the service entrance to the home. Typical residential power systems are three terminal networks as indicated in Figure A.1. Two out-of-phase service legs supply the 120 V circuits in the residence, and are assigned to branch circuits effectively at random. The 240 V appliances are wired from one 120 V leg to the other. The raw input is then a four dimensional function of time: current and voltage waveforms for each of the two legs. This is assumed to be converted into real and reactive power measurements, sampled at regular intervals, and quantized by an A/D converter for computer processing.

The output of the estimator consists of a set of appliance descriptions, and a description of their operating history. Appliances are described by a finite-state machine which indicates their operating states and allowed sequences of states. Figure A.2 indicates several common appliance FSM configurations. Most appliances are well described as a two-state (ON/OFF) FSM. More complex appliances,

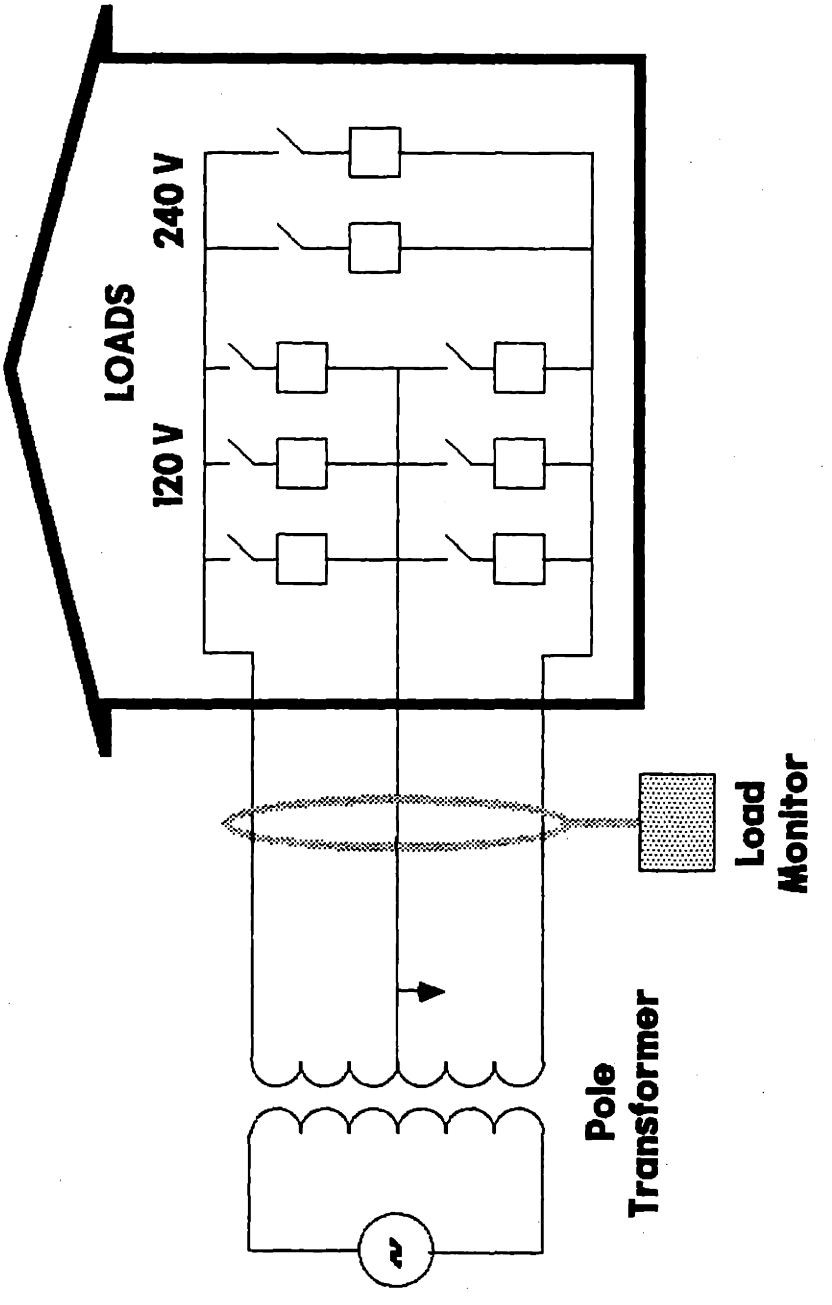
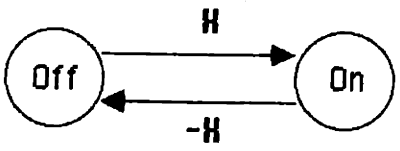
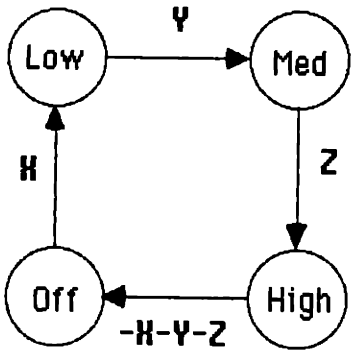


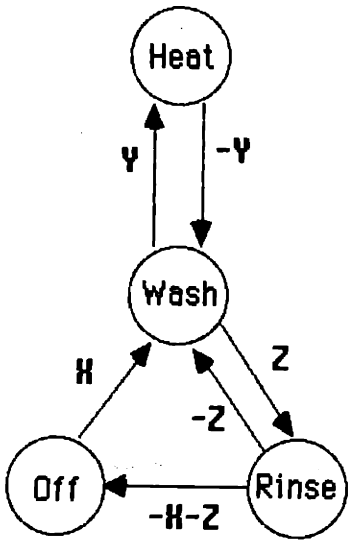
Figure A.1 Load Monitor Sensor Installation



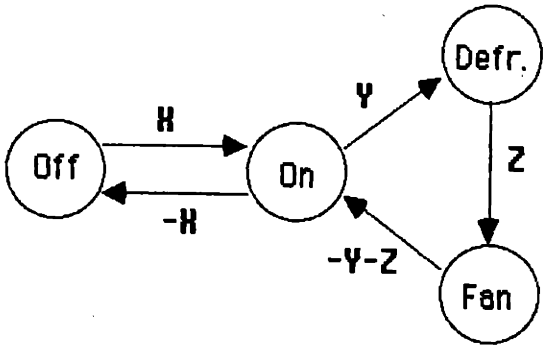
Two-State Appliance



3-Way Lamp



Dishwasher



Frostless Refrigerator

Figure A.2 Finite State Models for Appliances

such as dishwashers and washing machines, are characterized by finite-state controllers which cycle them through various states, e.g., *wash*, *rinse*, and *dry*. The refrigerator diagram of Figure A.2 indicates a defrost cycle in which the appliance occasionally departs from its typical two-state behavior.

To a first approximation, the observables labeling the arcs in these FSMs are the changes in complex power which are observed in the total load when the appliance makes the corresponding state transition. For example, when a 300 W refrigerator turns on, the total load increases, from a level which depends on the states of the other appliances, by 300 W, along with some characteristic quantity of reactive power. Power measurements are not the most consistent input variable however, as the electric utility does not provide a constant line voltage. The actual power change will vary $\pm 20\%$ as the line voltage varies $\pm 10\%$ around the nominal 120 V. To eliminate this variation, we calculate admittance measurements for the two legs of the house, which are fairly constant for each state transition. This gives a four-dimensional observation space consisting of the conductance and susceptance on each of the two legs. Admittance changes are additive because appliances are wired in parallel. They are voltage independent to the extent that the current/voltage relations for the electronic components of each appliance are linear.

For two-state appliances, the ON transition is the negative of the OFF transition, and this provides a strong constraint which can be exploited by an *ad hoc* estimator. An algorithm for appliance estimation using two-state models has been implemented and field tested with good results, as described in the references cited above. When the monitored load contains more complex appliances, this "two-state algorithm" either ignores them or decomposes them into several two-state

components, such as the heater and motor elements of a dishwasher. Our goal here is to consider the case in which the load is modelled with a set of FSMs which are not restricted to two states each.

We find it insightful to look at the problem from the perspective of communication theory. The load monitor can be viewed as a receiver for decoding additive signals on a *Multiple-Access Channel* (MAC). A multiple access channel is a medium which can be analyzed as several independent logical communication channels involving several transmitters. Each appliance state change can be interpreted as generating a message, $Y_{i,j}$, that appliance i changed state to j. The messages are ideally step functions in admittance of height $Y_{i,j}$, occurring at the time of the state change. They are summed, with noise, to form the input. In practice, the step functions also contain a somewhat inconsistent initial transient component, especially in large motors, which might be described in more detailed models.

A useful constraint, which reduces the set of possible structures, is that the loop-sum of these messages must be zero around any cycle of states in an appliance model.

$$Y_{i,j} + Y_{j,k} + \ldots + Y_{l,i} = 0$$

where $i, j, k, \ldots, l, i$ is a cycle of states. Fortunately, arbitrary constraints such as this are relatively easy to insert into local optimization techniques.

The channel is simply the house wiring, which has a number of desirable properties. It is relatively short, and features good signal to noise ratio in the low frequency range of interest. Furthermore, the transmission powers are relatively high, e.g., a 4 KW signal is present for several minutes to encode the one bit of information that the hot water heater turned on. Another pleasant feature from

the communication point of view is the low message rate compared to the channel capacity. Typical average rates found in field tests range from twenty to thirty messages (appliance state changes) per hour, peaking around twenty per minute during times of heavy appliance usage.

Balanced against these favorable conditions are several factors which make for a poor communication system. The most serious of these is that we must design a receiver with no detailed knowledge of the code table. We do not know in advance what the messages will be, or how to interpret those messages we do find. For example, there is a certain similarity between the electrical characteristics of most refrigerators, which results from the economics of appliance engineering and marketing, but there is still a wide range of variation within this class. This requires an adaptive receiver which partitions its decision space as a function of initial observations of the channel. To do this, the receiver must bring a great deal of *a priori* information into the decoding process.

Another problem in designing such a receiver is the presence of ambiguous codes. Different two-state appliances may coincidentally operate at the same power levels, and different multi-state appliances might include step changes of the same size. In the case of two-state appliances, ambiguous messages can not be distinguished, but in the case of multi-state FSMs, they may. As different FSMs which incorporate the same message may differ in the allowed messages preceding or following the ambiguous messages, state-space-based decoders can use the surrounding context for disambiguation. The Viterbi algorithm seems quite relevant here (Viterbi [1967], Fornay [1973]).

A final problem to be faced is the presence of simultaneous messages. When two appliances transitions occur simultaneously, or are separated by less than the

sampling time interval, the combined message which is received is the sum of their separate messages. In more controlled MAC situations, a variety of contention resolution mechanisms may be designed into the transmitters. Here we must deal with appliances of standard manufacture. Our only options are decoding techniques that rely on the additivity of the channel. If a received message can be recognized as a simultaneous multiple transmission, the decoding can take the form of a combinatorial search through pairs (or sets) of known messages which sum to the given message.

Following a divide and conquer approach, an algorithm for this receiver can be constructed out of three separate structure estimators. This will not be a globally optimal breakdown of the problem, but it separates the overall problem into manageable portions.

The first problem is to segment the input waveform into periods in which appliances are not changing state, separated by times at which appliances do change state. This is a waveform segmentation problem which could be attacked by the method of Chapter 6. The model considered there, in which the segments are each constant functions of time, is an excellent first approximation to actual data. It needs to be augmented slightly to allow short periods described as "transients" at the beginning of each constant period. This could take the form of allowing a more concise description of large deviations from the constant level for the initial seconds of each segment. To be practical, a segmentation algorithm of this kind would have to operate recursively, based only on a finite moving window of data, and a finite amount of memory. This might be designed by introducing slight approximations to the MI criterion.

The second structure estimation problem is one of clustering the magnitudes of the step changes into groups corresponding to appliance transitions. There is a certain amount of variation amongst the observations associated with each particular state-change. This may result from a variety of sources, including nonlinear current/voltage relations, varying start-up conditions, effects of very small or continuously variable appliances, and sampling noise. Measurements show however, that observations attributable to a single appliance state change tend to cluster relatively tightly in the four-dimensional observation space. The main problem in clustering is the structural issue of selecting the number of clusters. The approach developed in Chapter 5 might be adapted to the types of variation typical of load data.

The final structural estimation problem is the most difficult one, of estimating a set of FSMs which model the appliances generating the load. As indicated in Chapter 4, this is a very complex problem which we have not explored in depth. In the context of this application, it is not clear what range of appliance structures might be found, or how detailed an analysis is required for utility load gathering purposes.

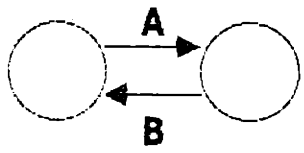
Based on the methods of Chapter 4, we present two examples of FSM estimation demonstrating two difficult aspects of the problem. In the first example, we estimate three simple appliances with a great deal of overlap in their alphabets. Figure A.3 shows three appliance models, the transition sequences they generate independently, and a result of interleaving their operation. To the algorithm of Chapter 4, we now add the zero loop-sum constraint. The constraints indicated would result from examining the positions of the clusters in the complex-power space. As shown in Figure A.4, the algorithm reconstructs this set of FSMs given

the combined data. With only the thirteen observations indicated, these three FSMs are one of the best possible estimates; after several more observations, this becomes the best estimate.

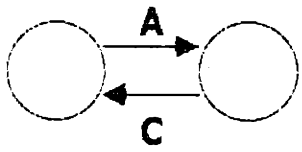
In the second example, we estimate the structure of a more complex appliance, the simplified dishwasher model of Figure A.2 (in which the *dry* state is not used). Again, the loop-sum constraint is used. Figure A.5 shows the observation, which corresponds to two "cycles" of the machine, and the estimated structure, which is correct. The second choice structure, according to the information criterion used, is shown in Figure A.6. Here the heating element has been separated out into a second appliance. This second choice will be evaluated as increasingly poor as more observations arrive, because it does not capture the fact that *EF* always is observed as a pair, preceded and followed by only certain other observations.

One important aspect of this problem is that noise models must be incorporated. The methods above assume perfect observations, but we can not expect perfect segmentation or clustering from the other subportions of the overall problem. This will require an estimator which does not fail in the presence of occasional misidentified transitions. A number of methods for describing noise transitions in the context of normal FSM routing come to mind. Routes may be allowed to restart at specifiable states, unexpected observations may be inserted in the observation stream, or a list of exceptions may be included. These methods need to be investigated, and a language selected which allows exceptions at an appropriate information cost. We have not considered these problems in detail, but hope to in the future.

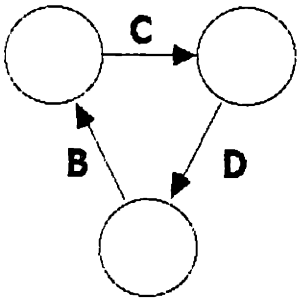
We should, in closing, point out a general concern regarding estimation in general and structure estimation in particular. There is a very real ethical issue



A B A B



A C A C A



C D B C

Combined: C A A C B A A D B C B A C

Constraints:

A+B=0
A+C=0
B+C+D=0

Figure A.3 Synthetic Data for Three Simple Appliances

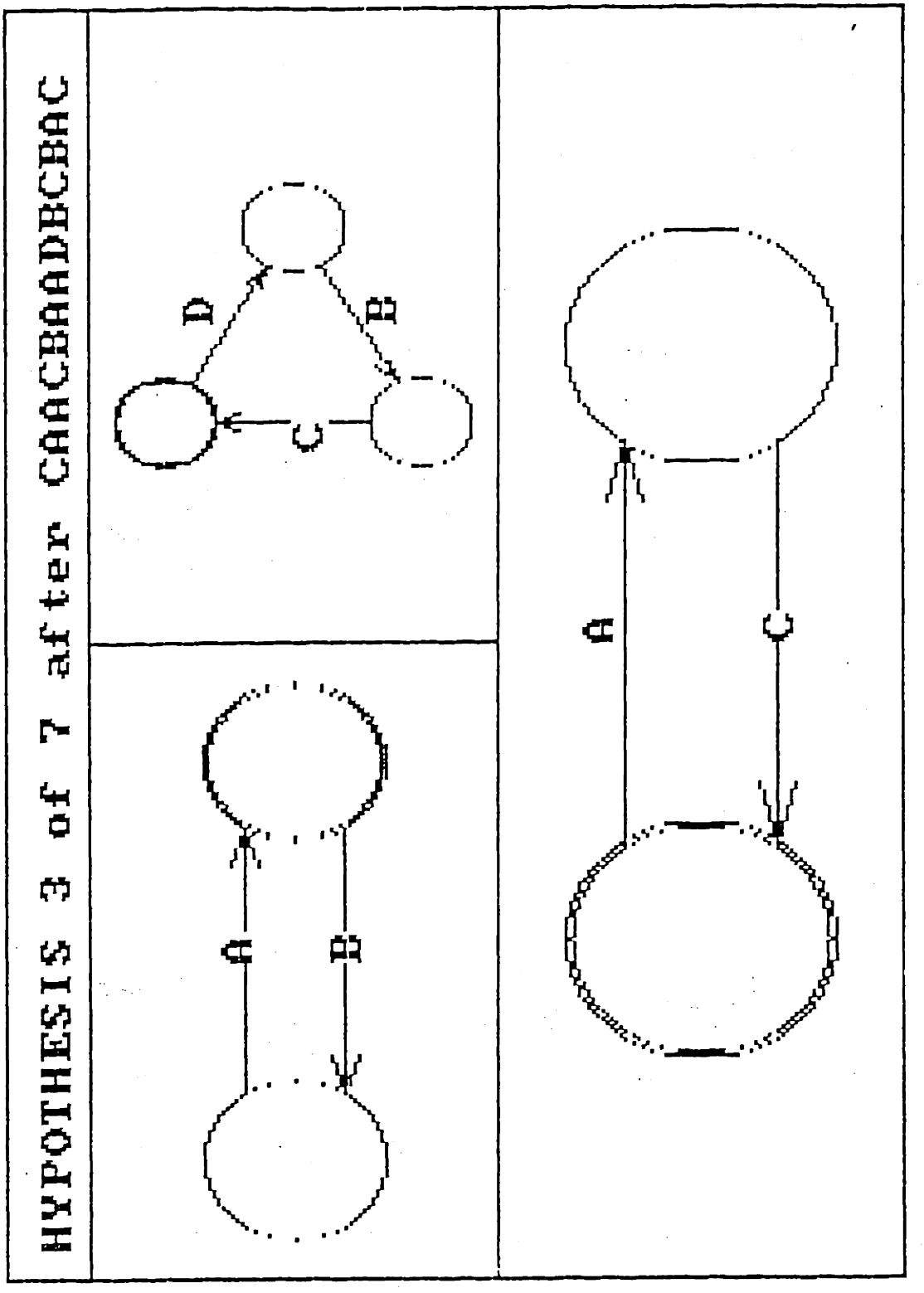


Figure A.4 Three Appliances Estimated, Given Observation CAACBAAD-BCBAC

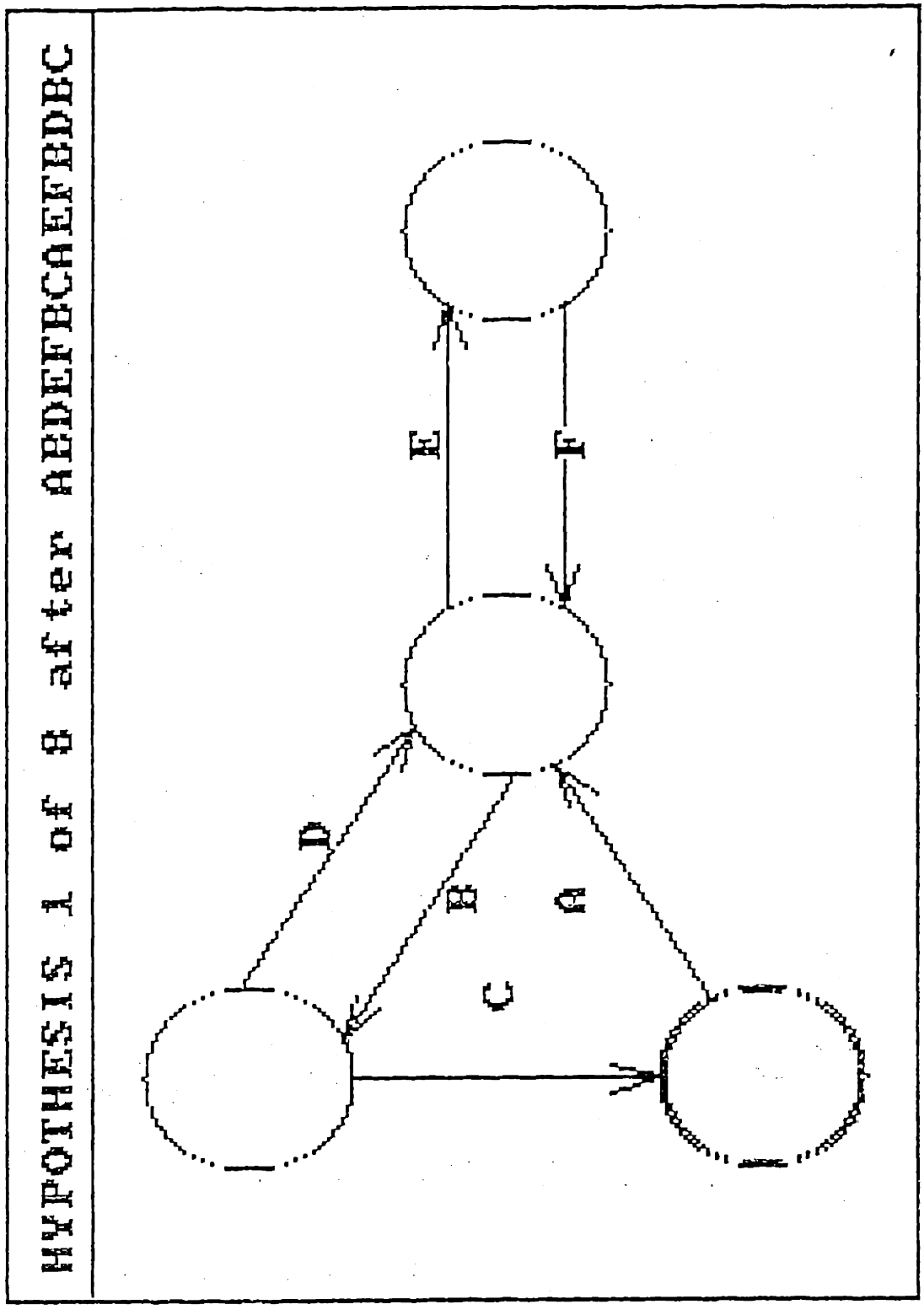


Figure A.5 Appliance Estimate for Simple Dishwasher Example, Given Data ABDEFBCAEFBDBC

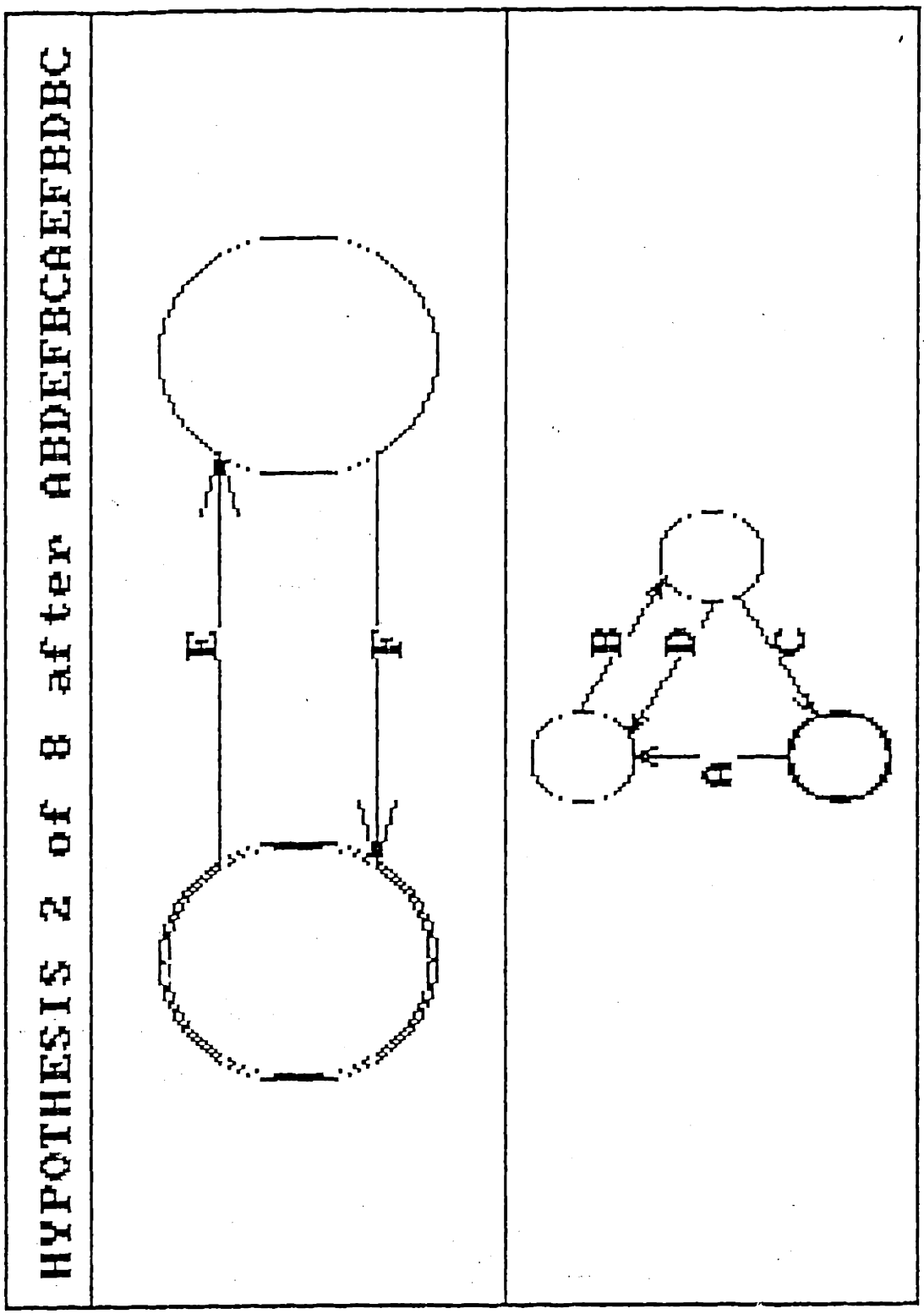


Figure A.6 Second Choice Estimate for Simple Dishwasher Example, Given Data ABDEFBCAEFBDBC

concerning the misuse of estimation techniques. By developing new estimation techniques, we are opening up new domains for misuse. A nonintrusive load monitor will certainly be seen as an invasion of privacy by many who are concerned with the general trend of increasing surveillance in our society. Marx [1984] points out some of the many aspects of this trend. The nonintrusive monitoring technique "sees" directly into the domain of the private residence which has traditionally been protected from public scrutiny. Data collected by utility personnel could be used, for example, by law enforcement agencies to check alibis of suspects, or by burglars to determine occupancy patterns. Another form of invasion would be for government agencies to monitor for illegal appliance activity, such as photocopiers in totalitarian countries.

We hasten to point out that electric power planners and analysts have a great and legitimate need for the data which can be collected with a nonintrusive load monitor. Energy planning is an important consideration in our society, and laboratory measurements are not sufficiently representative of typical appliance usage. Field data is essential for understanding the contributions of individual appliances to the aggregate load as a function of time and temperature. This, in turn, is necessary information for planning future generation and transmission capacity, for predicting the economic consequences of alternative rate schedules, and for understanding the effects of novel appliance constructions.

Typically, appliance load data is very expensive to collect in the field, as it necessitates entrance into a home for sensor and wiring installation, maintenance, data retrieval, and eventual sensor removal. Utility presence may affect the resident's energy consumption patterns, and it makes utilities liable for incidental damages. A nonintrusive technique alleviates these problems, and allows more

comprehensive data samples to be collected with the utility's resources. It could also be of great benefit to energy conscious residents trying to understand their own consumption patterns.

It is difficult to balance these benefits against the small, but very real possibility for abuse of the method. (Cynically, we would be surprised if it is not well known to government agencies for monitoring foreign embassies, but we have no evidence on this point.) It is possible to purposely defeat nonintrusive monitoring by charging and discharging an energy storage device to generate random step functions at short random intervals. However, this type of "jamming" is detectable in itself, and indicates something to those doing the monitoring.

This leads us to consider the role of science in society, and the responsibility of the scientist. Sinsheimer [1978] and Graham [1978] are helpful discussions in this regard. We feel that each case must be judged on its own merits, and the balance of the evidence in this case is in favor of developing the nonintrusive appliance load monitor. Its use should be carefully regulated however, and collected data should be stored in a format in which it is not tagged with its source. Hart [1986] considers these matters in greater detail.

BIBLIOGRAPHY

H. Akaike (1974) *A New Look at Statistical Model Identification*, IEEE-TR-AC-19, pp. 716–723.

— (1981) *Modern Development of Statistical Methods*, Ch. 6 of P. Eykhoff (ed.) Trends and Progress in System Identification, Permagon.

M.R. Anderberg (1973) *Cluster Analysis for Applications*, Academic Pr.

D. Angluin (1978) *On the complexity of minimum inference of Regular Sets*, Inf. Control 39, pp. 337–350.

D. Angluin and C.H. Smith (1983) *Inductive Inference: Theory and Methods*, Computing Surveys, Vol. 15, No. 3, pp. 237–269.

G. Bateson (1972) *Steps to an Ecology of Mind*, Chandler.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss, *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*, Annals Math. Stat 41, No. 1, pp. 164–171.

T. Bayes (1763) *An Essay Towards Solving a Problem in the Doctrine of Chances*, Phil. Tr. 53, pp. 370–418.

A.W. Biermann and J.A. Feldman (1972) *A Survey of Results in Grammatical Inference*, in S. Watanabe (ed.) *Frontiers of Pattern Recognition*, Academic Pr., pp. 31–54.

— (1972) *On the Synthesis of Finite-State Machines from Samples of Their Behavior*, IEEE-TR-C-21, pp. 592–597.

Borland International Inc. (1984) *Turbo Pascal Version 2.0 Reference Manual*, 4113 Scotts Valley Dr., Scotts Valley, CA 95066.

D.M. Boulton and C.S.Wallace (1973) *Information Measure for Hierarchic Classifications*, Computer Journal 16, No. 3, pp. 254–261.

— (1975) *Information Measure for Single-Link Classification*, Computer Journal 18, No. 3, pp. 236–238.

G.J. Chaitin (1966) *On the Length of Programs for Computing Finite Binary Sequences*, J. ACM 13, p. 547.

— (1977) *Algorithmic Information Theory*, IBM J. R&D 21, pp. 350–359, 496.

N. Chomsky (1956) *Three Models for the Description of Language*, IRE-TR-IT-2, pp. 113–124, reprinted with corrections in R.D. Luce, R. Bush and E. Galanter (eds.), *Readings in Mathematical Psychology*, Vol. II, Wiley.

C.M. Cook, A. Rosenfeld, and A.R. Aronson (1976) *Grammatical Inference by Hill Climbing*, Info. Sci. 10, pp. 59–80.

L. Davis (1985), *Applying Adaptive Algorithms to Epistatic Domains*, IJCAI, pp. 162–164.

B. De Finetti (1970) *Theory of Probability*, Volumes I and II, Wiley.

A.P. Dempster, N.M. Laird, and D.B. Rubin, (1977) *Maximum Likelihood from Incomplete Data via the EM Algorithm*, J. Royal Stat. Soc. (B), V. 39, No. 1, p. 1.

R.O Duda and P.E. Hart (1973) *Pattern Classification and Scene Analysis*, Wiley.

P. Elias (1975) *Universal Codeword Sets and Representations of the integers*, IEEE-TR-IT-21, No. 2, pp. 194–203.

M. Feder (1986) *Maximum Entropy as a Special Case of the Minimum Description Length Criterion*, IEEE-TR-IT-32, No. 6, pp. 847–849.

— (1987) *Personal Communication concerning MIT PhD dissertation in progress.*

T. Fine (1973) *Theories of Probability*, Academic Pr.

R.A. Fisher (1925) *Theory of Statistical Estimation*, Proc. Camb. Phil. Soc. 22, pp. 700–725, and in *Contributions to Mathematical Statistics*, (1950), Wiley.

— (1936) *The Use of Multiple Measurements in Taxonomic Problems*, Annals of Eugenics 7, pp. 179-188.

— (1958) *On Grouping For Maximum Homogeneity*, J. Am. Stat. Assoc 53, pp. 789–798.

J. Fodor (1975) *The Language of Thought*, Academic Pr.

G.D. Forney (1973) *The Viterbi Algorithm*, Proc. IEEE 61, No. 3, pp. 268–278.

K.S. Fu (1975), *Syntactic Methods in Pattern Recognition*, Academic Pr.

— (1982), *Syntactic Pattern Recognition and Applications*, Prentice Hall.

— (1983) *A Step Towards Unification of Syntactic and Statistical Pattern Recognition*, IEEE-TR-PAMI-5, No. 2, pp. 200–205.

K.S. Fu and T.L. Booth (1975) *Grammatical Inference: Introduction and Survey— Part I*, IEEE-TR-SMC-5, No. 1, pp. 95-111, — *Part II*, Vol. SMC-5, No. 4, pp. 409–423.

B.R. Gaines (1976) *Behaviour/Structure Transformations under Uncertainty*, Int. J. Man-Machine Stud. 8, pp. 337–365.

R. Gallager (1968) *Information Theory and Reliable Communication*, Wiley.

M. Garey and D Johnson (1979) *Computers and Intractibility*, Freeman.

M.P. Georgeff and C.S. Wallace (1984) *A General Selection Criterion for Inductive Inference*, in T. O'Shea (ed.) *Advances in Artificial Intelligence*, Elsevier, pp. 473–482.

E.M. Gold (1967) *Language Identification in the Limit*, Information and Control, Vol. 10, pp. 447–474.

— (1978) *Complexity of Automaton Identification from Given Data*, Inf. Control 37, pp. 302–320.

N. Goodman (1954) *Fact, Fiction and Forecast*, Harvard University Press.

L. Graham (1978) *Concerns about Science and Attempts to Regulate Inquiry*, Proceedings of American Academy of Arts and Sciences, *(Daedalus)*, V. 107, No. 2, pp. 1–21.

U. Grenander (1976, 78, 81) *Lectures in Pattern Theory. Vol. I: Pattern Synthesis, Vol. II: Pattern Analysis, Vol. III: Regular Structures*, Springer Verlag.

G. Hart (1985a) *Nonintrusive Appliance Load Data Acquisition*, Section 40 of *Proceedings: International Load Management Conference*, Electric Power Research Institute Report #EM-4643.

— (1985b) *Prototype Nonintrusive Appliance Load Monitor*, MIT Energy Laboratory Technical Report.

— (1986) *Computerized Surveillance via Utility Power Flows*, Unpublished manuscript.

J.A. Hartigan (1975) *Clustering Algorithms*, Wiley.

R.V.L. Hartley (1928) *Transmission of Information*, BSTJ 7, pp. 535–563.

F. Hennie (1968) *Finite State Models for Logical Machines*, Wiley.

J.H. Holland *Adaption in Natural and Artificial Systems*, Univ. Michigan Pr.

J.E. Hopcroft and J.D. Ullman (1969) *Formal Languages and their Relation to Automata*, Addison-Wesley.

— (1979) *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley.

J.J. Horning (1969) *A Study of Grammatical Inference*, Technical Report No. CS139, Comp. Sci. Dept., Stanford Univ., August 1969.

D. Hume (1748) *An Enquiry Concerning Human Understanding.*

E.T. Jaynes (1982) *On the Rationale of Maximum-Entropy Methods* Proc. IEEE 70, No. 9, pp. 939–952.

B. Julez (1969) *Pattern Discrimination*, in W. Reichardt (ed.) *Processing of Optical Data by Organizms and by Machines*, Academic Press, pp. 580–588.

J.G. Kemeny (1953) *The use of Simplicity in Induction*, Philos. Rev. 62, pp. 391–408.

Z. Kohavi (1970, 1978) *Switching and Finite Automata Theory*, McGraw Hill.

A.N. Kolmogorov (1933) *Foundations of the Theory of Probability*, Chelsea.

— (1965) *Three Approaches to the Quantitative Definition of Information*, Problems Inform. Transmission, Vol. 1, pp. 4–7.

—, (1968) *The Logical Basis of Information Theory and Probability Theory*, IEEE-TR-IT-14, No. 5, pp. 662–664.

S. Kullback (1959) *Information Theory and Statistics*, Wiley.

S. Levinson, L. Rabiner and M. Sondhi (1983) *Introduction to the Application of the Theory of Probabalistic Functions of a Markov Process to Automatic Speech Recognition*, BSTJ-63, No. 4, p. 1035.

L. Ljung (1987) *System Identification: Theory for the User*, in Press.

J.L. Marroquin (1985) *Probabilistic Solutions to Inverse Problems*, MIT EECS PhD Dissertation.

G. Marx and N. Reichman (1984) *Routinizing the Discovery of Secrets*, American Behavioral Scientist 27, No. 4, pp.423–452.

R. Michalski (1983) *Conceptual Clustering*, in Michalski, Carbonell and Mitchell (eds.) *Machine Learning: An Artificial Intelligence Approach*, Tioga Press.

R. Montague (1970) *Universal Grammar*, in his collected works *Formal Philosophy*, Yale Univ. Press, 1974.

— (1973) *The Proper Treatment of Quantification in Ordinary English*, in his collected works *Formal Philosophy*, Yale Univ. Press, 1974.

A. Nerode (1958) *Linear Automaton Transformations*, Proc. Am. Math. Soc. 9, pp. 541–544.

C.H. Papadimitriou and K. Steiglitz (1982) *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall.

E.A. Patrick (1972) *Fundamentals of Pattern Recognition*, Prentice-Hall.

D.B. Paul (1985) *Training of HMM Recognizers by Simulated Annealing*, IEEE Decision and Control, p. 13.

T. Pavlidis (1977) *Structural Pattern Recognition*, Springer-Verlag.

K.R. Popper (1962) *Conjectures and Refutations: The Growth of Scientific Knowledge*, Basic Books.

J.R. Quinlan and R.L. Rivest *Inferring Decision Trees Using the Minimum Description Length Principle*, Draft.

L. Rabiner, S. Levinson, M. Sondhi (1983) *On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Work Recognition*, BSTJ-62, No. 4, p. 1075.

Jorma Rissanen (1978) *Modeling by Shortest Data Description*, Automatica, Vol 14, pp. 465–471.

— (1983) *A Universal Prior for Integers and Estimation by Minimum Description Length*, Annals Stat., Vol. 11, No. 2, pp. 416–431.

— (1983) *A Universal Data Compression System*, IEEE-TR-IT-29, No. 5, pp. 656–664.

— (1984) *Universal Coding, Information, Prediction, and Estimation*, IEEE-TR-IT-30, No. 4, pp. 629–636.

— (1986) *Stochastic Complexity and Modeling*, Annals Stat., Vol 14, No. 3, pp. 1080–1100.

— (1986) *Complexity of Strings in the Class of Markov Sources*, IEEE-TR-IT-32, No. 4, pp. 526–532.

R.L. Rivest (1987) *Diversity-Based Inference of Finite Automata*, MIT LCS Report.

S. Rudich (1985) *Inferring the Structure of a Markov Chain from its Output*, IEEE Foundations of Computer Science, pp. 321.

F. Schweppe (1973) *Uncertain Dynamic Systems*, Prentice-Hall.

C. Shannon (1948) *The Mathematical Theory of Communication*, Univ. Illinois Press.

R. Sinsheimer (1978) *The Presumptions of Science*, Proceedings of American Academy of Arts and Sciences, *(Daedalus)*, V. 107, No. 2, pp. 23–35.

R.J. Solomonoff (1964) *A Formal Theory of Inductive Inference. Part I*, Information and Control, Vol. 7, pp. 1–22, —. *Part II*, pp. 224-254.

—, (1978) *Complexity-Based Induction Systems: Comparisons and Convergence Theorems*, IEEE TR-IT-24, No. 4, pp. 422–432.

—, (1986) *The Application of Algorithmic Probability to Problems in Artificial Intelligence* in L.N. Kanal and J.F. Lemmer (eds.) *Uncertainty in Artificial Intelligence*, pp. 473–491, Elsevier.

Texas Instruments Inc. (1985) *PC Scheme User's Guide* and *TI Scheme Language Reference Manual*, 12337 Technology Boulevard, Austin, Texas, 78759.

B.A. Trakhtenbrot and Ya. M. Barzdin', (1973) *Finite Automata, Behavior and Synthesis*, North Holland.

A. Van der Mude and A. Walker (1978) *On the Inference of Stochastic Regular Grammars*, Information and Control Vol. 38, pp. 310–329.

H.L. Van Trees (1968) *Detection, Estimation, and Modulation Theory*, Wiley.

A.J. Viterbi (1967) *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*, IEEE-TR-IT-13, pp. 260–269.

C.S.Wallace and D.M. Boulton, (1968), *An Information Measure for Classification*, Computer Journal 11, No. 2, pp. 185–194.

S. Watanabe (1985) *Pattern Recognition: Human and Mechanical*, Wiley.

N. Wiener (1948) *Cybernetics: or Control and Communication in the Animal and the Machine*, MIT Press.

P. Winston (1975) *Learning Structural Descriptions from Examples*, in P. Winston (ed.) *The Psychology of Computer Vision*, McGraw-Hill.

P. Winston and R. Brown, (eds.) (1979) *Artificial Intelligence: An MIT Perspective*, MIT Press.