A GENERAL SELECTION CRITERION FOR INDUCTIVE INFERENCE

Technical Note **372**

December 1985

By: Michael P. Georgeff
Artificial Intelligence Center
Computer Science and Technology Division
and
Center for the Study of Language and Information
Stanford University
Stanford, California

Christopher S. Wallace
Department of Computer Science
Monash University
Clayton, Victoria, Australia

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This is an extended version of a paper presented at the *Sixth European Conference on Artificial Intelligence* held in Pisa, Italy, September 5-7, 1984.

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

ABSTRACT

This paper presents a general criterion for measuring the degree to which any given theory can be considered a good explanation of a particular body of data. A formal definition of what constitutes an acceptable explanation of a body of data is given, and the length of explanation used as a measure for selecting the best of a set of competing theories. Unlike most previous approaches to inductive inference, the length of explanation includes a measure of the complexity or likelihood of a theory as well as a measure of the degree of fit between theory and data. In this way, prior expectations about the environment can be represented, thus providing a hypothesis space in which search for good or optimal theories is made more tractable. Furthermore, it is shown how theories can be represented as structures that reflect the conceptual entities used to describe and reason about the given problem domain.

1 Introduction

The capacity to form a good theory or explanation for a given body of data is an important component of intelligent behavior and has considerable practical significance in many technological applications.

The essence of most approaches to this problem is to define some criterion for what constitutes a good or useful theory, and then to search the space of all possible theories for the one that, under this measure, best describes the given data. One possible criterion is to consider the best theory to be the one that has highest joint probability with respect to the data. This leads to Bayesian decision methods [8,13]. Another approach is to consider the best theory for a given set of data to be the least complex theory that explains the data [8,15].

The hypothesis spaces considered in most of these cases are very general, and include theories based, for example, on grammars or vector spaces. However, although such theories are suitable for describing a wide range of data, they often have no simple conceptual interpretation. This is a serious problem when the user wants not only to find good theories, but ones that provide explanations in terms of the entities and constructs he is accustomed to reasoning about. Furthermore, it is difficult to provide the required a priori probabilities or complexity measures, as problem-specific information is not easily expressible in the theory description language.

In contrast, recent work in AI has recognized the need for structural theories that better match human conceptualizations [3]. However, most of these approaches have two drawbacks. First, they are suited only to problems in which the theories are deterministic and cannot be applied in domains where a significant element of probability or noise is involved [7]. Second, and more critical, is the fact that the criterion used for selecting good or optimal theories is based solely on the closeness of fit between theory and data. These methods thus have no way of accounting for the complexity or likelihood of theories; consequently, they can always improve the data fit by generating increasingly complex theories. In particular, they will even generate "acceptable" theories for totally random data.

In this paper we provide a framework for inductive inference that can be applied to arbitrarily complex hypothesis spaces containing either deterministic or nondeterministic theories. The selection criterion we propose incorporates a measure of theory complexity and likelihood as well as a measure of the degree of fit between theory and data. This allows prior expectations about the environment to constrain and shape the hypothesis space, so that search for a good theory within this space becomes tractable. Furthermore, we show how theories can be structured to match the conceptual entities used in describing and reasoning about the given problem domain.

2 The Nature of Theories

We consider a *theory* or *concept* to be an abstraction of a class of data. For example, consider the collection of points in two dimensional space as shown in Figure 1(a). This data could be viewed abstractly as comprising two circles with specified centers and radii (Figure 1(b)). Note that the points themselves need not actually lie on the circles —- from the point of view of the abstraction, this "scatter" is unimportant. Similarly, the position a point may occupy on the circumference of the circles is immaterial.

To describe any particular element or instance of data, we need not only to specify an appropriate theory, but also to provide information that identifies that element within the class of data represented by the theory. For example, the data of Figure 1(a) could be abstractly represented by the two circles given in Figure 1(b). To describe this particular element (i.e., the actual data in Figure 1(a), as distinct from other data that can be viewed abstractly as the two circles of Figure 1(b)), for each point we need to specify to which circle it belongs, its radial angle with respect to the center of this circle, and its distance from the circle perimeter (to a given level of accuracy). We will call this additional information the data specification with respect to the given theory, or, when no ambiguity can arise, simply the data specification. The theory specification and the data specification, *taken together*, are called an *explanation* of the data.

Let us now assume that there are several possible theories for representing some particular element of data. We would like some measure that would enable us to order these theories on the basis of how well they explain this data. The measure we propose is the length of explanation of the given data, and we say that one theory is *better* than another if it yields a *shorter* explanation.

Of course, the length of an explanation will depend on the languages or codes used for describing both the theory and the data with respect to that theory. These languages should reflect our prior expectations about the environment, and descriptions of common or important objects should be shorter (simpler) than descriptions of unusual or unimportant objects. We will therefore require that these languages be "efficient," i.e., that they provide optimal encodings of theories, with respect to their a priori probability of occurrence, and of data, with respect to each theory. Indeed, this

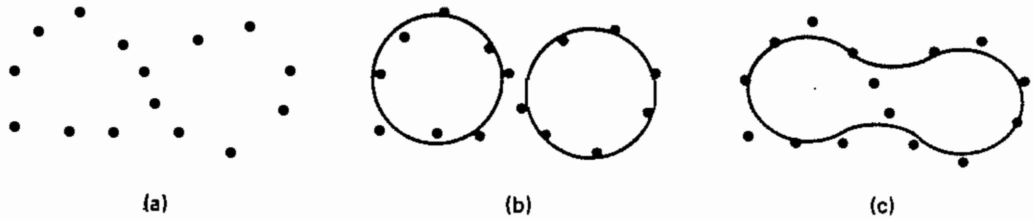

Figure 1: Data and Theories

is just what happens in both everyday and technical languages — the most common or useful entities are, on the whole, given shorter descriptions than rare ones.

For example, consider again the data shown in Figure 1(a). One might represent this data as two circles with specified centers and radii (Figure 1(b)), or, alternatively, as a dumbell-like shape with specified position and size (Figure 1(c)). In general, for efficient data specification languages, the better the fit of data to theory, the shorter the specification. Thus, by virtue of the better fit in this case, the data specification according to the circle theory can be expected to be shorter than the one according to the dumbell theory. If it is assumed that the specifications of the two theories are of equal length, then the explanation based on the theory that the data comprises two circles would be shorter than that based on the dumbell theory. The circle theory would therefore be the better one for representing this data.

On the other hand, if we know a priori that most of the objects under consideration are dumbell-shaped rather than circular, our language for describing theories should make it harder to describe circles than dumbells. Similarly, if we know what sizes of objects to expect, and in what positions to expect them, the descriptions of theories that place the objects close to these expected values should be simpler than the descriptions of theories that place the objects far away. In the particular case given in Figure 1, the explanation based on the dumbell theory may then turn out to be shorter than the one based on the two-circles theory, despite the poorer fit of the data. The dumbell theory would then be considered the better theory for describing the data in the light of our expectations regarding the problem domain.

This means that what constitutes a good theory will always be dependent on our expectations about the world. From an objective point of view, this might seem disappointing. But it is exactly these expectations that make the induction problem tractable.

3 Formal Description

Let $\mathcal{D}$ be a given class of data. Then a *theory* T is considered to be an abstraction of a subset D of $\mathcal{D}$, namely, that subset that might be expected to be observed if T were true. Each element of D has an associated probability of occurrence under the theory T. The subset D is called the data *represented* by T.

Now consider a language for describing (denoting) the elements of D, given the theory T. Such a language will be called a *data specification language* for T. A data specification language is said to be *efficient* if and only if it is an optimal encoding of the data in D (i.e., the encoding minimizes the expected message length of data, given the associated probabilities of elements in D under T). For a given theory T, the set of sentences in the data specification language that describe some data element d in D will be denoted $s(T,d)$. Obviously, if the data specification language is efficient, $s(T,d)$ will be a singleton set for all d in D. In such cases, we will not distinguish between singleton sets and the element that each contains.

A *hypothesis space* H is a countable set of theories over the data space $\mathcal{D}$ in which each theory has an associated probability of occurrence in H. We will call a language, the sentences of which describe the theories in H, a *theory specification language*. A theory specification language is said to be *efficient* if and only if it is an optimal encoding of the theories in H. For a given H, the set of sentences in the theory specification language that describe some theory T in H will be denoted $h(H,T)$. If the theory specification language is efficient, $h(H,T)$ will be a singleton set for all T in H.

An *explanation* of an element d of $\mathcal{D}$ in terms of a theory T, denoted $e(H,d,T)$, is then defined to be

$$e(H,d,T) = h(H,T) \; . \; s(T,d) \quad ,$$

where "." represents concatenation.

Consider that a given hypothesis space H is described by an efficient theory description language and that all theories in H use an efficient data specification language. A theory T_1 is then said to be *better than* a theory T_2 for describing some data d in D if

$$length(e(H,d,T_1)) < length(e(H,d,T_2)) \quad ,$$

where $length(x)$ represents the length of x (or, strictly, the length of the one element in the singleton set x). [1]

We are interested only in theories that capture some useful structural properties of data or, equivalently, in encodings of data that shorten its description. Thus, a theory T in H is said to be a *null theory* if $length(e(H,d,T)) = length(d)$. If T is such that $length(e(H,d,T)) > length(d)$, then we say that d is *not explained* by T — what structure T may have captured is more likely to be the result of chance.

Note that, if we interpret $length(h(H,T))$ as the log probability of T and $length(s(T,d))$ as the log conditional probability of d, given T, then $length(e(H,d,T)$ is exactly the [unnormalized] log conditional probability of T, given d, as determined by Bayes' theorem.

One way of viewing the relationship between hypothesis spaces, theories, and data is as follows. Assume that, for a given hypothesis space H, we have a compiler $C(H)$ that takes as input the description of a theory in a theory specification language and outputs a program $P(T)$. This program in turn accepts a description of some data d in a data specification language and outputs d. Thus C and P together form a decoding device for data encoded in two parts: a so-called theory specification part and a data specification part. If the theory specification language is efficient and, for all theories, the data specification language is efficient, the best theory for describing d will then be the one for which the total length of input (i.e. to both the compiler C and the resulting program P) is a minimum. If for some theory the input is longer than the output, then that theory does not provide a reasonable explanation of the given data.

In their work on classification techniques, Michalski and Stepp [9] similarly distinguish between theory and data descriptions. So-called l-complexes are considered to denote sets of data, and thus correspond to theory specifications. A measure of "sparseness" is used to determine the degree of fit between data and theory, and corresponds to the length of data specification under the assumption that the probability distribution over data elements is uniform. However, the goodness of an l-complex

[1] In comparing languages, we will assume them to be over the same alphabet.

(theory) is determined solely according to this measure of sparseness, and thus the approach takes no account of prior knowledge about the likelihood of theories within the hypothesis space.

Failure to take proper account of theory likelihood usually gets one into trouble. In the learning system LEX, for example, Mitchell [11] found it impossible to give any justification for the "inductive leaps" attempted by the generalizer. Thus, given two integral expressions differing only in that one contains a 3 whereas the other contains a 5, Mitchell notes that the system has no basis for generalizing 3 and 5 to "any integer," rather than "any prime integer," or, for that matter, "any integer except 251". Indeed, under Michalski's sparseness measure, the last generalization would be preferable to the first. The reason for this is not the "syntactic" nature of the task [11], but rather the lack of a selection criterion that exploits prior knowledge about possible generalizations.

Another serious deficiency of selection criteria that do not include a measure of theory complexity is that it is then always possible to improve the data fit by making the theory more complex. In particular, this can result in the attribution of structure to totally random data.

4 Structural Descriptions

While the preceding section provided a formalism for choosing a best theory in a given hypothesis space, it does not offer much guidance in constructing theory and data specification languages.

There are various approaches one could adopt. For example, one could choose to specify theories in terms of probabilistic Turing machines, then to specify data (relative to a given Turing machine) by prescribing the moves required of the Turing machine to produce that data. This sort of approach underlies much work in grammatical inference [5,8]. Alternatively, one could describe theories by selecting a point in some vector space, and then describe the data (relative to the selected point) by giving its distance from the point. This approach is the basis for standard statistical classification techniques [13].

However, by representing theories in this manner, it often becomes very difficult to determine efficient encodings of the theories and data appropriate to the particular problem domain under consideration. The difficulty is that the kind of problem-specific knowledge essential to providing this information is not readily expressed in terms of such general structures.

It is therefore preferable to consider theories to be composed of several simpler parts, each of which has some clear conceptual interpretation. The theory description is then the concatenation of a sequence of component descriptions. Typically, the components will correspond closely to features and concepts which would be mentioned in a natural-language description of the theory. In the theory description language, the existence and type of each component is specified by a description of length minus-the-log-probability of finding such a component, given the existence and types of components already described. Where there are no strong prior expectations about the relative probabilities of different types of components, this length is essentially just the log of the number of different components which might have been described, given the previously specified components.

Where a component requires specification of real-valued parameters, the number of different possible components is, of course, uncountably infinite. In such cases, it can be shown that the parameter values stated in the component description should be stated to a precision (number of significant digits) consistent with the error expected in estimating the parameter values from the given data.

These rules generally suffice to define an efficient theory specification language. However, it may frequently happen that the order of the description of some set of N components affects neither the meaning of the description nor the way in which each component is described. The language is then redundant, since $N!$ different, equally long descriptions of the same theory can be given. We do not need actually to construct the coded description of a theory and data, but rather need only to calculate the length of the description. Hence, rather than trying to avoid such simple redundancies, we merely correct the calculated length of the redundant description by subtracting log $N!$.

A similar approach can be adopted for specifying the data for each theory in the hypothesis space. That is, the data can be represented as a structure with associated component probabilities, and a data specification language chosen so that the length of any sentence in the language will be equal to the sum of minus-the-log-probability of the occurrence of each of the structure's components.

Structured theory descriptions can be viewed as "frames" [10] in which each component represents a different "slot" value. However, they are more general in that the notion of a single slot default value is extended to be a probability distribution over all possible slot values. This way of representing theories or concepts also generalizes the ideas of concept modeling proposed by Cohen and Murphy [2], and avoids most of the difficulties associated with classical and prototype theories of representation.

5 Inference of Line Segments

Let us consider the problem of trying to best fit one or more straight line segments to a set of data points in a two-dimensional field, where some points may be noise and the number of line segments is unknown.

We suppose the data to comprise N coordinate pairs $\langle x_i, y_i \rangle$, $i = 1,..N$, of points in a square field of size R by R, where each coordinate is measured to precision ϵ. Since each coordinate can be specified by a word of length $log(R/\epsilon)$, the data can be specified directly by a sentence of length $2N\ log(R/\epsilon)$. Assuming no coincident points, this is the best one can do in the absence of any theory about the data.

However, a possible class of theories about the points of interest in the scene may be that many of the points form one or more straight line segments. Each such theory could then be described by giving the following component descriptions:

1. A statement of the number of line segments in the scene. The length of this component will depend on the number of lines expected in such scenes.
2. A statement of the approximate number of points treated as "background noise," i.e., not assigned to any line. The coding of this component may incorporate a prior expectation about the frequency of noise points (given, say, the characteristics of the sensors).
3. For each line, a statement of the coordinates of its center, its angle, and its length. The coding of this component will depend on the expected values of these attributes (given, say, the physical world) and the degree of precision to which these attributes are specified.

In the example given below, we assume that any numbers of lines are equally likely, and thus take the length of the first component to be a constant. Therefore, in comparing theories, the contribution of this component can be disregarded. We assume the fraction f of noise points to have a beta prior distribution of form $10(1-f)^{10}$, with mean 0.1, and assume the expected values of position, angle, and length of the line segment to be uniformly distributed.

Given a theory (specifying one or more line segments, their location, angle and length, and the number of noise points), the description of each point $\langle x_i, y_i \rangle$ in the scene then has two parts. The first declares the point i to be either noise or a member of one of the lines specified by the theory. If it is noise, the second part specifies $\langle x_i, y_i \rangle$ using a code word of length $2\ log(R/\epsilon)$. If point i is a member of a line segment, the

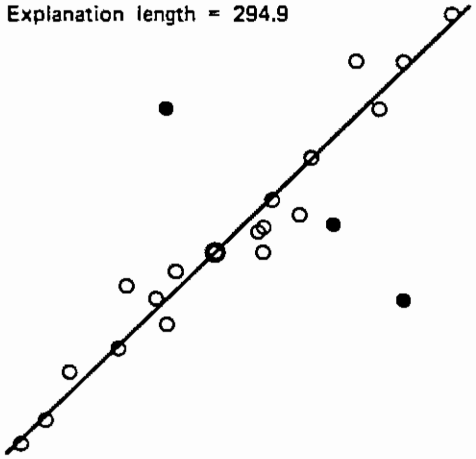

Figure 2(a): Best One-Line Theory

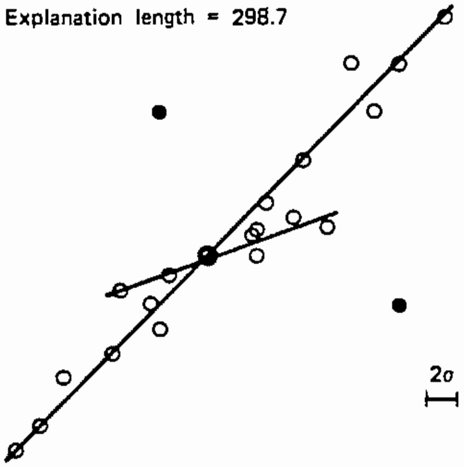

Figure 2(b): Best Two-Line Theory

second part specifies its position along the line and its distance therefrom. The length of the specification of these two attributes will depend on prior expectations about the scatter of points along and about the line, respectively. In the example given below, we assume a uniform distribution of position along the line and a Gaussian distribution of distance normal to the line.

In Figure 2 we show one such set of points together with the best single-line theory (Figure 2(a)) and the best two-line theory (Figure 2(b)). The points treated as noise by the explanations are shown solid. The length of the data when no lines are assumed (that is, the "null" theory), is 317.7, the length of the one-line explanation is 294.9 and the two-line explanation 298.7. As both explanation lengths are shorter than the null theory length, both are reasonable theories for the data. The one-line theory is the one preferred, as the corresponding explanation is shorter.

More details about the encoding of the theory and data specifications can be found in one of our earlier reports [17]. However, it is important to note that the fine details of the encoding (e.g., the value chosen for the standard deviation of the Gaussian distribution, the Gaussian distribution itself, or the assumption of equal expectations as to the number of line segments) have a relatively small effect on code length. The major determinant of code length (and hence of goodness of theory) is the complexity of the theory structure. For example, independently of how much of a message must be used to specify the number of line segments, a theory consisting of a large number of lines requires a long message just to describe the characteristics of each line, and in the data specification part, the particular line to which each point helongs. So there is a strong bias against theories with a large number of components — a bias that can only be countered either by prior knowledge that such complex theories are highly likely or by very compelling evidence in the data.

6 Inference of the Structure of Automata

The second example we will consider is the problem of inferring the structure of a probabilistic finite state automaton (PFSA) from a data string. Because it is not a "real-world" problem, we can bring very little problem-specific information to bear on its resolution. Thus, choosing appropriate a priori probabilities for the components of the PFSA, or finding useful heuristics for searching the hypothesis space, is made exceedingly difficult. Nevertheless, grammatical inference has received a lot of attention in the research literature, and it is appropriate to show how our approach deals with the problem.

A PFSA, M, is a finite-state automaton with a stochastic transition function. It has no input, but emits a symbol on each state transition. Let $t_M(s,c)$ be the value of its transition function when moving from state s and emitting symbol c. For each state s and symbol c, $t_M(s,c)$ is a pair $< s', p >$ called a *move*, where s' is the next state and p is the probability that, when in state s, M will make this move. For all s, the sum of p over all c is 1. Such a PFSA is shown in diagrammatic form in Figure 3(b). (The transition probabilities have been replaced with relative frequencies labelling the arcs.)

The PFSAs we consider are restricted to have at most one transition arc from each state labeled with a given output symbol. These PFSAs represent the theories in our hypothesis space H.

Now assume that, for some data string d, a PFSA, M, is chosen as a possible theory for d. The first part of an explanation for d under this theory will therefore consist of a description of the machine M. This description consists of three components: (1) a specification of the number of states n; (2) for each state, a specification of the transition probability for each output symbol; and (3) for each state and output symbol in that state, a specification of the destination state of the transition labeled with that output symbol.

To describe fully the data string d, we now need to include in our explanation a specification of how to select this particular string from all the others generated by M. This can be done by specifying the sequence of moves that M needs to make in order to generate d. This specification has to be efficient — that is, it should take into account the probabilities of the moves as contained in the specification of M. Thus, we take the length of the data specification part to be simply the sum of minus-the-log-probabilities of all the transitions made. Details of both the theory and data specifications can be found elsewhere [17].

For example, consider the data string

CAAAB/BBAAB/CAAB/BBAB/CAB/BBB/CB

where "/" is a delimiter (that is known a priori to return to the initial state). Figure 3(a) shows how the length of explanation of the best n-state machine varies with n over the range 1 to 7, and Figure 3(b) shows the structure of some of the best n-state machines. It is clearly seen that the best theory overall, according to our measure, is the four-state machine.

In comparison, for a given data string d, Gaines [7] prefers that PFSA that minimizes the length of the data specification alone, and does not take into account the

Number of States	Length of Explanations
1	(67.6)
2	60.0
3	57.4
4	54.1
5	60.7
6	61.7
7	(70.0)

Figure 3(a): Length of Explanations of Automata Output

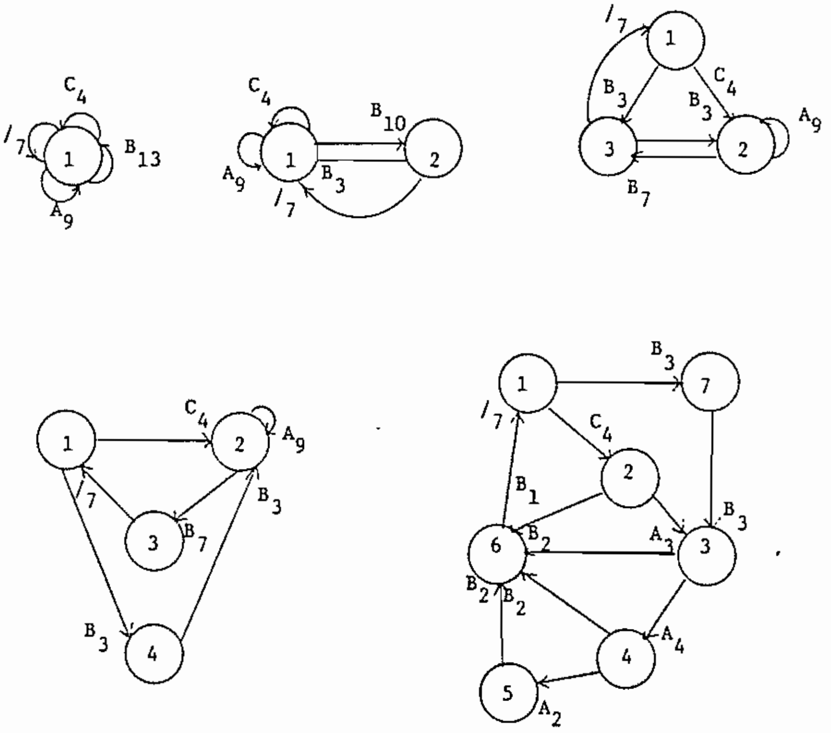

Figure 3(b): Structure of Best Automata of 1,2,3,4 and 7 States contribution of describing the PFSA chosen. He is thus unable to formally compare machines that have different numbers of states. In this example, Gaines finds PFSAs of up to 7 states with, of course, increasing probability that the machine will generate the given data string. On intuitive grounds, Gaines suggests that the four-state model is the most satisfactory. The four-state machine also corresponds to the grammar derived by Feldman et al. [6] and Evans [4], who employed heuristic schemes for this particular data.

Note that the length of the data specification part of the seven-state machine is the minimum over all machines. An increase in the number of states thereafter produces no further decrease in length. Thus it is only the great complexity of the seven state theory that prevents it from the being the best theory for the data. In fact, the machine is so complex that the total length of the explanation exceeds the "null" theory length of 66 — in effect, the seven-state machine does not explain the data at all. On the other hand, the one-state machine is so simple that, despite a very short theory description, it requires a very long data specification; it likewise provides no explanation of the data.

The approach described herein has also been applied to a variety of other problems, including numerical taxonomy and classification [1,16], theory discrimination [12], and linear feature matching [14].

7 Theory Generation and Search

The aim of this paper has been to provide a selection criterion for inductive inference based on conceptual structures rather than undifferentiated metrics. However, this is only part of the inductive-inference problem. In addition, one needs techniques for generating new theories and searching efficiently for good or optimal ones [3]. Although we offer no solutions for these other problems, we indicate below how the selection criterion we propose can help significantly in restricting the number of theories that need be generated and in reducing the search.

The fact that the selection criterion incorporates a measure of a theory's complexity and likelihood as well as the fit between theory and data means that unlikely theories will have a high associated cost, even before comparison with the data, and consequently never need be generated. Furthermore, unlike most other measures used, the length of explanation provides a measure by which theories can be *rejected.* Thus, in cases where the search cannot guarantee optimality, we can compare the length of the explanation under the theory with the length of the original data string, and thereby at least determine whether or not the theory is *acceptable*. In particular, this prevents the assignment of structure to random data.

A rejection criterion is particularly important for staged search, where the most probable or simplest theories are considered first, and others examined only if necessary. In this approach, it is essential to have some means of deciding whether or not to proceed to the next stage; the length of explanation provides a basis for doing this. In the extreme case, comparison of the length of explanation under the currently best theory with the length of the data under the null theory determines whether one has a theory at all. If the theory does not result in a reduction in coding, it clearly does not explain the data and one *must* go on and consider other theories. One can stop when the reduction in coding is substantial enough to be satisfactory, depending on the resources that one has available for further searching. Of course, it may be that no theory in the hypothesis space can provide an explanation for the data.

The selection criterion proposed herein can also be employed with hierarchical representations. Explanation length can be used to select among competing subtheories at some low level of abstraction, which in turn can form the basis (i.e., the "data") for theories at a higher level of abstraction. There is no guarantee that such an approach will lead to the best global theory, but it is reasonable to expect in most natural domains that the resulting global theory will be at least near-optimal.

For example, in object recognition, the approach described in Section 5 could first be used to find straight-line segments in a set of points. The resulting lines could then be further "explained" by a higher-level theory that describes the lines in terms of regions or shapes. At still higher levels in the hierarchy, these regions or shapes would form the data to be explained by theories about objects. Just as for the lowest level in the hierarchy, the length of explanation could be used to choose among alternative higher-level descriptions.

8 Conclusions

In this paper we have provided a general selection criterion for inductive inference. The approach allows the construction of structured theories that can provide explanations of phenomena in terms of abstract entities and concepts. Prior expectations about the environment can be readily expressed, and thus the search for an adequate theory made tractable.

In particular, we have introduced a measure of the complexity of a theory (the length of the theory specification) that is compatible with the measure of fit of the theory with the data (the length of the data specification). Other measures of fit and model complexity have been proposed, but it is not clear how they might or should be combined to provide an overall criterion of selection among competing theories. Because our measures are compatible, we may simply add them to give a measure (the length of explanation) that can be used to choose the best of a set of theories. Moreover, our measure provides an absolute criterion for rejecting theories.

With the introduction of the theory specification language, we have clarified the role and construction of prior probability distributions over families of theories. We have shown how structured theories that reflect the conceptual entities of the problem domain provide a natural framework in which expectations about the domain may be utilized and combined to define a prior distribution over a complex family of theories. This in part indicates why general theory specification languages (such as Turing machines, grammars, or vector spaces) have proved to be unsuitable for concept learning; it is simply too difficult to express prior knowledge about the problem domain under consideration. Furthermore, this framework makes it clear that any reasonable assignment of prior probabilities to theories is dominated by the latter's relative complexity rather than by subjective expectations.

References

[1] D. M. Boulton and C. S. Wallace. An information measure for hierarchic classification. *Computer Journal*, 16, 1973.

[2] B Cohen and G. L. Murphy. Models of concepts. *Cognitive Science*, 8:27–58, 1984.

[3] T. G. Dietterich and R. S. Michalski. Inductive learning of structural descriptions. *Artificial Intelligence*, 16:257–294, 1981.

[4] T. G. Evans. *Grammatical Inference Techniques in Pattern Analysis, Volume 2*, pages 183–202. Academic Press, New York, New York, 1971.

[5] J. Feldman. Some decidability results on grammatical inference and complexity. *Information and Control*, 20:244–262, 1972.

[6] J. A. Feldman, J. Gips, J. Horning, and S. Reder. *Grammatical Complexity and Inference*. A.I. Memo 69, Stanford University, Stanford, California, 1969.

[7] B. R. Gaines. Behavior structure transformations under uncertainty. *International Journal of Man-Machine Studies*, 8:337–365, 1976.

[8] E. B. Hunt. Artificial intelligence. Academic Press, New York, New York, 1975.

[9] R. S. Michalski and R. E. Stepp. An application of AI techniques to structuring objects into an optimal conceptual hierarchy. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 460–465, Vancouver, Canada, 1981.

[10] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–277, McGraw-Hill, New York, New York, 1975.

[11] T. M. Mitchell. Learning and problem solving. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 1139–1151, Karlsruhe, Germany, 1983.

[12] J. D. Patrick and C. S. Wallace. Stone circle geometries: an information theory approach. In Heggie, editor, *Archeoastronomy in the Old World*, Cambridge University Press, Cambridge, England, 1982.

[13] G. S. Sebestyen. Decision-making processes in pattern recognition. MacMillan Press, New York, New York, 1962.

[14] G. B. Smith and H. C. Wolf. Image to image correspondence: linear feature matching. In *Proceedings of the Second NASA Symposium on Mathematical Pattern Recognition and Image Analysis*, Johnson Space Center, Houston, Texas, 1984.

[15] R. Solomonoff. A formal theory of inductive inference I and II. *Information and Control*, 7:1–22 and 7:224–254, 1964.

[16] C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 11, 1968.

[17] C. S. Wallace and M. P. Georgeff. *A General Objective for Inductive Inference.* Computer Science Technical Report, Monash University, Victoria, Australia, 1983.