PATTERNS OF INDUCTION AND ASSOCIATED KNOWLEDGE ACQUISTION ALGORITHMS[1]

Frederick Hayes-Roth
Computer Science Department[2]
Carnegie-Mellon University
Pittsburgh, Pa. 15213

May 13, 1976

ABSTRACT

The common need of both Artificial Intelligence and Pattern Recognition for effective methods of automatic knowledge acquisition is considered. A pattern of induction is defined as a framework which relates a theory of behavior generation, underlying knowledge structures, and a learning methodology. One particular learning theory, called interference matching, suggests that knowledge structures which underlie behavior descriptions can be directly abstracted from those descriptions. Because of the close connection between descriptions and inferences in such a framework, the strengths and weaknesses of several types of descriptions are considered. Algorithms which exploit this theory are presented for three classes of problems: pattern learning and classification; induction of quantified production rules; and the induction of syntactic categories and phrase structure rules. Preliminary results are presented and directions for future research are outlined.

---

1 This paper will appear in the Academic Press publication of invited papers of the

1. INTRODUCTION

Despite many impressive advances in the areas of knowledge representation and engineering, Artificial Intelligence (AI) has made virtually no progress on general learning problems in the last 20 years. Both AI and Pattern Recognition (PR) currently experience a pressing need for automatic methods of knowledge acquisition, but their problems are somewhat different. Current efforts in AI aimed at building large-scale knowledge-based systems (e.g., for speech, vision, text understanding) are virtually overwhelmed by the task of knowledge engineering. The goal of this task is the implementation of all potentially valuable "knowledge sources," problem solving modules which exploit the known physical, syntactic, contextual, and semantic relations to constrain the search for solutions. The cost--in terms of people, time, and machine resources--of translating this human knowledge into computer programs is nearly insupportable. Furthermore, even after these handcrafted knowledge sources are developed, they are difficult to evaluate comparatively because each tends to be "one-of-a-kind," a body of code specially tailored to operate in one specific system and to employ only one particular subset of the many potentially relevant problem solving techniques. Thus, to a large extent, the immediate need for general learning procedures in AI is to automate much of the work of knowledge programming. The field of PR, on the other hand, needs general learning procedures because the conventional methods of pattern description and learning do not perform well in most complex environments. The inadequacy of the well known dimensional, parametric, and syntactic techniques of representation and classification is made apparent by their inability to contribute significantly to modern AI understanding systems. In short, the field of PR has reached a point where its principal tools no longer seem sufficiently suited to the recognition problems being encountered. The development of general learning procedures which can generate symbolic pattern representations and facilitate improved classification in such domains is a goal of great importance for the PR field.

This paper considers three types of general learning problems related to pattern classification, rule induction, and syntax learning. Each of these is approached within the theoretical framework of a related pattern of induction or learning paradigm. A pattern of induction is an analytical framework which relates and organizes the various components of a learning problem and its solution. The first three components of the induction pattern are as follows: (1) a model of a knowledge-based system which defines a body of knowledge or information (K) and a behavior generating function (B) which operates on the knowledge to produce observable behaviors

(training data); (2) a collection of observed behaviors which constitute the training data (I) for the induction algorithm; and (3) a learning algorithm (L) which operates on the training data to infer the knowledge K which produces or "causes" the observations I. If we assumed that the behavior generating function B were known and invertible, the ideal knowledge acquisition algorithm L would be its inverse, such that $L(I) = B^{-1}(I) = K$; that is, we would simply apply the inverse function B^{-1} to I to identify K. This view gives rise to the last component of the induction pattern, which is: (4) an induction theory which relates a learning algorithm L to a presumed behavior generator B.

All three learning problems considered in this paper are approached uniformly by means of the same induction theory, which is called interference matching (IM). Basically, this theory holds that the knowledge underlying many training examples of the same pattern or rule may be directly identified by producing a representation (an abstraction) of the examples which emphasizes their commonalities and attenuates their differences. This theory is an extension of a primitive notion of Galton [4] called the composite photograph theory, which suggests that people learn to identify different views of the same object by developing a pattern template (a "composite photograph" transparency) by superimposing in memory many descriptions (transparencies) of varying training views of the object. Under this theory, the resulting template would retain only the essential features, those common to all examples. Any novel view of the same object would then be expected to exhibit (match) all criterial properties of the template. Of course. Galton's conception is completely dependent upon the presumed capacity to superimpose the multiple views in such a way as to preserve all criterial commonalities. Even the slightest difference in size or orientation would nullify the effectiveness of direct, physical superimposition as a method of producing abstractions. Interference matching generalizes the process of extracting commonalities of pattern descriptions to feature and relational representations. It is considered in detail in section 4, after the various types of representations are introduced in section 3.

The three types of learning problems considered in this paper are: symbolic pattern learning, the discovery of disjunctive and conjunctive formulae of the predicate calculus which characterize diverse examples of one pattern class and distinguish that class from other classes; rule learning, the identification of universally quantified [condition=>action] productions derived from training data consisting of before-and-after pairs of events which have been transformed by an unknown rule; and category learning, the formation of sets of functional substitutes or alternatives which constitute the domain of choices allowable within particular syntactic behavior rules. Each of

these problems is described in more detail in section 2. Section 3 discusses a number of alternative knowledge representation schemes in terms of their capacity to facilitate learning from examples. The representations considered include simple feature codes, topologically organized feature manifolds which facilitate generalization and discrimination of patterns over noisy and continuous attribute values, complementary feature codes for missing (negative) attributes which facilitate discovery of disjunctions, and relational descriptions which facilitate learning of structured patterns, production rules, and categories. Section 4 explains interference matching and two of its products, the abstraction and the residuals of compared representations. Abstractions produced by IM provide the solution to the first two types of learning problems, and residuals provide the solution to the category learning problem. Related algorithms for knowledge acquisition are described in section 5, and directions for future research are considered in the last section.

2. THREE LEARNING PROBLEMS

Three learning problems of considerable generality and wide applicability are considered here. The first is the pattern learning and classification problem: Given descriptions of an unclassified test item and several examples of each of a number of mutually exclusive pattern classes, identify the most probable class to which the test item can be assigned. The basic approach followed in solving this problem is to hypothesize that any set of properties (a characteristic) manifested by the test item might reliably distinguish the training exemplars of one class from all other classes. Each such hypothesis is more or less plausible depending on the empiricial likelihood that the associated characteristic occurs primarily among examples of a single class. The likelihood that a characteristic is matched only by examples of one class is called the diagnosticity of the characteristic with respect to the class. To the extent that a characteristic is diagnostic with respect to some class, it is plausible to suppose that a test item matching the characteristic is actually an example of that class. The method used for determining classifications is simply to assign each test item to the one class which is indicated by the most diagnostic characteristic which the test item manifests [6, 7].

The second learning problem is that of rule induction and response selection: Given a description of a novel stimulus and training descriptions of previously experienced antecedent-consequent event pairs (or condition=>action sequences),

choose the most plausible response to the current test stimulus. A special case of rule induction and response selection is that of classifying a test item as belonging to one of several alternative pattern classes. More generally however, we may wish to respond to a stimulus by transforming the stimulus description by adding or deleting relations (asserting or denying predicates). Examples of the kinds of rules which might be induced are the problem solving rules of STRIPS [3] which relate conditions of a problem domain to actions taken on related objects, the rules of transformational grammar which relate before-and-after deep structures of sentences, and the premise-conclusion rules of inferential reasoning systems such as MYCIN [15].

The basic approach taken to this type of problem is to hypothesize that each [condition=>action] pattern which is a characteristic of some [antecedent→consequent] training sequences defines a plausible rule. Those hypothetical rules whose inferred action components are most reliably supported by the training data are considered most plausible. Maximally reliable support for a hypothetical [condition=>action] rule can be claimed whenever all training data whose antecedent events satisfy (match) the condition component of the rule are associated with consequent events which also satisfy the hypothesized action or response component. This is equivalent to extending the notion of diagnosticity to apply to the measurement of the degree to which the presence of a certain condition indicates that a particular response pattern will follow (in the training data). The learning methods which are developed to handle the first learning problem (pattern learning and classification) can be extended in a straightforward manner to solve these rule induction and response selection problems too.

The third learning problem considered is the category and syntax learning problem: Given training descriptions of behaviors generated by a system which arbitrarily selects and systematically relates elements chosen from specific classes of alternatives, identify the unknown classes and the systematic ways in which they are related. The classes are called categories; systematic constraints on the use of categories are called syntax. An example of this sort of problem is to infer from a corpus of natural language in which one class of words (adjectives) systematically precedes another class (nouns), the categories adjective and noun and the relationship that an element of the class of adjectives is usually followed by an element of the class of nouns. As another example, consider learning the category of animate noun from the fact that only such nouns are regularly used as agents of instrumental and reflexive actions and, also, as the objects of verbs expressing affect (such as "love," "admire," etc.). The approach taken to this problem is to hypothesize that several

objects which reliably occur in identical relationships with some other objects constitute a category. It may be inferred that membership in that category is a necessary and sufficient condition for objects to particpate in the observed relationships. Thus a reliably occurring relationship is inferred as a syntactic pattern (e.g., the adjective→noun sequence pattern) at the same time as the actual objects which play consistent functional roles (e.g., the adjectives "big," "brown," "cute," ... reliably precede nouns) enumeratively define related syntactic categories.

Obviously, the approaches outlined above to all three of these learning problems are combinatorial in nature. In every case, it was superficially suggested that each possible hypothesis (characteristic, rule, or category) be considered as a potential solution to the corresponding induction problem. Two separate issues of feasibility must be considered. First, if the combinatorics of this hypothesize-and-test method could be sufficiently controlled, would the proposed method produce good results? That is, would such a method be an effective learning algorithm. Both theoretical arguments and some preliminary experimental data suggest that the answer is yes. Secondy, given that such an exhaustive evaluation of plausible inferred knowledge (patterns, rules, categories) is an effective learning procedure, can heuristic methods be devised which can adequately control the combinatorics of the search? The answer to this question is approached in two ways in this paper. Firstly, a number of alternative knowledge (data) representation schemes have been studied with respect to their providing a feasible basis for induction algorithms. Some surprisingly simple representations make possible quick, effective solutions to seemingly complex pattern learning problems. These alternative representations are considered in some detail in the next section before the related learning algorithms are presented. The second avenue of approach, development of specific heuristics to constrain the amount of computing of the learning algorithms themselves, is discussed in section 5.

3. DATA AND KNOWLEDGE REPRESENTATIONS AS BASES FOR LEARNING

Because the computational complexity of learning algorithms is likely to be exponential, the choice of data representation may significantly affect its feasibility. Four alternative types of representations are considered in this section. Each entails a different combination of desirable and undesirable properties, and these are briefly considered. The four types of representations, ordered in increasing completeness of representational power, are based on (1) simple feature or property lists, (2)

topologically organized feature manifolds, (3) enumerated complementary feature code manifolds, and (4) general relational descriptions. Each of these repesentation schemes is now considered in turn.

Simple feature descriptions are well known. This sort of representation employs the concept of an exogenously identified object which is described by a list of features (i.e., properties, unary predicates or attribute-value pairs). As in all of the four representation schemes considered, the segmentation and identification of objects as well as the feature coding processes are exogenous to the representation itself. The inability of feature list descriptions to express general relationships among several objects in a single event is their chief weakness. This limitation is shared by all of the representation schemes except relational coding. Conversely, it is just because they are so simple that property list descriptions are attractive. While only simple combinations of attributes can be abstracted from training descriptions and employed for pattern recognition, such processing can be performed with great efficiency. Each feature can be associated with a single value in a bit vector--the value being one if the feature is present and zero if not--and matching operations performed by simple bit comparison operations [7, 10]. Specifically, the bit-wise logical product ($\wedge$) of two feature bit vectors is their maximal abstraction, the set of all features common to both of them. Furthermore, if a pattern template is represented by a bit vector of criterial features, the determination of whether any event description matches the template can be performed by a simple bit-wise masking operation.

The attractiveness of bit operations to effect learning (abstraction) and pattern recognition in the framework of feature list descriptions is seriously diminished by the fact that such matching operations are fundamentally all-or-none. Each bit in a feature vector represents an attribute that is or is not present and provides no basis for fuzzy comparisons of two objects. In this context, the term fuzzy refers to a graded measure of the degree to which the values of the same attributes of two objects are similar. The capacity to retain the information that two objects are fuzzily equal is important in many learning problems involving continuous, ordinal, or noisy data. In these tasks, it is often necessary to infer from examples of a pattern an exact range of values on each attribute dimension which is characteristic of the pattern to be learned. While feature list matching is inherently all-or-none, special organizations of features have been identified which enable the efficient production of graded comparisons in such domains. A collection of features organized to facilitate particular learning tasks is referred to as a feature manifold, and two special manifolds which are well suited to the fuzzy match problem are now discussed.

Two general principles of organization for fuzzy match feature manifolds have so far been identified. First, values on any ordinal dimension may be described by features which are organized in an overlapping receptive field (ORF) manifold to facilitate, simultaneously, maximum generalization and discrimination during learning. Consider the problem of representing sampled values of a continuous attribute (e.g., amplitude, frequency, duration) to facilitate the discovery of the range of variability exhibited by successive pattern examples. Many sophisticated approaches to similar problems (e.g., the mixture problem) have been developed which depend upon an a priori parametric model of the data (Cf., [2]). While such an approach may be very efficient in some domains, ORF representations provide an efficient basis for the generation of non-parametric inferences.

Basically, the maximum possible range of attribute values [A, Z] on any dimension of interest is divided into adjacent overlapping intervals. Each interval [Q,Q+G] is assigned to a single feature which represents a range of values in which an observed value may be contained. The term receptive field is borrowed from the psychology of vision. In that context, the receptive field of an individual neuron refers to the specific retinal pattern which causes it to fire. In a similar way, the receptive field of a feature is just the range of possible attribute values which cause it to be true. Adjacent receptive fields overlap so that their ranges include common values. The largest range of values of any feature, G, is the maximum generalizability of any single conjunctive pattern description. The amount of separation between adjacent overlapping fields, D, is the maximum amount of discriminability between any two patterns and corresponds to the psychophysical concept of a just noticeable difference (JND). The values of A, Z, G, and D are the basic parameters of the ORF manifold representation [1, 11]. With only minor modifications needed if D does not evenly divide G or G does not evenly divide (Z-A), the feature manifold F of the simplest ORF representation is defined to be G = {[A, A+D], [A, A+2D], ..., [A, A+G-D], [A, A+G], [A+D, A+G+D], [A+2D, A+G+2D], ..., [Z-G-D, Z-D], [Z-G, Z], [Z-G+D, Z], ..., [Z-D, Z]}. The data representation of any attribute value h in the manifold F is $R_F(h)$ = {[a, b] : h ∈ [a, b] and [a, b] ∈ F}. As a result of such representation, two data values h and h' may be fuzzily compared by simple set intersection of $R_F(h)$ and $R_F(h')$: $R_F(h) \cap R_F(h')$ = {[a, b] : h ∈ [a, b] and h' ∈ [a, b] and [a, b] ∈ F} = $[\min_F(h, h'), \max_F(h, h')]$, where $\min_F(h,h')$ is the minimum of h and h' modulo the precison specified by D and $\max_F(h, h')$ is the corresponding maximum. For example, if A=1, Z=10, D=1, G=4, F={[1,2], [1,3], [1,4], [1,5], [2,6], [3,7], [4,8], [5,9], [6,10], [7,10], [8,10], [9,10]}, h=4 and h'=6, then $R_F(h) \cap R_F(h')$ = {[2,6], [3,7], [4,8]}. This set comprises just those features which would all be true of data values in the range [4,6]. Such a maximally informative abstraction (range

generalization) from the data values h=4 and h'=6 is what was desired. Notice that this intersection can be achieved by the same bit-wise logical product that was suggested for the all-or-none simple feature comparison problem. ORF organization thus is an effective representational basis for abstracting fuzzy commonalities from ordinal (temporal, spatial, amplitudinal, etc.) data.

The second principle for the organization of fuzzy match feature manifolds is that of a radius of generalization for value coding. Essentially, any observed value h of an attribute is generalized so that it is treated as if it were actually a range of values, $[h-\epsilon, h+\epsilon]$. For example if some scalar attribute had features for each of the values in F' ={1,2,...,10} and ϵ=3, and any data value h were represented as $R_{F',\epsilon}(h)$ = $\{f : f \epsilon F' \text{ and } f \in [h-\epsilon, h+\epsilon]\}$, then the data values h=4 and h'=6 would be repesented $R_{F',3}(h) = \{1,2,3,4,5,6,7\}$ and $R_{F',3}(h') = \{3,4,5,6,7,8,9\}$and the common abstraction would be $R_{F',3}(h) \cap R_{F',3}(h') = \{3,4,5,6,7\}$. Thus, the radius of generalization ϵ provides a basis for fuzzy matching of data like that previously considered with ORF manifolds. Here however, the manifold consists of simple feature values organized so that whenever the feature h is directly matched, all adjacent features within radius ϵ of h are also excited. This organization of features for value representation is referred to as a radial generalization (RG) manifold. The organization of an RG manifold provides a suggestive basis for interpretation of the observation that perceptual system ganglia and cortex are organized so that physically adjacent neurons have approximately equal receptive field values. As a result, the excitation of any neuron is likely to occur only if adjacent neurons are also excited to some extent.

Interestingly, both the ORF and RG manifolds produce equivalent repesentations and abstractions. This can be seen by comparing the abstractions $R_F(4) \cap R_F(6)$ and $R_{F',3}(4) \cap R_{F',3}(6)$ of the preceding examples. The first represented all data values in the range [4, 6]. The second abstraction, which was {3,4,....,9}, contains features that would all be true only of values in the range [4,6]. Thus, the generalization parameter G of the ORF manifold corresponds to the ϵ of the value generalization manifold. In either case, this radius of generalization corresponds to the amount of variability of data values which may be tolerated as perturbation or noise in evaluating the difference between an observed (measured) data value and the "true" underlying value. Alternatively, the radius of generalization can be interpreted as the maximum difference between the values of the same attribute which can be viewed as fuzzily equal. Given the equivalence of the two organizational principles, it is apparent that if ϵ is a function of the data value (for example, the error of measurement is often an increasing function of its magnitude), the corresponding generalizability parameter G of

the ORF manifold must also vary accordingly. The choice of a particular organization between these two types of manifold is largely arbitrary. Functionally, the ORF and RG manifold representations are not identifiably different, although it seems that RG representations may actually be easier to implement since all feature coding may be completed without comparisons between the observed data values and the boundaries of the receptive field intervals. The common strength of both representations is their ability to compensate for error of measurement and accomplish fuzzy comparisons of data values, up to a specified precision or JND, in ordinal or continuous scales by intersection of bit-vectors corresponding to discrete features.

While abstractions derived from any of the preceding representations will reflect only common, conjunctive characteristics of compared data, the third type of data representation, enumerative complementary coding, provides a basis for abstracting disjunctive characteristics of patterns through simple bit matching operations. The organizing principles of complementary feature manifolds are two: first, mutually exclusive attribute values are organized into sets called categories; and second, one feature is defined for every attribute which is and one is defined for every value which is not a property of the data to be coded. For example, in speech learning, the category of vowels might be V = {AH, AX, AO, EH, ER, ..., UH}. One feature might be assigned to represent the presence of each vowel in each example of an unknown pattern. Under enumerative complementary coding, features would also be assigned to represent the absence of each possible vowel type. Let the "positive" features be defined by the set V, and let the special feature ϵV represent that at least one vowel was detected. Let the complementary set of "negative" features be W = {$\neg$AH, $\neg$AX, $\neg$AO, ..., $\neg$UH}. Now consider comparing two data examples (B, AH, G) and (B, AX, G) of some unknown phonetic sequence pattern. If only positive feature codes were used, the two examples would exhibit features AH and AX, respectively, as well as the common feature ϵV. The comparison of these two descriptions would be just {ϵV}, representing only that each datum contained some vowel. Alternatively, suppose each of the examples were also described by complementary features. The data (B, AH, G) and (B, AX, G) would be described as C_V(B,AH,G) = {ϵV, AH, $\neg$AX, $\neg$AO, ..., $\neg$UH} and C_V(B,AX,G) = {ϵV, AX, $\neg$AH, $\neg$AO, ..., $\neg$UH}. A feature comparison would be C_V(B,AH,G) $\cap$ C_V(B,AX,G) = {ϵV, $\neg$AO, $\neg$EH, ..., $\neg$UH} = {ϵV} $\cup$ (W - {$\neg$AH, $\neg$AX}) = AH $\vee$ AX. That is, the result of forming the intersection of complementary codes of data based on a category V is to produce a set of negative features which exactly represents the disjunction of common positive features. It is interesting again to speculate upon the relationship between such feature manifolds and neural organizations. It is a well known fact that much of the perceptual system is organized in paired opponent

processes, consisting of two or more mutually inhibitory assemblages of neurons whose receptive fields are complementary (e.g., represent the mutually exclusive alternatives of light on vs. light off at the same locus). Simply stated, a complementary manifold may be conceived of as a set of mutually inhibitory detectors, where the "excitation state" of one feature in a category causes the "inhibited state" of all the others to be registered. Both types of detector states are considered as elements of the pattern description.

The last type of representation used is relational coding, which provides a basis for describing events in terms of objects, attributes, and relationships among objects. The basic elements of a relational representation are parameters, properties, and case frames. A parameter is a unique symbol which represents a constant or names an object in one or more relations. A property is a feature or attribute of an object. A case frame is a set of property:parameter terms which represents a relationship among the objects named by the parameters. A case frame is a generic type of relation and thus corresponds to an n-ary predicate. Instances of case frames, produced by substituting constants or object names for the generic parameters, are called case relations. Entire events are described by sets of case relations called parameterized structural representations (PSRs). A PSR is normally interpreted as a conjunction of the corresponding constituent predicates [6, 7, 9, 10].

A simple example will illustrate these concepts. Consider the problem of representing the pronunciation of the word "America." Using the ARPA speech understanding project phonetic alphabet, one pronunciation is (AX, M, EH', R, IH, K, AX'). This may be easily described in terms of the case frame {phone:x, begin:t_b, end:t_e, stress:k} meaning that the phone between times t_b and t_e is of type x and its stress is level k (0, 1, or 2). A PSR for the preceding "America" pronunciation would then be:

T = {{phone:AX, begin:t_1, end:t_2, stress:0},
{phone:M, begin:t_2, end:t_3, stress:0},
{phone:EH, begin:t_3, end:t_4, stress:2},
{phone:R, begin:t_4, end:t_5, stress:0},
{phone:IH, begin:t_5, end:t_6, stress:0},
{phone:K, begin:t_6, end:t_7, stress:0},
{phone:AX, begin:t_7, end:t_8, stress:1}}

This PSR is interpreted as follows: The word "America" is described by T as a pattern in which the unstressed phone AX spans the time interval t_1 to t_2, the unstressed phone M spans the interval t_2 to t_3, the maximally stressed phone EH spans the interval t_3 to t_4, etc. It should be noticed that the parameters t_1, ..., t_8 may

be interpreted either as constants or variables. In the former case, T would be a description of some sequence of phones from a particular time t_1 to a time t_8 containing the word "America." If these parameters are considered as variables, on the other hand, T represents a template for the word "America." If the description of some stimulus event S is given such that correspondents (parameters) in S can be found to bind to the variables t_1, ..., t_8 which insure that all case relations in T are true of S, S matches the template T and contains the word "America."

While it is usually true that pattern templates can be described by conjunctive sets of case relations as in the preceding example, it is sometimes desirable to represent disjunctive or negated case relations. In the framework of PSRs, this is done by augmenting the PSR to reflect component weights and overall thresholds for matching. For example, the series-parallel network of Fig. 1 represents the eight alternative pronunciations of "America" used in our speech work.

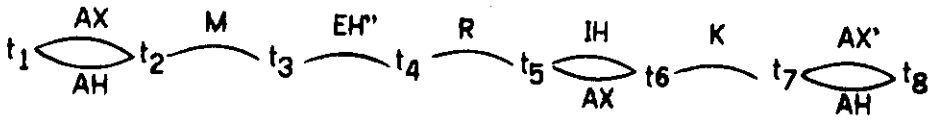

Figure 1. A network representation of eight pronunciations of "America"

Parallel paths represent acceptable phone alternatives at any point, while the left-to-right sequence of arcs represents a conjunctive set of necessary phones. The fact that either AX or AH may occur between t_1 and t_2 is represented by the PSR:

S_1 = {{phone:AX, begin:t_1, end:t_2, stress:0}=>+1,
{phone:AH, begin:t_1, end:t_2, stress:0}=>+1}≥1

This subtemplate asserts that if an unstressed AX occurs between times t_1 and t_2, +1 is to be added to the match count of S_1. Similarly, +1 is to be accumulated if AH occurs. The total match count is then checked to see if it is ≥ the specified PSR threshold of 1; if the threshold is equalled or exceeded, S_1 is matched, otherwise not. By convention, each case relation which is matched increments the match count of the containing PSR by 1 and the default threshold of a PSR with n case relations is n. Recursively, any matched PSR nested within another PSR counts as a single matched case relation and by default contributes 1 to the match count. Negated relations can be similarly represented by adding -1 to the match count for conditions that are matched but are intended not to be matched. Thus, the PSR corresponding to the template in Fig. 1 is just:

T' = {{{phone:AX, begin:t_1, end:t_2, stress:0},

{phone:AH, begin:t_1, end:t_2, stress:0}}≥1,
{phone:M, begin:t_2, end:t_3, stress:0},
{phone:EH, begin:t_3, end:t_4, stress:2},
{phone:R, begin:t_4, end:t_5, stress:0},
{{phone:IX, begin:t_5, end:t_6, stress:0},
{phone:AX, begin:t_5, end:t_6, stress:0}}≥1,
{phone:K, begin:t_6, end:t_7, stress:0},
{{phone:AX, begin:t_7, end:t_8, stress:1},
{phone:AH, begin:t_7, end:t_8, stress:1}}≥1}

The default threshold for T' is the sum of immediately nested PSRs or case relations which is 7. Thus, if each of the seven conditions is satisfied by some stimulus data description S, S contains the word "America." For example, noting that corresponding parameters in T and T' have been identically named, the pattern T which describes one particular pronunciation is seen to satisfy all the conditions of T'. Furthermore, it will be true that all specific occurrences of the word "America" will match T' and, thus, it would be conceivable to abstract the definition of T' directly from the common elements of the descriptions of such examples. Such comparisons of relational event descriptions require a general interference matching procedure that can identify which objects correspond and which case relations are true of corresponding objects in two compared events. One such procedure, SPROUTER [12], has been implemented, and it is discussed in the next section.

In summarizing the current section, it should be noted that a variety of methods exist for representing training data to facilitate the discovery of the criterial characteristics of patterns by noting the common properties of several examples of the same pattern. Feature representations permit comparison operations through bit matching operations, but relational representations apparently necessitate more complex matching procedures. Interestingly, some problems which at first view appear to require relational coding and matching can be solved by appropriate application of the feature manifold techniques. In particular, the complex pronunciation template T' can be directly inferred from examples which are simply described using an ordinal feature manifold and a complementary feature manifold for the temporal positions of phone labels [11]. Specifically, if the set of possible phone labels is P, let the ordinal feature manifold $F = \{(p,t) : p \in P \text{ and } t = 1,2,...,7\}$, where any feature (p,t) represents the occurrence of a phone p in temporal position t in an input sequence. Let the complementary manifold be $F' = \{(\in P,t), (\neg p,t) : p \in P \text{ and } t = 1,2,...,7\}$, where $(\neg p,t)$ means p is not a phone in temporal position t. If every example of the word "America" is represented by all appropriate features in F and F', the maximal abstraction of these examples will be a feature list equivalent to the PSR template T' (ignoring stress

properties). This may be easily seen by considering just the abstraction of two examples, E_1 = (AX, M, EH, R, IH, K, AH) and E_2 = (AH, M, EH, R, AX, K, AX) whose representations are denoted $R(E_1)$ and $R(E_2)$. The abstraction $E_1 * E_2 = R(E_1) \cap R(E_2)$ = $\{(\in P,1), (\neg p,1) : p \in (P-\{AX,AH\})\} \cup \{(M,2), (\in P,2), (\neg p,2) : p \in (P - \{M\})\} \cup \ldots \cup \{(\in P,7), (\neg p,7) : p \in (P - \{AH,AX\})\}$ = ((AX,1) ∨ (AH,1)) ∧ (M,2) ∧ (EH,3) ∧ (R,4) ∧ ((IH,5) ∨ (AX,5)) ∧ (K,6) ∧ ((AX,7) ∨ (AH,7)) = ((AX ∨ AH), M, EH, R, (IH ∨ AX), K, (AX ∨ AH)). If this abstraction were matched to any of the other six pronunciations, the result would be equal to $E_1 * E_2$. In this case, just two examples and simple bit-wise intersections would suffice to produce learning of a symbolic pattern equivalent to the seemingly complex PSR T'.

Thus, it is apparent that some learning problems are made particularly simple by applying the IM procedure to an appropriately organized feature-based description of examples. The main objective of the current section has been to develop a familiarity with the variety of representation schemes available and to suggest the importance of choosing a representation which is as simple as possible for any particular induction problem. Later in this paper, several more examples will be given of the use of feature manifolds as bases for description and learning.

4. INTERFERENCE MATCHING: ABSTRACTIONS AND RESIDUALS

Interference matching is the process of comparing two event descriptions to identify their commonalities and differences. Any set of properties which are common to two compared representations is an abstraction of them. A maximal abstraction comprises all properties common to the two. Properties which are true of one description but not the other constitute the residual of the first. Interference matching of any of the feature based representations can be performed by computng the set intersection of the property list descriptions. More simply, each attribute-value pair can be associated with a particular bit in a bit vector, and the abstraction of two descriptions is simply the set of features corresponding to the bits in the logical product of the two bit vectors. If E and F are the two bit vectors, E*F = E ∧ F is the maximal abstraction, and the residual of E with respect to E*F, E/E*F = E ∧ (¬F). In the next section, several results of using maximal abstractions of feature-based repesentations to learn patterns and syntactic categories are reported.

To perform interference matching of relational descriptions is far more difficult. Basically, since PSRs can describe graphs, the production of a maximal abstraction of

two PSRs is at least as difficult as finding a maximal subgraph of two graphs. Such problems are NP-complete, meaning that it is widely believed that this problem cannot be solved in an amount of time which is less than an exponential function of the number of case relations in the two compared PSRs. However, a heuristic program called SPROUTER has been implemented which performs interference matching and nearly always finds optimal abstractions in limited time and space [12].

The simplest way to understand interference matching in the framework of relational descriptions is as follows. Let E and F be two PSRs in which all parameters are variables.[1] Suppose $E = \{R_1, ..., R_m\}$ and $F = \{S_1, ..., S_n\}$ where each R_i and S_j is a case relation, and let the parameters of E and F be $P_E = \{e_1, ..., e_v\}$ and $P_F = \{f_1, ..., f_w\}$, respectively, where it is assumed that $|P_E| \leq |P_F|$. A maximal abstraction of E and F can be computed by forming a 1-1 binding function $B: P_E \rightarrow P_F$ such that each element $e \in P_E$ from the event E has the parameter $f = B(e) \in P_F$ as its correspondent in F. Ignoring alphabetic differences between corresponding parameters, the maximal abstraction of E and F under the binding function B is $E *_B F$, the set of case relations common to both E and F when corresponding parameters are treated as identical. As an example, if the events "Mary is a tall, dark, female" and "John is a tall, fair, male" are represented, respectively, by E = {{name:m, word:p}, {sex:m, value:s}, {female:s}, {height:m, value:g}, {complexion:m, value:j}, {"Mary":p}, {tall:g}, {dark:j}} and F = {{name:n, word:q}, {sex:n, value:t}, {male:t}, {height:n, value:h}, {complexion:n, value:k}, {"John":q}, {tall:h}, {fair:k}} and the binding function B = {(m,n), (p,q), (s,t), (g,h), (j,k)}, then $E *_B F$ = {{name:v_1, word:v_2}, {sex:v_1, value:v_3}, {height:v_1, value:v_4}, {complexion:v_1, value:v_5}, {tall:v_4}}. The residuals of E and F with respect to $E *_B F$ are $E/E*_BF$ = {{female:v_3}, {"Mary":v_2}, {dark:v_5}} and $F/E*_BF$ = {{male:v_3}, {"John":v_2}, {fair:v_5}}.

From this simple example, it is possible to see how the residuals of IM can be used to identify the elements of a category. Consider the parameters v_2, v_3, and v_5 which occur in both residuals. v_2, for example, is associated with the attribute values "John" in $F/E*_BF$ and "Mary" in $E/E*_BF$. The comparability of "John" and "Mary," which is indicated by the fact that they serve similar descriptive functions in E and F and results in their attribution as different properties of the same parameter in the abstraction $E*_BF$, suggests that they are both elements of the same syntactic category (say N). Thus, the hypothesis that there is a category N such that {"Mary", "John"} $\subset$ N represents an inference that properties which play comparable roles in different event

1 One may convert any constant c into a variable in a term such as attribute:c by replacing the term by the variable term, attribute:x, and adding a unary relation {c:x} to the PSR.

descriptions are syntactically substitutable for one another. While at the outset of learning the only category known may be the set of all words, W = {"a", "be", ..., "Joe", "John", ..., "Mary", ..., "zoo"}, the use of similar assertions (e.g., {name:v_i, word:v_j} where {"Joe":v_j} or {"John":v_j} or ... {"Mary":v_j) supports the inference that a particular subset of W is the category of names, N = {"Joe", "John", ..., "Mary"}. More complex bases for category induction are discussed in [9] and also in the next section.

This section concludes with a brief description of the relational interference matching program SPROUTER (see [12] for more details). The program accepts as inputs (1) a lexicon of case frames which specify the types of properties and case relations which are used to describe events and (2) two descriptions, E and F, which are sets of parameterized case relations (PSRs). It is presumed that preprocessing has been done so that all relevant properties are present in E and F and all parameters are variables. SPROUTER computes a number of distinct binding functions B_1, ..., B_k and, for each, outputs (1) a PSR corresponding to the maximal abstraction E $*_{B_i}$ F and (2) a special recognition network called an ACORN [13] which can be used to decide, for any other PSR S, if S matches E $*_{B_i}$ F. This is particulrly valuable if we are hypothesizing classification rules and must evaluate a rule's diagnosticity by determining how many training examples of each class match it.

The method SPROUTER employs is as follows. Suppose the PSR E has the smaller cardinality of E and F. One node in the ACORN is generated ("sprouted") for each generic abstraction of E which is also an abstraction of F. Originally, (terminal) nodes are created for each generic case frame present in both E and F. Iteratively, case relations A_E and B_E from two nodes A and B are selected and are conjoined to form a tentative higher-order binary node. This node represents the abstraction of E corresponding to the set of case relations {A_E, B_E}. If at least one corresponding pair of case relations in F can be found, the new node is permanently added to the sprouting ACORN. Otherwise, the tentative node is deleted. The process is limited by a best-first search of alternative binding functions after a breadth-first growth process has generated a pre-specified number of nodes. When the process terminates, the maximal nodes in the ACORN correspond to maximal abstractions of E and F.

Hayes-Roth and McDermott discuss the strengths and weaknesses of this particular method of IM and suggest directions for further improvements. The major issues considered are: (1) methods for increasing the amount of real-world knowledge that can be brought to bear in specific abstraction problems to increase SPROUTER's appreciation of the relative utilities of the various commmonalities it finds; and (2) the

need which arises in some contexts to generalize the concept of a binding function to permit one-many mappings between parameter sets. These issues, while very important, are simply beyond the scope of this paper.

5. KNOWLEDGE ACQUISITION PROCEDURES

In this section, algorithms are described for the three learning problems introduced in section 2 and, where possible, empirical results obtained using these methods are presented.

Pattern learning and classification. The algorithm used to solve these problems is called SLIM (Space Limited Interference Matching, [7]). Briefly, SLIM compares the examples of each pattern class using IM to develop maximal abstractions. Each abstraction is evaluated for its diagnosticity, and those which are expected to produce the greatest net (weighted) number of correct less incorrect judgments are given highest priority in the competition to persist in the limited memory space. Subsequently, novel test items are classified according to the most diagnostic abstraction they match. If a test item T matches no stored abstraction, a new abstraction T*A is generated from T and each stored abstraction A. Since T does match each T*A, T is classified in the class indicated by the most diagnostic T*A.

The example used to illustrate pattern learning through SLIM is taken from [11]. The problem is to classify speech syllable types from training examples which are sequences of sets of machine generated alternative labels for a series of acoustic segments. The difficult aspects of this problem include: (1) the syllable types are theoretical clusters of confusable speech patterns which do not necessarily share a close relationship with characteristics of the machine generated data themselves; (2) even if a theoretical syllable type consisting of n acoustic segment labels were valid, the machine segmentation of a spoken syllable frequently contains errors of insertion and deletion; and (3) in addition to the fact that the machne segmentation often inserts or deletes some segments compared to a theoretically perfect segmentation, the segmenter-labeller which generates hypothetical phonetic labels for each segment of speech makes many errors, including both assignments of multiple, incorrect labels and failures to assign the correct label to each segment. Thus, the type of training examples one might receive for the syllable corresponding to the theoretical sequence (M,EH,R) in "America" would be as follows (where multiple labels within column i represent alternative phonetic hypotheses for the i-th segment):

	SEGMENTS				
	1	2	3	4	5
Theoretical:	M	EH	R		
Example 1:	T B -	N M V F	AX EL ER	AX EL EH IH	M N R N
Example 2:	M N -	AX ER IH	B R G H		

The basic approach taken to this learning problem is to develop a method of description which relates the training data to the presumed underlying behavior generator. The model posits that the underlying knowledge is the sequence pattern (M,EH,R) and the result of applying the behavior generating function (the effect of speech plus the effect of machine segmentation and labelling) is to insert and delete segments as well as add incorrect additional labels and delete proper labels from the segments that remain. Thus, interference matching (SLIM) will be an effective learning procedure only if the feature coding employed is robust with respect to the sources of variability. Only in that case will an abstraction of training data retain criterial properties of the underlying pattern (knowledge).

The data repesentation employed in this case is a composition of both the ORF and the enumerative complementary coding feature manifold techniques. Each training example $E = (A_1, ..., A_k) = (\{a_{11}, ..., a_{1n_1}\}, \{a_{21}, ..., a_{2n_2}\}, ..., \{a_{k1}, ..., a_{kn_k}\})$, where $A_i = \{a_{i1}, ..., a_{in_i}\}$ contains the n_i alternative machine generated hypothetical labels assigned to the i-th segment, is coded as having the following features: $\{(a_{ij},i) : a_{ij} \in A_i, i = 1, ..., k\} \cup \{(\neg b,i) : b \text{ is a possible label not in } A_i\}$. To compensate for the expected error $|i-i'|$ between the theoretical position of the i-th segment and the position i' of the corresponding machine generated segment in the sequence $(A_1, ..., A_k)$, overlapping receptive fields are used. Because the expected error is an increasing function of i, the features used are $(a, [i-G_i,i+G_i])$ where the radius of generalization G_i is the smallest integer which is at least i/2. The labels used were the 44 acoustic labels of the Hearsay II speech understanding system [5].

SLIM was used to search for diagnostic abstractions of the training examples of the 12 most frequent syllable types. There were between 4 and 16 training examples

of each class and five novel test items per class (total test set size = 60). The segmenter-labeller used performs at about 60% accuracy at labelling and about 80% at segmentation (by a number of measures, see [5]). SLIM's performance in this task was 85% correct classifications of novel test items on the basis of the most diagnostic matched abstraction. A performance of 98% was achieved when SLIM was applied in its "filtered classification mode," in which the abstraction process is re-performed for each item to be classified and all intermediate abstractions are masked by the features which are present in the test item. These results can be compared to the performance of a refined, theoretically motivated syllable classifier which was specially designed and hand-tuned to perform the syllable recognition task in the Hearsay II system [16]. On exactly the same task, using the best matched syllable network template, the handcrafted knowledge source module achieved 68% correct. We think that this test provides impressive evidence of the capacity of the proposed techniques to achieve effective learning even in very noisy problem domains.

Rule learning. The algorithm used to induce quantified production rules from before-and-after examples of situations where the rule was applied is called SPROUTER [12]. Basically, as previously described, SPROUTER computes maximal abstractions of any two relational descriptions. In particular, if A_i is a PSR describing an antecedent situation which was transformed into a consequent situation described by the PSR C_i (i=1,...,n), SPROUTER compares the two-tuples (A_i, C_i) and (A_j,C_j) to produce the abstraction (A_i*A_j, C_i*C_j). The procedure used for rule induction is, roughly, to compute $(A_1*...*A_n, C_1*...*C_n)$ and infer that $A_1*...*A_n \Rightarrow C_1*...*C_n$.

As an illustration, consider the problem of inducing an unknown rule of transformational grammar from the following three antecedent-consequent example pairs.

(1) "The little man sang a lovely song." -->
"A lovely song was sung by the little man."
(2) "A girl hugged the motorcycles." -->
"The motorcycles were hugged by a girl."
(3) "People are stopping friendly policemen." -->
"Friendly policemen are being stopped by people."

The relational descriptions of these sentence pairs consist of three types of components: (1) syntactic phrase structures and markers (e.g., NUMBER:SINGULAR, TENSE:PRESENT); (2) a property which distinguishes elements of the antecedent sentence from elements of the consequent sentence (EVENT:e1 and ANTECEDENT:e1 as opposed to EVENT:e2 and CONSEQUENT:e2); and (3) same-type relations joining any pair of antecedent and consequent syntactic components which are identical types (i.e.,

are distinct tokens of the same type or, equivalently, are the roots of identical directed phrase structure graphs). The PSR for the first sentence pair is shown below.

```
{{ANTECEDENT:e1, CONSEQUENT:e2},
 {S:s1, NP:np11, VP:vp1, EVENT:e1},
 {S:s2, NP:np21, VP:vp2, EVENT:e2},
 {NP:np11, DET:the1, ADJ:little1, NOUN:noun11, EVENT:e1},
 {NP:np21, DET:a1, ADJ:lovely1, NOUN:noun21, EVENT:e2},
 {NOUN:noun11, NST:man1, NUMBER:n11, EVENT:e1},
 {NOUN:noun21, NST:song1, NUMBER:n12, EVENT:e2},
 {SINGULAR:n11, EVENT:e1},
 {SINGULAR:n12, EVENT:e2},
 {VP:vp1, AUX:aux11, VERB:verb11, NP:np22, EVENT:e1},
 {SAME!NP:np21, SAME!NP:np22},
 {NP:np22, DET:a2, ADJ:lovely2, NOUN:noun22, EVENT:e1},
 {SAME!NOUN:noun21, SAME!NOUN:noun22},
 {NOUN:noun22, NST:song2, NUMBER:n13, EVENT:e1},
 {SINGULAR:n13, EVENT:e1},
 {VP:vp2, AUX:aux12, PB:pb1, VERB:verb12, PP:pp1, EVENT:e2},
 {AUX:aux11, AUXST:have1, TENSE:t11, NUMBER:n15, EVENT:e1},
 {AUX:aux12, AUXST:have2, TENSE:t12, NUMBER:n16, EVENT:e2},
 {SAME!AUX:aux11, SAME!AUX:aux12},
 {VERB:verb11, VST:sing1, TENSE:t21, NUMBER:n15, EVENT:e1},
 {VERB:verb12, VST:sing2, TENSE:t22, NUMBER:n16, EVENT:e2},
 {SAME!VERB:verb11, SAME!VERB:verb12},
 {PB:pb1, PBST:be1, TENSE:t23, NUMBER:n16, EVENT:e2},
 {SAME!TENSE:t11, SAME!TENSE:t12},
 {SAME!TENSE:t21, SAME!TENSE:t22, SAME!TENSE:t23},
 {SINGULAR:n15, EVENT:e1},
 {SINGULAR:n16, EVENT:e2},
 {PRESENT:t11, EVENT:e1},
 {PRESENT:t12, EVENT:e2},
 {PAST-PART:t21, EVENT:e1},
 {PAST-PART:t22, PAST-PART:t23, EVENT:e2},
 {PP:pp1, PREP:by1, NP:np12, EVENT:e2},
 {SAME!NP:np11, SAME!NP:np12},
 {NP:np12, DET:the2, ADJ:little2, NOUN:noun12, EVENT:e2},
 {SAME!NOUN:noun11, SAME!NOUN:noun12},
 {NOUN:noun12, NST:man2, NUMBER:n14, EVENT:e2},
 {SAME!NUMBER:n11, SAME!NUMBER:n12, SAME!NUMBER:n13,
  SAME!NUMBER:n14, SAME!NUMBER:n15, SAME!NUMBER:n16},
 {SINGULAR:n14, EVENT:e2},
 {THE:the1, EVENT:e1},
 {THE:the2, EVENT:e2},
 {SAME!WORD:the1, SAME!WORD:the2},
 {LITTLE:little1, EVENT:e1},
 {LITTLE:little2, EVENT:e2},
 {SAME!WORD:little1, SAME!WORD:little2},
 {MAN:man1, EVENT:e1},
```

{MAN:man2, EVENT:e2},
{SAME!WORD:man1, SAME!WORD:man2},
{HAVE:have1, EVENT:e1},
{HAVE:have2, EVENT:e2},
{SAME!WORD:have1, SAME!WORD:have2},
{SING:sing1, EVENT:e1},
{SING:sing2, EVENT:e2},
{SAME!WORD:sing1, SAME!WORD:sing2},
{A:a1, EVENT:e1},
{A:a2, EVENT:e2},
{SAME!WORD:a1, SAME!WORD:a2},
{LOVELY:lovely1, EVENT:e1},
{LOVELY:lovely2, EVENT:e2},
{SAME!WORD:lovely1, SAME!WORD:lovely2},
{SONG:song1, EVENT:e1},
{SONG:song2, EVENT:e2},
{SAME!WORD:song1, SAME!WORD:song2},
{BE:be1, EVENT:e2},
{BY:by1, EVENT:e2}}

This corresponds to Fig. 2. The PSRs for all 3 sentence pairs were supplied to SPROUTER, which abstracted the structure illustrated in Fig.3.

---

insert Figs.2-3 here

Here, same-type relations which were common to all examples have been retained and are represented by arrows linking subgraphs in the antecedent sentence structure with corresponding subgraphs in the consequent sentence structure. These correspondences represent identical quantified variables in the left and right-hand ssides of the inferred production. When the rule is applied, the actual parameter in a stimulus event which matches the antecedent will be bound to the corresponding quantified variable and should be substituted into the corresponding locus of the consequent structure. If this is done, the inferred production will effect the active-to-passive transformation rule.

How general is such a rule learning paradigm? In my opinion, all rule learning will correspond, in part, to the preceding methodology. The basic elements of the induction procedure are: (1) a set of antecedent-consequent examples $I = \{(A_i, C_i)\}$; (2) instantiation of a set of predicates which can describe the criterial properties, relations, and common subpattern types of each exemplar pair; (3) an interference matching algorithm to identify the rule $F = [A_i * ... * A_n \Rightarrow C_i * ... * C_n]$; (4) a method for evaluating the goodness of the rule F by ascertaining the diagnosticity of the rule, i.e., the extent to which F is a reliable description of the way in which each A_i can be

altered to produce the corresponding C_i; and (5) a method of implementing the inferred rules as universally quantified productions so that the most reliable rules execute first. The keystone of this framework is (2), the instantiation of criterial predicates. Of course, in any reasonably complex learning environment, either there may be an excessive number of potentially criterial properties to evaluate or, worse, the properties that are criterial to the rule may themselves be yet unknown (undiscovered, undefined). In the former situation, heuristics must be used to evaluate only the most promising properties first. In the latter case, discovery of criterial properties must precede or, at least, occur simultaneously with discovery of an unknown rule. Such a situation arises when, for example, one wishes to infer Newton's law F = ma and any of the concepts of force, mass, acceleration or multiplication is unknown. In the next subsection a similar problem, that of discovering syntactic categories, is considered in some detail.

Category and syntax learning. The algorithm used to discover syntactic categories attempts to invert the assumed syntactic behavior generator. For the sake of simplicity, assume surface strings of words are generated by context free rules in Chomsky normal form, such as D → EF. That is, words or phrases in category E precede words or phrases in category F when generated as respellings of category D. To induce categories of single words from a corpus of text $T = t_1 t_2 ... t_L$, one must identify systematic sequential constraints on word sets. Suppose two sets of words $W = \{w_1,...,w_m\}$ and $Y = \{y_1, ..., y_n\}$ exist such that $(\forall i,j)$ $Pr[t_{k+1}=y_j \in Y \mid t_k=w_i \in W]$ is significantly better than chance. In that case, one may reasonably infer that W and Y are subsets of some categories E and F such that, for some D, D → EF is a rule of the language. (This may also re represented as the probabilistic rule E => F.) If W and Y are maximal, in the sense that the addition of any word to either one reduces the significance of the prediction of Y from W, a reasonable inference is that D → WY is a rule of the language. Once such inferences are made, the categories W and Y become unary predicates W(w) and Y(y); i.e., the category name is a property of any word which is contained in the corresponding set. Subsequent inferences may depend upon discovery of categories of categories, n-ary relations of categories, or sequences of categories [9]. For example, if the categories of determiner (words that precede adjectives, numbers, and nouns), adjective (words that precede numbers and nouns and succeed determiners), and noun (words that follow determiners, adjectives, and numbers and precede verbs) have been induced, the rule NP → (determiner)(adjective)(number)noun can be inferred from the fact that all eight possible sequences of these categories reliably precedrelative qualifying pronouns (e.g., which, who, that, ...) as well as precede and succeed verbs.

Three algorithmic approaches to this sort of induction problem are being pursued. First, Rich [14] showed that fairly effective categorizing of words could be accomplished simply by clustering words according to proximity in Euclidean N-space where each word w_i was represented by a vector of N numbers $(v_1,...,v_N)$ where v_j represented the probability that word w_j followed w_i in the training corpus. This method has two chief problems. First, clustering must be heuristically controlled either by some arbitrary proximity criterion or by some predetermined number of clusters. Secondly, the clustering can assign any word token to at most one category (cluster). Thus, distinct syntactic roles or senses of the same word token cannot be found. The second approach to this problem aims to correct these deficiencies by attempting to compute directly the probability that any two words are in one of the same categories by computing the extent to which the two words have significant concordances of predecessors and successors. Once hypothesized subsets of size n are formed, the algorithm is iterated to attempt to combine these into sets of size n+1 which preserve the significant concordance of successors and predecessors (n=2,3,...).

Finally, the third approach being pursued employs the enumerated complementary coding technique and IM to generate category inferences. Each sequential pair of words $S = w_iw_j$ from the training corpus is represented by the set of properties $R(S) = \{(w_i,1), (w_j,2)\} \cup \{(\neg w_{i'},1), (\neg w_{j'},2) : i' \neq i, j \neq j'\}$ which, in turn, is efficiently represented by the bit string $B(S) = (b_{11}, ..., b_{1M}, c_{11}, ..., c_{1M}, b_{21}, ..., b_{2M}, c_{21}, ..., c_{2M})$, where $b_{hk}=1$ only if (w_k,h) is a feature and $c_{hk}=1$ only if $(\neg w_k,h)$ is a feature of R(S). Then, the bit-wise logical product of B(S) and B(T) = S*T represents the common information about sequences S and T. For example, if S = (a, dog), T = (the, dog), then S*T represents ((a v the), (dog)). The statistical information relevant to assessing the goodness (diagnosticity) of the inference that the category {a,the} predicts the category {dog} includes F({a}), F({the}), F({dog}) and F({dog} | {a,the}) which are, respectively, the frequencies of the words "a," "the," and "dog" and the frequency with which the word "dog" follows the words "a" or "the." One possible measure of the goodness of the inference that some rule of grammar is X → {a, the} {dog} then is M({dog} | {a,the}) = F({dog} | {a,the}) / expected F({dog} | {a,the}) where the expected F({dog} | {a,the}) = F({dog}) F({a,the}) / N and N is the total number of words in the training corpus. Basically, this measure M(V|U) is the ratio of the observed positive frequency of the sets U followed by V divided by the expected (chance) frequency of such sequences.

If the goodness measure M replaces the expected net number of correct classifications (the performance) of hypothetical classificatory rules, if inferred syntax

rules are represented by bit string products of the sort described above and ordered so that the best rules occur highest in the working list of intermediate abstractions, and if the observed positive frequency of a rule is conditionalized upon (reduced by) the postive frequency of the higher performing rules with which it is redundant (see [7]), the SLIM algorithm is effective for finding categories and related syntactic rules. As an illustration, the following corpus of partial sentences (noun phrases) were described using enumerative complementary codes and supplied for training of categories.

s_1	a dog ...
s_2	the dog ...
s_3	a cat ...
s_4	the cat ...
s_5	a big dog...
s_6	a green cat ...
s_7	the yellow dog ...
s_8	the big cat ...

The following sequences of categories were inferred and are written here as prediction rules:

RULE	GOODNESS	INTERPRETATION
{big,green,yellow} => {dog,cat}	2.5	adjective => noun
{a,the} => {big,green, yellow,dog,cat}	1.67	determiner => adjective ∨ noun

Note that these two rules account for all 12 sequential pairs of words in the training set. The first rule accounts for 4 pairs of words when a chance co-occurrence would yield 1.6 sequences of these words. The second rule accounts for 8 sequential pairs of words when 4.8 are expected. All other possible rules are lower performing and redundant with these.

A full investigation of such syntactic inference is beyond the scope of this paper. It would be necessary to augment training descriptions by properties corresponding to membership of words in inferred categories and then to reiterate this non-supervised learning procedure. This would facilitate discovery of rules based on higher-order syntactic relations as in transformational grammar. For the present purposes, it suffices to say that the outlook is bright for learning categories and rules of syntax by simple generalization of the methods which are proving so useful for the induction of patterns and rules from examples.

6. FUTURE DIRECTIONS

Within the framework of the present paper, it is easy to understand where the chief obstacles and most promising applications of such learning techniques are likely to be in the near future. The most difficult problems remaining to be solved concern the combinatorics of relational matching and the need to restrict the evaluation of potential predicates to the most criterial ones first. The possibility of finding a feature manifold representation for general relational structures (like those found for disjunctions and sequences) must be considered a significant goal. The wide-scale application of the interference matching algorithm to many AI and PR learning problems is apparently warranted. Both in areas where much theoretical knowledge exists about potentially criterial properties and in areas where combinations of large numbers of primitive features need to be evaluated as possible bases for pattern description, interference matching provides an effective technique for abstracting patterns and rules from examples.

REFERENCES

1. Burge, J., and Hayes-Roth, F. A novel pattern learning and recognition procedure applied to the learning of vowels. Proc. 1976 IEEE Intl. Conf. Acoustics, Speech, and Signal Processing, 1976.
2. Diday, E., Schroeder, A., and Ok, Y. The dynamic clusters method in pattern recognition. Proc. IFIP Congress, 1974.
3. Fikes, R. E., and Nilsson, N. J. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence 2, (1971), 189-208.
4. Galton, F. Inquiries into Human Faculty and its Development. Dent, London, 1907.
5. Goldberg, H. A comparative evaluation of parametric segmentation and labelling strategies. Unpublished doctoral dissertation, Department of Computer Science, Carnegie-Mellon University, 1976.
6. Hayes-Roth, F. A structural approach to pattern learning and the acquisition of classificatory power. Proc. First Intl. Jt. Conf. Pattern Recognition, 1973.
7. Hayes-Roth, F. Schematic classification problems and their solution. Pattern Recognition 6, 2 (Oct. 1974), 105-114.
8. Hayes-Roth, F. An optimal network representation and other mechanisms for the recognition of structured events. Proc. Second Intl. Jt. Conf. Pattern Recognition, 1974.
9. Hayes-Roth, F. Uniform representations of structured patterns and an algorithm for the induction of contingency-response rules. Information and Control, (in press).
10. Hayes-Roth, F. Representation of structured events and efficient procedures for their recognition. Pattern Recognition (in press).

11. Hayes-Roth, F., and Burge, J. Characterizing syllables as sequences of machine-generated labelled segments of connected speech: a study in symbolic pattern learning using a conjunctive feature learning and classification system. Working paper, Department of Computer Science, Carnegie-Mellon University, 1976.
12. Hayes-Roth, F., and McDermott, J. Knowledge acquisition from structural descriptions. Working paper, Department of Computer Science, Carnegie-Mellon University, 1976.
13. Hayes-Roth, F., and Mostow, D. J. An automatically compilable recognition network for structured patterns. Proc. Fourth Intl. Jt. Conf. Artificial Intelligence, 1975.
14. Rich, E. Personal communication, 1975.
15. Shortliffe, E. H. MYCIN: Computer-based medical consultations. New York: American Elsevier, 1976.
16. Smith,A. R. Word hypothesization in the Hearsay II speech understanding system. Proc. 1976 IEEE Conf. Acoustics, Speech, and Signal Processing, 1976.

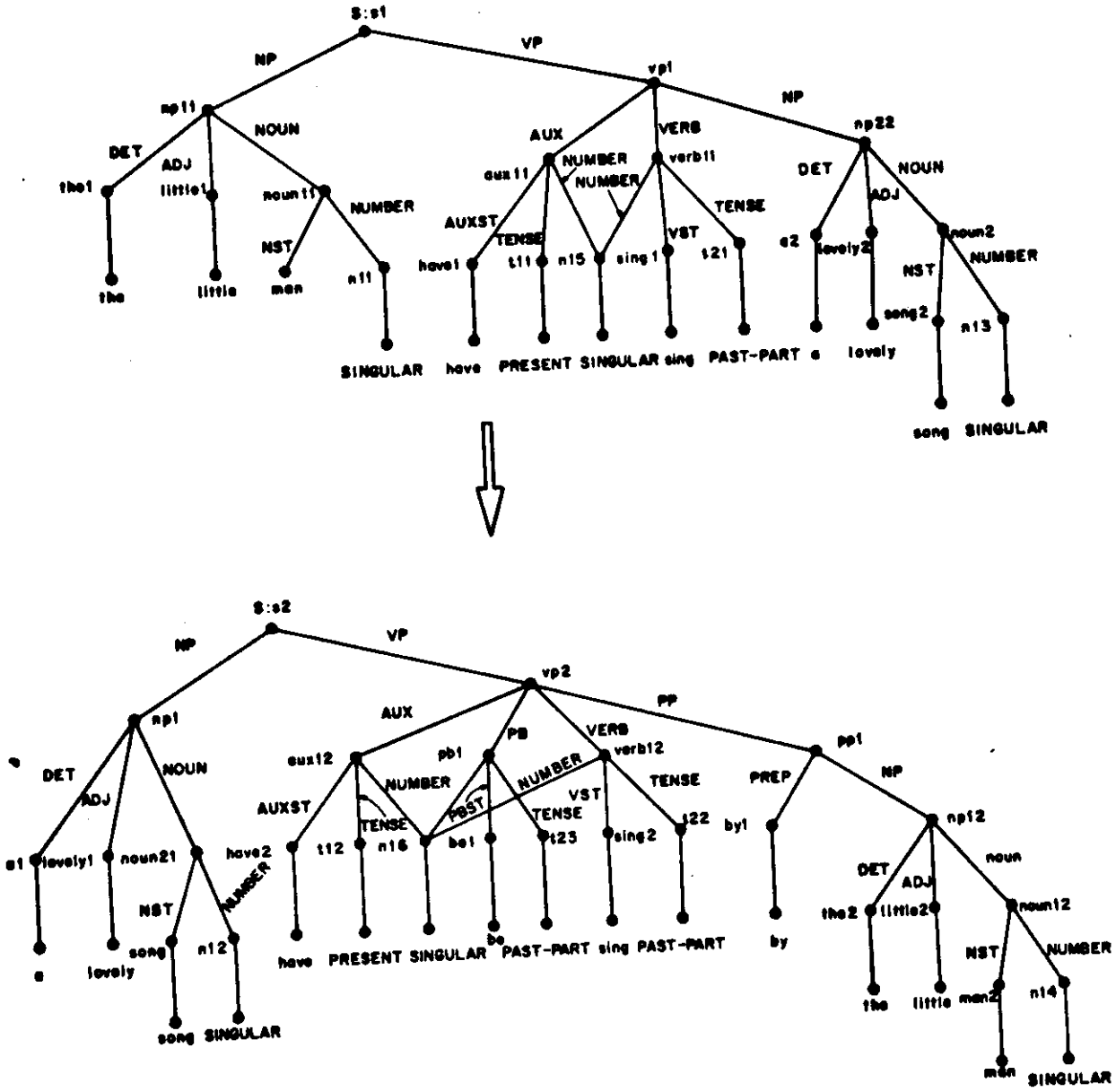

Figure 2. Graphical representation of the first example of the active-to-passive transformational rule.

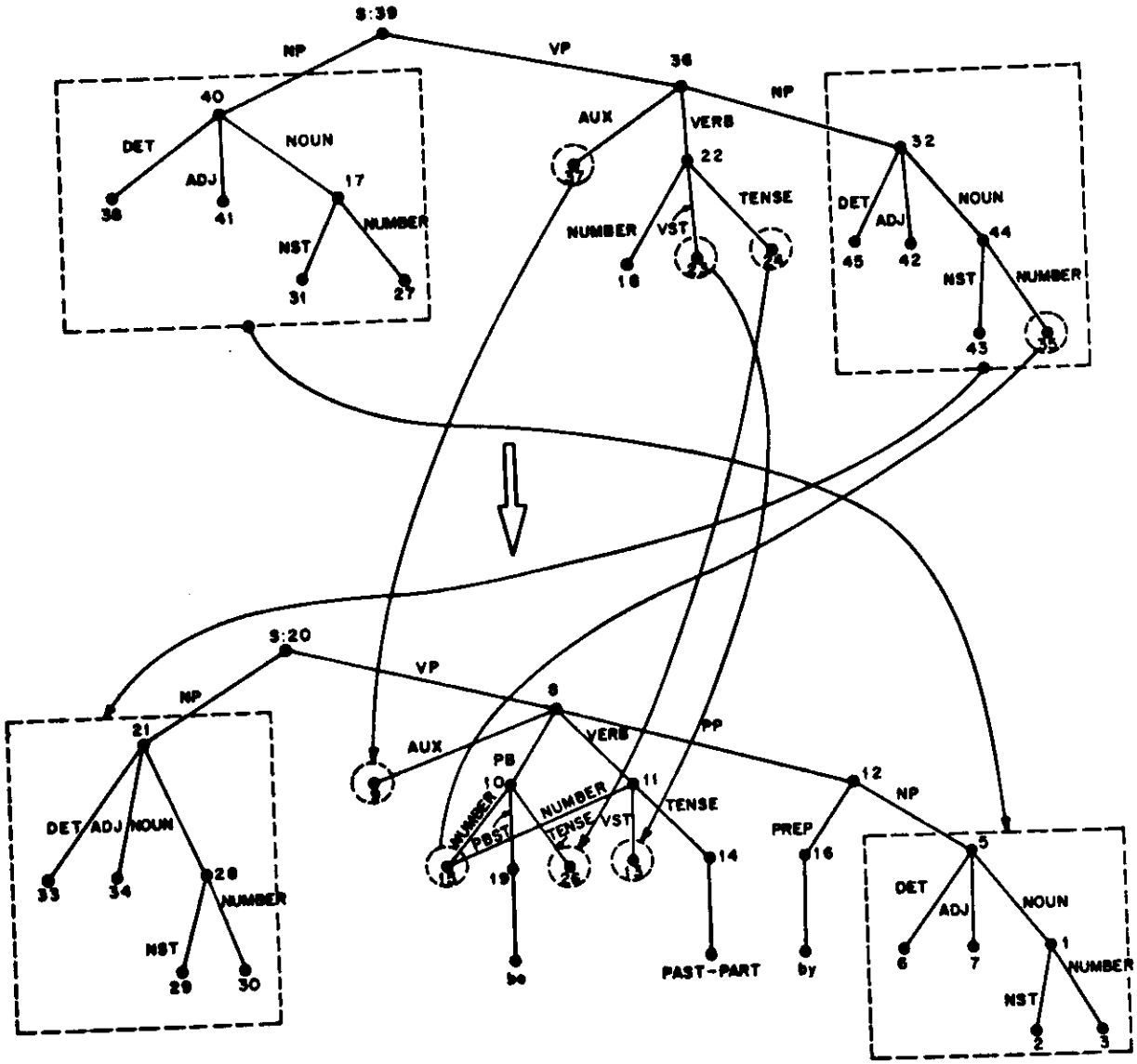

Figure 3. Graphical representation of the induced active-to-passive transformational rule.