Machine Learning 4:7–40, 1989

Learning Conjunctive Concepts in Structural Domains

DAVID HAUSSLER (HAUSSLER@SATURN.UCSC.EDU)
Department of Computer Science, University of California, Santa Cruz, CA 95064 USA

Editor: Tom Mitchell

Keywords: concept learning, PAC learning, empirical learning, VC dimension, version spaces, structural domains, conjunctive concepts.

Abstract. We study the problem of learning conjunctive concepts from examples on structural domains like the blocks world. This class of concepts is formally defined, and it is shown that even for samples in which each example (positive or negative) is a two-object scene, it is NP-complete to determine if there is any concept in this class that is consistent with the sample. We demonstrate how this result affects the feasibility of Mitchell's version of space approach and how it shows that it is unlikely that this class of concepts is polynomially learnable from random examples alone in the PAC framework of Valiant. On the other hand, we show that for any fixed bound on the number of objects per scene, this class is polynomially learnable if, in addition to providing random examples, we allow the learning algorithm to make subset queries. In establishing this result, we calculate the capacity of the hypothesis space of conjunctive concepts in a structural domain and use a general theorem of Vapnik and Chervonenkis. This latter result can also be used to estimate a sample size sufficient for heuristic learning techniques that do not use queries.

1. Introduction

Since the publication of Winston's results on learning blocks-world concepts from examples (Winston, 1975), considerable effort has gone into improving and generalizing his learning algorithm and into developing a more rigorous and general model of this and related AI learning problems (Vere, 1975; Hayes-Roth, 1978; Knapman, 1978; Michalski, et al., 1983; Dietterich, 1983; Bundy, et al., 1985; Sammut, 1986; Kodratoff, 1986). Whereas much of the earlier learning work, especially that associated with the field of Pattern Recognition (e.g. Duda, 1973), relied on an *attribute-based* domain in which each instance of a concept is characterized solely by a vector of values for a given set of attributes, this work uses a *structural* domain in which each instance is composed of many objects and is characterized not only by the attributes of the individual objects it contains, but by the relationships among these objects. The classic example is Winston's arch concept, defined as any scene that contains three blocks, two having the attributes required of posts and a third having the attributes required of a lintel, with each of the posts supporting the lintel and the posts set apart from each other. This concept can be formalized by inventing variables x and y for the posts and z for the lintel and giving an expression in the predicate calculus roughly of the form

"there exist distinct x, y, z such that f_1 and f_2 and . . . f_s," where the f_i's are atomic formulae in the variables x, y and z that describe attributes of and relations between the objects represented by these variables. A concept of this type will be called an *existential conjunctive concept.* The notions of an *instance space* in a structural domain and the class of existential conjunctive concepts over this instance space are defined formally in Section 1 below. Instances in the instance space will be called *scenes* in deference to the pioneering work of Winston, even though our treatment is by no means limited to a blocks-world-like domain. Instances and concepts will be represented by labeled graphs, as described in Section 1.

Mitchell (1982) gives an elegant framework for viewing the process of learning from examples and illustrates this framework by analyzing the process of learning simple existential conjunctive concepts. A general version of this framework can be described as follows. Let us assume that we are trying to learn some unknown target concept defined on the instance space. This concept may or may not be an existential conjunctive concept (i.e., the target concept is allowed to be any subset of the instance space). We are given a sequence of *examples* of this target concept, each of which is either an instance contained in (i.e., satisfying) the concept (a *positive* example) or an instance not contained in the concept (*negative* example), each labeled accordingly. This is called a *sample* of the target concept. The task is to produce an existential conjunctive concept that is consistent with the sample, in that it contains all instances from positive examples and none from negative examples, or to detect when no existential conjunctive concept is consistent with the sample. Thus we assume a restricted *hypothesis space* H consisting of only existential conjunctive concepts. The set of all *hypotheses* $h \in H$ that are consistent with the sample is called the *version space* of the sample (with respect to the hypothesis space H). The version space is empty in the case that no hypothesis in H is consistent with the sample.

Mitchell shows how this learning task (and related tasks) can be solved by maintaining only two subsets of the version space: the set S of the most specific hypotheses in the version space and the set G of the most general hypotheses. These sets are updated with each new example. There are two computational problems associated with this method. The first is that in order to update the sets S and G, we must have an efficient procedure for testing whether or not one hypothesis is more general than another and whether or not a hypothesis contains a given instance. Indeed, the latter would seem to be a requirement for the existence of any practical learning method. Unfortunately, both of these problems are NP-complete if we allow arbitrarily many objects in scenes and arbitrarily many variables in existential conjunctive hypotheses (see Hayes-Roth, 1978 and Section 1 below). This problem is avoided by fixing the maximum number of objects in a scene (and hence variables in a consistent concept) to a reasonably small number. For example, Mitchell uses two objects per scene in the running example of (Mitchell, 1982).

The second problem is that the size of the sets S and G can become unmanageably large. In Haussler (1988), it is shown that using the hypothesis space of conjunctive concepts in a simple attribute-based domain (corresponding to existential con-

junctive concepts on scenes with only one object), the size of G can already grow exponentially in the number of examples if the number of attributes is large. However, in this case S never contains more than one hypothesis (see Bundy, et al., 1985), so a consistent (and maximally specific) hypothesis can be found by computing only S (using the positive examples) and then checking to see if any negative example is contained in S in a second pass through the sample. This is not possible in structural domains. In fact, we show that *both* the size of S and the size of G can grow exponentially in the number of examples when structural domains are used. More precisely, even if we restrict ourselves to instance spaces like the one in Mitchell's paper in which

1. each scene has exactly two objects,
2. there are no binary relations defined between the objects, and
3. each object has only Boolean-valued attributes,

then using the hypothesis space of existential conjunctive concenpts and letting the number of attributes grow, the size of both S and G grow exponentially in the number of examples. Furthermore, in this case it is NP-complete to determine if the version space is nonempty, i.e., if there is any existential conjunctive concept consistent with a given sample (Theorem 1, Section 2).

The version space paradigm of learning from examples is a rather demanding one in that it aims at either exact identification of the target concept or an exact description of the set of consistent hypotheses in the case that the number of examples is insufficient for exact identification. Another paradigm has recently been introduced by Valiant in which the goal of learning is merely to find a hypothesis that is a good approximation to the target concept in a probabilistic sense (Valiant, 1984). This framework has been called "Probably Approximately Correct" (PAC) learning (Angluin, 1988). We describe this framework in Section 3 and consider the problem of learning existential conjunctive concepts in this sense.

It turns out that using results from (Pitt, 1986), Theorem 1 implies that existential conjunctive concepts are not learnable in the strict PAC sense, from random examples alone (Proposition 2, Section 3.1). In Section 3.2 we use results of Vapnik and Chervonenkis to show that the problem is not that too many training examples are needed, but only that it is, in general, computationally difficult to find a hypothesis that is consistent (or nearly consistent) with the training examples. We show that for reasonable sample sizes, any hypothesis that is consistent with all but a small fraction of the training sample is good with high probability (Propositions 3 and 4, Section 3.2). This implies that heuristic techniques, such as those considered in Vere (1975), Hayes-Roth (1978), Michalski (1983), Dietterich (1983), and Sammut (1986), will be effective in producing accurate hypotheses, so long as the training set is reasonably large, and they do not run into computational difficulties (Theorem 2, Section 3.2).

Finally, in Section 3.3 we show how these computational difficulties can be overcome if we allow the learning algorithm to make certain types of queries while

it is learning, in addition to receiving random training examples. The type of query we consider is a *subset query,* as defined in Angluin (1988). Here the learning algorithm asks if a hypothesis is contained in (i.e., equal to or more specific than) the target concept. We assume an oracle is available that answers yes or no. We show that existential conjunctive concepts are PAC learnable using subset queries and random examples, so long as we restrict the maximum number of objects per scene to a fixed constant (Theorem 3, Section 3.3). We close in Section 4 with a number of open problems.

2. Attributes, relations, and existential conjunctive concepts

2.1. Attributes and relations

We define a set of attributes for which each object we consider has particular values. For example, we might have attributes **shape, color,** and **size,** and a particular object (a small red square) might be characterized as having the value *square* for the attribute **shape,** *red* for **color** and 2 for **size.** The values an attribute can have are defined *a priori,* as is its *value structure,* which may be either *tree-structured* or *linear* (Michalski, 1983). In a tree-structured attribute, the values are ordered hierarchically as illustrated in Figure 1a for the attribute **shape.** The lowest or *leaf* values of this tree are the only *observable* values, i.e., actual objects must have one of these values for the attribute shape. The other values, called *abstract* values, are used only in logical formulae that represent concepts, as defined below. We assume that the node for each abstract value in the tree has at least two children. The values of a linear attribute are all directly observable and are linearly ordered, as in the attribute **size,** which may be defined, for example, to take only integer values between 1 and 5. At the other extreme, a linear attribute may be defined to take on any real number as its value.

A scene that contains several objects is characterized not only by the attributes of its objects but by the relations between its objects. Here we will restrict ourselves to binary relations, but, for consistency with our treatment of attributes (henceforth viewed as unary relations), we will allow these binary relations to take on any of several values, with the same two types of possible value structures. To illustrate the flexibility of this model, we give a few examples of binary relations that might be used to characterize the spatial relationship between an ordered pair of objects in a two-dimensional scene. First, the relation **distance-between** may be defined as a linear binary relation in analogy with the attribute **size,** perhaps using the Euclidean distance between the centers of mass. In addition, the relative position in the x-y plane of two objects might be characterized similarly using two linear binary relations *delta_x* and *delta_y* that give the difference in x coordinates and the difference in y coordinates of the centers of mass. Alternatively, a more qualitative binary relation to describe spatial relationship is given by the tree-structured relation **rel_pos** illustrated in Figure 1b.

Attributes (Michalski '83)

size: 1, 2, ..., 5 **(linear)**

shaded: yes, no **(Boolean)**

shape: (tree-structured)

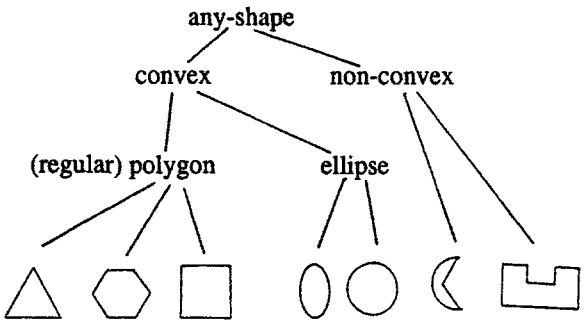


Fig. 1a. The value structures for the attributes **size, shaded,** and **shape.**

Binary Relations:

dist-between: touching, close, far **(linear)**

rel-pos:

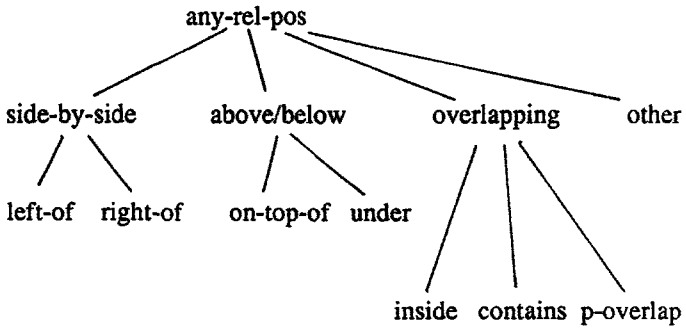
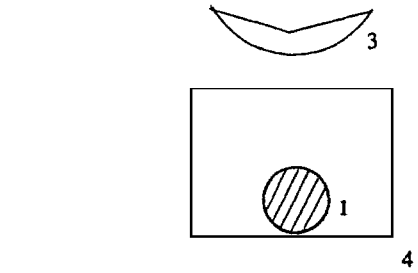


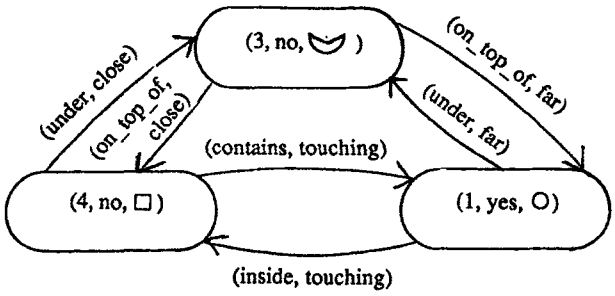
Fig. 1b. The value structures for the relations **dist-between** and **rel-pos.**

2.2. *Instances and existential conjunctive concepts*

Henceforth we will assume a fixed set R of relations consisting of n attributes $A_1, \ldots, A_n$ and l binary relations $B_1, \ldots, B_l$. We will also assume a fixed upper bound k on the number of objects per scene. A scene with t objects, $0 \leq t \leq k$, will be represented as a complete directed graph on t nodes (i.e., there are two directed edges between every pair of nodes, one going each way), with each node representing an object in the scene and labeled by the n-tuple that gives the observed value of each attribute for that object, and a directed edge from the node representing obj_1 to the node representing obj_2 labeled with an l-tuple that gives the observed values of each binary relation on the ordered pair (obj_1, obj_2). A graph of this type will be called an *instance graph*. This representation is illustrated in Figure 2, where the triples in the nodes give the values of the attributes **size, shaded,** and **shape,** respectively, and the pairs on the edges give the values of the relations **rel_pos** and **distance_between,** respectively. Note that this representation implies that binary relations are not defined between an object and itself; this is



an instance



its graph representation

Fig. 2. Graph representation of scenes (numbers denote size values).

reserved for attributes. Finally, the *instance space* is defined as the set of all instance graphs. Here, for simplicity, we identify scenes with the labeled graphs that represent them.

By using variables to denote unknown objects, we can define the set of (elementary) *atomic formulae* over R as in (Michalski, 1983). Atomic formulae are either *unary* or *binary*. A unary atomic formula $f(x)$, where x is a variable, has either the form $(A(x) = v)$, where A is a tree-structured attribute in R and v is a value of A, or the form $(v_1 \leq A(x) \leq v_2)$ where A is a linear attribute in R and v_1, v_2 are values of A such that $v_1 \leq v_2$. In the former case the atomic formula $f(x)$ restricts the value of A for the object x to be in the set of observable values in the tree for A that lie in the subtree below v, including v itself if v is observable. For example, the atomic formula *shape*(x) = *convex* restricts x to having shape *triangle, hexagon, square, proper-ellipse,* or *circle* (see Figure 1a). In the latter case the value of A is restricted to be between v_1 and v_2, inclusive, with respect to the linear order on A. An object *satisfies* $f(x)$ if its value for the attribute A complies with these restrictions.

A similar semantics applies to binary atomic formulae. Thus a binary atomic formula $f(x, y)$, where x and y are distinct variables, has either the form $(B(x, y) = v)$, where B is a tree-structured binary relation in R and v is a value of B, or the form $(v_1 \leq B(x, y) \leq v_2)$ where B is a linear binary relation in R and v_1, v_2 are values of B such that $v_1 \leq v_2$. An ordered pair of objects (obj_1, obj_2) in a scene satisfies the atomic formula $f(x, y)$ if the binary relation between these objects has the appropriate value, as defined above for unary relations.

An *existential conjunctive expression* over R (see Figure 3) is a formula ϕ of the form

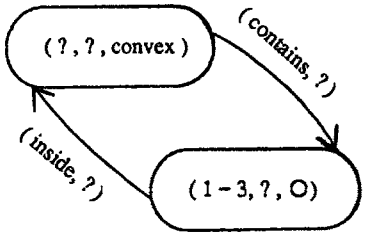
$$\exists^* x_1, \ldots, x_r : f_1 \text{ and } f_2 \text{ and } \cdots \text{ and } f_s,$$

where $s \geq 1$, each x_j, $1 \leq j \leq r$, is a variable and each f_i, $1 \leq i \leq s$, is an atomic formula over R involving either a single variable from $\{x_1, \ldots, x_r\}$ or an ordered pair of distinct variables from this set. We have dropped the names of the variables appearing in the individual atomic formulae to simplify the notation. The first part of this expression (up to the colon) may be read "there exist distinct objects x_1 through x_r such that . . ." Thus a scene (or its instance graph) *satisfies* ϕ if it contains r *distinct* objects $obj_1 \ldots, obj_r$ such that for every i, $1 \leq i \leq s$, if $f_i = f_i(x_j)$ then obj_j satisfies f_i and if $f_i = f_i(x_j, x_k)$ then the ordered pair (obj_j, obj_k) satisfies f_i. Note that the scene may also contain objects other than these r objects (see Stepp, 1987).

The set of all instance graphs that satisfy ϕ is called the *concept* represented by ϕ. The class of all such concepts (varying ϕ) is referred to as the class of *existential conjunctive concepts.* Different existential conjunctive expressions can represent the same concept. However, each nonempty existential conjunctive concept can be associated with an existential conjunctive expression that is unique up to renaming of the variables and rearranging the order of the atoms. Before describing

"$\exists^*$ x,y : (shape(x) = circle) and (1 ≤ size(x) ≤ 3)
and (shape(y) = convex) and (rel-pos(x,y) = inside)
and (rel-pos(y,x) = contains)"

an existential conjunctive expression



its concept graph

Fig. 3. Graph representation of concepts.

this canonical form, let us say that an atomic formula is *useless* if it is satisfied by any object. This happens if and only if its value is the root of a tree-structured attribute or the entire range of values of a linear attribute.

Proposition 1. (i) Each nonempty existential conjunctive concept can be represented by an existential conjunctive expression that

1. contains no useless atomic formulae,
2. for every variable contains at most one atomic formula in that variable for each unary relation, and
3. for every ordered pair of distinct variables contains at most one atomic formula in those variables for each binary relation.

(ii) Moreover, each such expression represents a distinct existential conjunctive concept, modulo renaming the variables and rearranging the order of the atomic formulae.

Sketch of Proof. Condition (1) can clearly be enforced without loss of generality. To see how condition (2) can be enforced, suppose that an existential conjunctive expression contains two atomic formulae $f_1(x)$ and $f_2(x)$ that both impose restrictions on the range of values of attribute A. These can be replaced by the single atomic

formula $f_3(x)$ that restricts the value of A to the intersection of the ranges allowed in $f_1(x)$ and $f_2(x)$. Since we are assuming that the concept is nonempty, this intersection is nonempty. Hence $f_3(x)$ is well-defined, giving an interval range when A is linear and a single value (the most specific of the two values specified by $f_1(x)$ and $f_2(x)$) when A is tree-structured. Condition (3) can be enforced in similar manner. This establishes (i). Part (ii) is easily verified. □

Henceforth we will assume, unless otherwise indicated, that a concept is nonempty and an expression is in the canonical form given in Proposition 1, thereby avoiding the need to distinguish between expressions and the concepts they represent.

2.3. The generalization relation among existential conjunctive concepts

An expression in canonical form with r variables can also be represented as a complete directed graph on r nodes, similar to the way a scene is represented (see Figure 3). In this case, each node represents a variable of ϕ and the labels of nodes and edges represent restrictions imposed by the atomic formulae of ϕ. Thus to label the graph, in addition to tuples of observable values, we will allow tuples that include abstract values for tree-structured relations and ranges of the form $v_1 - v_2$, with $v_1 \leq v_2$, for linear relations. (When $v_1 = v_2$ only a single value will be used.) When no atomic formula is present for a given variable or pair of variables that involves a given relation, we will use the special symbol "?" to indicate that any value is possible. Such a graph is called a *concept graph.*

The graphical representation of existential conjunctive concepts is very useful for placing these concepts into a partial order from the most specific concepts to the most general concepts, as is used in the version space framework. This partial order is just the set containment relation: a concept ϕ_1 is (the same as or) more general than another concept ϕ_2 if $\phi_2 \subseteq \phi_1$. This relation can also be defined directly on concept graphs.

Let us first say that if l_1 and l_2 are tuples of restrictions labeling nodes or edges in two different graphs, then l_1 is *stronger* than l_2 if every component of l_1 represents a set of values that is contained in the set of values represented by the corresponding component of l_2. If G_1 and G_2 are the graphs of existential conjunctive concepts ϕ_1 and ϕ_2, respectively, then it is easily verified that ϕ_1 is more general than ϕ_2 if and only if there is a 1-1 mapping Θ from the set of nodes of G_1 into the set of nodes of G_2 such that each node in G_2 in the range of Θ is labeled with a stronger tuple of restrictions than the corresponding node in G_1 and each directed edge between two nodes in G_2 in the range of Θ is labeled with a stronger tuple of restrictions then the corresponding edge in G_1. Furthermore, we have used the "single representation trick" (Cohen, 1982), representing both scenes and concepts with the same type of graph, and thus it is easily verified that we can also check if a concept is satisfied by a given scene by checking if the concept graph is more general than the scene's instance graph. Figure 4 illustrates a mapping that shows that the scene in Figure 2 is an instance of the concept in Figure 3.

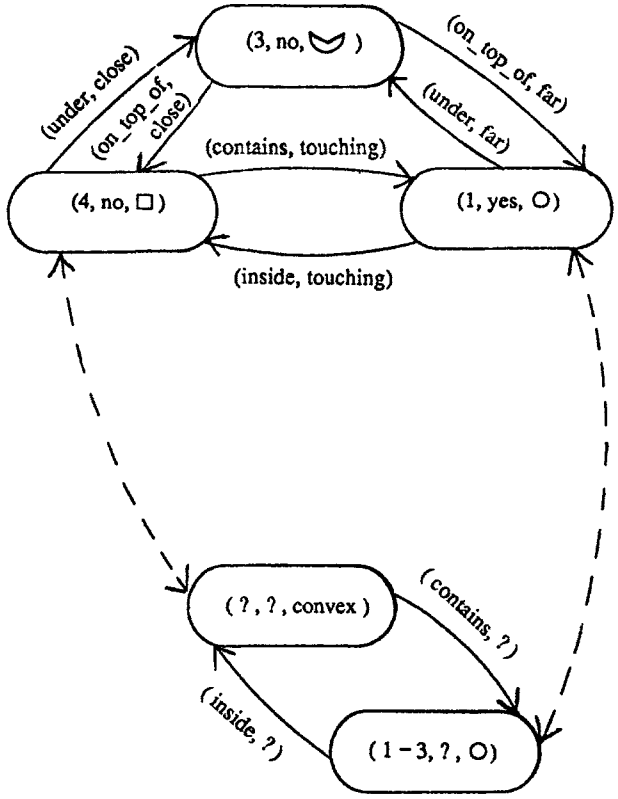


Fig. 4. The generalization relation among concept and instance graphs.

An interesting special case occurs when the set of relations R contains no unary relations and only one binary relation, which has only two observable values: *present* and *not_present*. In this case, concept graphs labeled only with observable values correspond to simple, unlabeled directed graphs (edges marked *present* are there, others are not) and the problem of whether one such concept is more general than another, or indeed, the problem of whether a scene satisfies such a concept, is exactly the problem of subgraph isomorphism for directed graphs, known to be NP-complete (Garey, 1979). Hence we cannot expect to be able to solve these problems efficiently in the general case for concepts with many variables and scenes with many objects.[2]

It is for this reason that we limit the number of objects per scene to at most k, for some reasonably small constant k. Given this restriction, we would like to design learning algorithms that are efficient even when the number n of attributes and the number l of binary relations are large. In the next section we show that this also may be difficult, even when the number k of objects per scene is restricted to two.

3. Difficulty of finding consistent existential conjunctive concepts

We assume the reader is familiar with the basics of the theory of NP-completeness as described in Garey (1979), and in particular with the fact that the following problem is NP-complete:

Definition. SAT: Given a pair (V, C) where $V = \{v_1, \ldots, v_n\}$ is a set of Boolean variables and $C = \{C_1, \ldots, C_t\}$ is a collection of clauses, each of which is a set of literals (variables or their negations) over V, determine if (V, C) is satisfiable, i.e., if there is a truth assignment for the variables $v_1, \ldots, v_n$ (each variable set to either *true* or *false*) such that each clause contains at least one literal that has value *true*.

The purpose of this section is to derive the following result.

Theorem 1. The problem of determining if there is an existential conjunctive concept consistent with a sequence of m of examples over an instance space defined by n attributes (where m and n are variable) is NP-complete, even when there are no binary relations defined, each attribute is Boolean valued, and each example contains exactly two objects.

Proof. We describe a reduction of SAT to this problem, related to the reduction given in Theorem 3.2 of Pitt (1986).

Let (V, C) be an instance of SAT with $V = \{v_1, \ldots, v_n\}$ a set of Boolean variables and $C = \{C_1, \ldots, C_t\}$ a set of clauses over V.

Define a set of $2n$ Boolean attributes $A = A_1, \ldots, A_{2n}$, each of which corresponds to one of the $2n$ literals $v_1, \ldots, v_n, \bar{v}_1, \ldots, \bar{v}_n$. Let ψ be a mapping from literals to the indices of their corresponding attributes, i.e. $\psi(v_i) = i$ and $\psi(\bar{v}_i) = n + i$, $1 \leq i \leq n$. It is assumed that ψ is also extended to map from clauses to sets of indices in the trivial manner, i.e., $\psi(S) = \{\psi(s) : s \in S\}$.

A two-object scene over the attributes $A_1, \ldots, A_{2n}$ (with no binary relations) is defined by an unordered pair (X, Y), where X is a Boolean vector of length $2n$ giving the values of the attributes for one object in the scene, and Y is the vector for the other object. To simplify our notation (following Pitt, 1986), let $\bar{1}$ denote the vector of all 1's and $\bar{0}$ the vector of all 0's, and for any $S \subseteq \{1, \ldots, 2n\}$ let $\bar{0}_S$ denote the vector that is all 1's except at the indices given in S and 1_S denote the vector that is all 0's except at the indices given in S.

Using these conventions, define a sample Q with the $n + 2$ positive examples

$$\{(\bar{1}, \bar{1}), (\bar{1}, \bar{0})\} \cup \{(\bar{0}_{\psi(v_i)}, \bar{0}_{\psi(\bar{v}_i)}) : 1 \leq i \leq n\}$$

and the t negative examples

$$\{(\bar{0}_{\psi(C_j)}, \bar{0}_{\psi(C_j)}) : 1 \leq j \leq t\}.$$

Thus (A, Q) is an instance of the above problem of finding a consistent existential conjunctive concept.

Lemma 1. The SAT instance (V, C) is satisfiable if and only if (A, Q) has a solution, i.e., if and only if there is an existential conjunctive concept consistent with all examples in Q.

Proof. Assume that (V, C) is satisfiable. Let L be the set of literals that are true in an assignment that satisfies (V, C), i.e., such that for all i, $1 \leq i \leq n$ either $v_i \in L$ or $\bar{v}_i \in L$ but not both, and for all j, $1 \leq j \leq t$, $L \cap C_j \neq \emptyset$. Let ϕ be the existential conjunctive concept defined by

$$\phi = \exists^* x : (A_{i_1}(x) = 1) \text{ and } \cdots \text{ and } (A_{i_n}(x) = 1), \text{ where } \{i_1, \ldots, i_n\} = \psi(L).$$

We will show that ϕ is consistent with the sample Q.

The concept ϕ clearly includes the positive examples $(\bar{1}, \bar{1})$ and $(\bar{1}, \bar{0})$ of Q; in each case we may take x to be the first object listed, for which all attributes have the value 1. In each of the remaining positive examples of Q there is an index i, $1 \leq i \leq n$, such that for one object, all attributes except $A_{\psi(v_i)}$ have value 1, and for the other object, all attributes except $A_{\psi(\bar{v}_i)}$ have value 1. Since L does not contain both v_i and $\bar{v}_i$, if $v_i \notin L$ then we may take x to be the former object, and if $\bar{v}_i \notin L$, we may take x to be the latter object. Hence all of the positive examples of Q are included in ϕ.

For each negative example in Q there is a clause C_j of C such that in each object, each of the attributes corresponding to the literals in C_j has value 0 and the remaining attributes have value 1. Since L contains a literal from each clause in C, L contains a literal in C_j, and hence there is an atomic formula in the conjunctive expression of ϕ that requires that the attribute corresponding to this literal have the value 1. Thus neither object satisfies the conjunctive expression of ϕ, and hence no negative example is included in ϕ. Hence ϕ is consistent with all the examples in Q.

For the other direction of the proof, assume that there exists some existential conjunctive concept ϕ that is consistent with the sample Q. First note that the conjunctive expression of ϕ cannot have any atomic formulae that sets the value of an attribute to 0, for otherwise ϕ would not include the positive example $(\bar{1}, \bar{1})$. However, this implies that ϕ cannot contain both atomic formulae for a variable x and atomic formulae for a different variable y, for otherwise we could not find an assignment of the objects in the positive example $(\bar{1}, \bar{0})$ that would satisfy ϕ (recall that the quantifier $\exists^*$ means that distinct variables x and y must be mapped to distinct objects). Hence we may assume that ϕ is of the form

$$\phi = \exists^* x : (A_{i_1}(x) = 1) \text{ and } \cdots \text{ and } (A_{i_u}(x) = 1),$$
where $\{i_1, \ldots, i_u\} \subseteq \{1, \ldots, 2n\}$.

Since ϕ also includes the remaining positive examples of Q, for any literal v_i, $1 \leq i \leq n$, ϕ cannot include both an atomic formula for the attribute corresponding to v_i and an atomic formula for the attribute corresponding to $\bar{v}_i$; otherwise, ϕ would not include the positive example $(\bar{0}_{\psi(v_i)}, \bar{0}_{\psi(\bar{v}_i)})$, in which each object has one of these attributes set to the value 0. Finally, since ϕ does not include any of the negative examples of Q, for each clause C_j, $1 \leq j \leq t$, ϕ must contain an atomic formula for an attribute corresponding to a literal that appears in C_j; otherwise, the negative example $(\bar{0}_{\psi(C_j)}, \bar{0}_{\psi(C_j)})$, in which all attributes except those corresponding to literals in C_j have the value 1 for both objects, would be included in ϕ.

It follows that the set of literals L corresponding to the attributes that appear in the conjunctive expression of ϕ can be used to define a truth assignment that satisfies (V, C): for any variable $v \in V$, if $v \in L$ then set v to true, else set v to false. Hence (V, C) is satisfiable. □

Since the problem of finding a consistent existential conjunctive hypothesis can be solved in nondeterministic polynomial time for a fixed number k of objects per scene, and the above reduction can be accomplished in polynomial time, the theorem follows. □

One sidelight of the above proof is that it actually shows that the problem in question is NP-complete even if, in addition to the restrictions listed in the statement of the theorem, we restrict ourselves to existential conjunctive concepts whose canonical expressions have only one variable. This may appear contradictory at first, since such expressions are essentially equivalent to variable-free pure conjunctive expressions, e.g., as studied in (Haussler, 1988), for which there are many known learning algorithms. However, these algorithms work only in the attribute-based domain, where there is only one object in each example and hence no ambiguity regarding the mapping of attributes in the example to attributes in the hypothesis. The above result shows that as soon as we introduce even the minimal amount of ambiguity, i.e., by having two objects in each example instead of just one, then the problem of finding a consistent hypothesis becomes substantially more difficult.

Another interesting sidelight of the above proof is that it indicates how to construct samples in which the size of the sets S and G of Mitchell's version space algorithm are exponential. This is to be expected, since otherwise Mitchell's algorithm could be used to solve the version space nonemptiness problem for existential conjunctive concepts in polynomial time, implying that P = NP. An explicit construction can be given as follows. (The remainder of this section may be skipped without loss of continuity.)

Let (V, C) be an instance of SAT with $V = \{v_1, \ldots, v_n\}$ a set of Boolean variables and $C = \{C_1, \ldots, C_n\}$ a set of clauses over V, where $C_j = \{v_j, \bar{v}_j\}$ for all j, $1 \leq j \leq n$. Clearly this is a trival instance of SAT, in the sense that every truth assignment of the variables in V satisfies this instance.

Using the reduction of SAT to the problem of finding a consistent existential conjunctive concept given in Lemma 1 above, we can construct a set of $2n$ Boolean attributes $A = A_1, \ldots, A_{2n}$ and a sequence Q of $m = 2n + 2$ two-object examples

that have the following property: If L is the set of literals that are true in a truth assignment that satisfies (V, C) then the existential conjunctive concept

$$\exists^* x : (A_{i_1}(x) = 1) \text{ and } \cdots \text{ and } (A_{i_n}(x) = 1), \text{ where } \{i_1, \ldots, i_n\} = \psi(L), \quad [1]$$

is consistent with Q, and conversely, any existential conjunctive concept that is consistent with Q has the form (1) for some L that constitutes the set of literals that are true in some truth assignment satisfying (V, C). This can be verified by the arguments given in the proof of Lemma 1, and by the additional observation that for the particular choice of C above, in order to contain a literal in each clause of C, L must include either v or $\bar{v}$ for each variable $v \in V$.

It follows that there is a 1-1 correspondence between the hypotheses in the version space of Q (with respect to existential conjunctive concepts) and the satisfying assignments of (V, C). This implies that the version space has size 2^n, since this is the number of distinct satisfying assignments of (V, C). Furthermore, any two distinct concepts in this version space, i.e., any two expressions of the form given in (1) that are not identical up to a rearrangement of atomic formulae, are clearly incomparable with respect to the partial order of increasing generality. Hence the set S of maximally specific concepts in the version space equals the set G of maximally general concepts in the version space, which in turn equals the entire version space. Hence both S and G have sizes exponential in n, and hence in m as well.

4. Learning from random examples

Theorem 1 indicates that the computation time in the worst case is more than polynomial in the sample size for any learning algorithm that learns existential conjunctive concepts by drawing a set of examples and then producing an existential conjunctive hypothesis consistent with these examples, unless P = NP. There are several ways one might try to get around this negative result. We will look at two of them in Sections 3.2 and 3.3 below. Both involve making the assumption that the examples of the target concept are generated by choosing instance graphs independently at random (with replacement) from the instance space according to some fixed probability distribution on this space, and labeling each instance as positive or negative according to whether or not it is an instance of the target concept. Such examples will be called *random examples.*

We begin by discussing a specific framework for studying learning from random examples.

4.1. Valiant's PAC learning framework

Assume that given a sequence of random examples of some unknown existential conjunctive target concept, a learning algorithm produces a hypothesis that is itself an existential conjunctive concept, specified by a concept graph or an existential conjunctive expression. This hypothesis need not be consistent with all the examples. Rather, we define a successful learning algorithm as one that, with high probability, produces a hypothesis that will perform well on further random examples drawn according to the same fixed distribution. More formally, we define the *error* of a hypothesis as the probability that it disagrees with the target concept on a randomly drawn instance. A successful learning algorithm is then one that, from random examples of any existential conjunctive target concept, with high probability, finds a hypothesis with small error. (The assumption that the target concept is existential conjunctive is not actually needed in some of the results below (Propositions 3 and 4)).

If we now add the requirement that

(1) the learning algorithm uses only polynomial sample size and computation time and
(2) it produces a hypothesis with small error with high probability for any probability distribution on the instance space,

then we get (essentially) the learning framework defined by Valiant (1984). This has been called *Probably Approximately Correct (PAC) learning* (Angluin, 1988). Here the polynomial in condition (1) is a function of the complexity of the learning task (i.e., the number of attributes and relations, and the syntactic size of the target concept in canonical form) and the inverses of the *accuracy* and *confidence* parameters, usually denoted ϵ and δ, respectively. Learning with accuracy $1 - \epsilon$ and confidence $1 - \delta$ means getting a hypothesis with error at most ϵ with probability at least $1 - \delta$.

We are already fixing the number of objects per scene to at most k for some constant k to avoid NP-completeness problems in determining concept membership. Thus we would not demand that the polynomial in (1) depend on k as well. This means that we would look for algorithms that scale well with the number of relations, but not necessarily with the number of objects per scene. A more detailed discussion of the PAC framework may be found in Haussler (1988) and Haussler, et al. (1988).

Since it is conceivable that one could find a hypothesis with small error without fitting the training examples exactly, at first glance there appears to be some hope that this PAC framework, by itself, could avoid the problems indicated by Theorem 1. However, results in Pitt (1986) show that finding hypotheses with small error for an arbitrary probability distribution on the instances is at least as hard as finding

a consistent hypothesis. So this hope is unfounded. This observation is formalized as follows.

Let RP denote the class of all problems (expressed as formal languages) that have randomized polynomial time algorithms. These are deterministic algorithms that are allowed to flip a fair coin to decide their next move (Gill, 1977). (This class is actually called VPP in Gill (1977).) It is easy to show that RP $\subseteq$ NP. Furthermore, just as it is strongly suspected that P $\neq$ NP, so it is strongly suspected that RP $\neq$ NP. Using the results from (Pitt, 1986) (see Natarajan, 1987, Blumer, et al., 1988), Theorem 1 of the previous section implies.

Proposition 2. Existential conjunctive concepts are not learnable in the PAC framework described above unless RP = NP. Moreover, this holds even when no binary relations are defined, each attribute is Boolean-valued, and each example contains exactly two objects. □

Hence it appears that we cannot avoid the computational complexity problems indicated by Theorem 1 by adopting the PAC model. In fact, adopting this model seems only to make matters worse, since even if we did find a consistent (or nearly consistent) hypothesis, we have no guarantee that it will have small error. We deal with this problem in the next section. For now we note only that Proposition 2 does not follow from Theorem 1 in a variant of the PAC framework in which the hypothesis space is allowed to be different from the target class (see e.g., Kearns, et al., 1987, Haussler, et al., 1988). Thus there may be efficient PAC algorithms for existential conjunctive concepts that use different hypothesis spaces. Some very weak results of this type (from a practical point of view) are given in Haussler (1987). We will not pursue this further here.

4.2. Sample size required to learn existential conjunctive concepts

Many heuristic techniques have been developed for finding existential conjunctive concepts consistent with a set of examples (Vere, 1975; Hayes-Roth, 1978; Michalski, 1983; Kodratoff, 1986). Some of these techniques appear to work well in applications of practical interest even though Theorem 1 indicates that it is very likely that examples could be constructed that they will flounder on. We may try to get around Theorem 1 by postulating such a heuristic. However, suppose that we do find a heuristic technique that, in practice, often produces a hypothesis consistent with the sample in a reasonable time. What guarantee do we have that this hypothesis will have small error with respect to the target concept? We will show that learning any algorithm for existential conjunctive concepts, when it succeeds in finding a hypothesis consistent with a large enough random sample, inevitably produces a hypothesis that with high probability has small error. In fact, this also holds when hypotheses are not completely consistent with all training examples, but merely disagree with but a small fraction of training examples. Hence

each of the above referenced learning algorithms for existential conjunctive concepts can be an effective method, if not always an efficient method.

Let us first return to Mitchell's version space framework, as discussed in the introduction. We define for any hypothesis space H and any sequence of examples Q, the *version space of Q with respect to H* as the set of all hypotheses in H that are consistent with all examples in Q. Assuming a fixed probability distribution on the instance space, as in (Haussler, 1988) we can define a situation in which enough critical examples have been drawn so that no hypothesis in the version space has large error.

Definition. Given a hypothesis space H, a target concept ϕ, a sequence of examples Q of ϕ, and an *error tolerance* ϵ, where $0 \leq \epsilon \leq 1$, the version space of Q (w.r.t. H) is *ϵ-exhausted* (w.r.t. ϕ) if it does not contain any hypothesis that has error more than ϵ with respect to ϕ.

If the version space (w.r.t. H) is ϵ-exhausted for small ϵ, then any hypothesis in this version space is a good approximation to the target concept in the sense that its error is small relative to the target, hence any learning algorithm that finds a hypothesis in H consistent with the sample has found a good hypothesis. Furthermore, as the sample gets larger, fewer hypotheses in H will be consistent with it, so the version space will shrink, increasing the probability that it becomes ϵ-exhausted. This provides a way to force a learning algorithm that searches a hypothesis space H for a consistent hypothesis to produce either a good hypothesis or no hypothesis at all: we choose some small ϵ and only run it on random samples Q that are large enough so that with high probability the version space of Q (w.r.t. H) is ϵ-exhausted with respect to any possible target concept in H. This way, if it produces a consistent hypothesis, then with high probability this hypothesis has error less than ϵ, simply because all hypotheses with error greater than ϵ have been eliminated from the version space. The question that remains is how large Q must be.

In Haussler (1988) the notion of bias (Mitchell, 1980) inherent in a restricted hypothesis space is quantified in a way that relates the bias of a hypothesis space to the number of samples required to ϵ-exhaust a version space within it with high probability. To do this, bias is quantified by the *growth function* (or *capacity*) (Vapnik, 1982).

Definition. Let X be an instance space and let H be a hypothesis space defined on X. For any finite set $S \subseteq X$ of instances, $\Pi_H(S) = \{S \cap h : h \in H\}$, i.e., the set of all subsets of S that can be obtained by intersecting S with a concept in H. Equivalently, one can think of $\Pi_H(S)$ as the set of all ways the instances of S can be divided into positive and negative instances so as to be consistent with some hypothesis in H, i.e., the set of all dichotomies of S induced by hypotheses in H. For every $m \geq 1$, $\Pi_H(m)$ denotes the maximum of $|\Pi_H(S)|$ over all S of size m. Thus $\Pi_H(m)$ is the maximal number of dichotomies induced on any m instances by

the hypotheses in H. We will refer to $\Pi_H(m)$ as the *growth function* or *capacity* of H. □

The capacities of a variety of attribute-based concept classes are calculated in (Haussler, 1988) (see also Pearl, 1978). The following result from (Blumer, Ehrenfeucht, Haussler, and Warmuth et al., 1988) (essentially due to Vapnik and Chervonenkis (Vapnik, 1982) and given in other forms in (Blumer, et al., 1988) (Haussler, 1986)) relates the capacity of a hypothesis space to the rate at which version spaces within it become exhausted.

Proposition 3. Let H be a hypothesis space[4] and $0 < \epsilon < 1$. If Q is a sequence of m independent random examples (chosen according to any fixed probability distribution on the instance space) of any target concept ϕ, then the probability that the version space of Q (w.r.t. H) is not ϵ-exhausted (w.r.t. ϕ) is less than

$$2\Pi_H(2m)2^{-\epsilon m/2}. \square$$

This result can also be extended to take into account a slightly larger version space, one that includes all hypotheses that disagree with the sample Q on at most a small fraction of the examples in Q (see Mitchell, 1982). This can be useful when the target concept is not itself existential conjunctive, and more importantly, when there is a stochastic element in the labeling of the training examples, as well as in the selection of training instances. This occurs, for example, when there is noise either in the classification labels themselves, or in the measurement of the values of the attributes.

In order to model this more general setting, let us now extend the fixed probability distribution used above from a distribution on the instance space to a distribution on the space of all possible examples. Formally, we define a distribution D on the space $X \times \{+, -\}$, where X is the instance space (e.g., all possible instance graphs with up to k nodes on a given set of relations), and $\{+, -\}$ is the set of possible classification labelings of examples. We use two classification values here only as a matter of convenience, in keeping with our focus on simple concept learning. Random examples are now generated by drawing (independently with replacement) directly from the distribution D. Each draw provides a whole example, not just an instance, so we no longer need the notion of a target concept. The distribution D itself acts as a kind of stochastic target concept. This is in fact the typical model used in the pattern recognition literature (Vapnik, 1982; Duda, 1973) (see also the appendix of Blumer, et al. (1988) for further discussion).

Using this model, the *error* of a hypothesis h with respect to the distribution D is defined as before: it is the probability that the hypothesis disagrees with an example drawn at random from D. In that case that the distribution D assigns probability 0 to all but one classification for any given instance, this reduces to the notion of the error of h relative to a certain target concept, defined above. Using the more general results of Vapnik and Chervonenkis for arbitrary distributions D

on $X \times \{+, -\}$ (see Theorem A3.3, Blumer, et al., 1988), Proposition 3 can now be generalized to

Proposition 4. Let X be an instance space, H be a hypothesis space on X, and $0 < \epsilon < 1$. If Q is a sequence of m independent random examples chosen according to any fixed probability distribution on $X \times \{+, -\}$, then the probability that there is any hypothesis in H that disagrees with a fraction less than $\epsilon/2$ of the examples in Q, yet has error more than ϵ with respect to D, is less than

$$8\Pi_H(2m)e^{-\epsilon m/16}. \square$$

In essence, this result says that when the capacity of H does not grow too quickly as a function of the sample size m, then for large enough sample size, for any hypothesis h in H, the observed rate of error of h on the training sample is not much less than its true rate of error, i.e., the rate of error that would be observed on a large enough independent test sample. Thus a learning algorithm cannot be fooled into proposing a bad hypothesis just because it looks good on the training data. The choice of demanding observed rate of error less than $\epsilon/2$ is somewhat arbitrary; any fixed fraction of ϵ would do. Only the constants in the probability bound above would change (see Blumer, et al., 1988).

To complete our analysis, we now calculate an upper bound on the capacity of the hypothesis space used in learning existential conjunctive concepts.

As in previous sections we will assume that scenes contain at most k objects, where $k \geq 2$, and there is a fixed set of n attributes and l binary relations, each tree-structured or linear, that characterize objects and relations between objects in a scene. To simplify notation, we will assume that $n = l$ and use n to denote both the number of attributes and the number of binary relations. This set of scenes defines the instance space X. Given a sequence of examples from this instance space, our heuristic learning algorithm will search some restricted hypothesis space for a hypothesis that is consistent with these examples. For the present purposes, this will always be one of the hypothesis spaces H_s, $0 \leq s \leq nk^2$, consisting of all existential conjunctive concepts that can be represented by an existential conjunctive expression with at most k variables and syntactic size at most s (i.e., using at most s atomic formulae). The restriction that $s \leq nk^2$ comes from that fact that using the canonical form of Section 1, no expression needs more than one atomic formula for every combination of attribute and variable and one atomic formula for every combination of binary relation and ordered pair of distinct variables ($k^2 = k + k(k - 1)$). Since all concepts that are represented by expressions that use at most k variables are included in H_{nk^2} and since, in view of our limitation of at most k objects per scene, any expression with more than k variables represents the empty concept, the hypothesis space H_{nk^2} is actually the class of all existential conjunctive concepts over the instance space X.

Lemma 2. For all $m, s \geq 1$, $\Pi_{H_s}(m) \leq k\ (k^6nm^2)^s/s!$, where k is the maximum number of objects per scene and n is the maximum number of attributes defined on objects and relations defined between pairs of objects.

Proof. Let S be a set of m scenes chosen from the instance space X. Let $OBJ(S)$ be a list of all objects from scenes in S and $PAIR(S)$ be a list of all ordered pairs of distinct objects from scenes in S, where both objects in each pair come from the same scene. Since each scene contains at most k objects, $OBJ(S)$ includes at most km objects and $PAIR(S)$ includes at most $k\ (k-1)m$ pairs. To establish our result, we calculate an upper bound on the size of $\Pi_{H_s}(S) = \{Z \subseteq S : Z = S \cap h$ for some $h \in H_s\}$. Our strategy is to determine some conditions that imply that two distinct hypotheses h_1 and $h_2 \in H_s$ induce the same subset Z, in the sense that $S \cap h_1 = Z = S \cap h_2$. From this we will get an upper bound on the number of distinct Z's induced by hypotheses in H_s.

We need the following notation. For any existential conjunctive expression

$$\phi = \exists^* x_1, \ldots, x_r : f_1 \text{ and } f_2 \text{ and } \cdots \text{ and } f_s,$$

and variable x_i of ϕ, let $RES_{x_i}(\phi)$ be the pure conjunctive concept formed by the conjunction of all unary atomic formulae of ϕ involving x_i. For example, if ϕ is

$$\exists^* x, y : (shape(x) = polygon) \text{ and } (rel\_pos(x, y) = ontopof)$$

$$\text{and } (shape(y) = square) \text{ and } (color(x) = red)$$

then $RES_x(\phi)$ is

$$(shape = polygon) \text{ and } (color = red).$$

Similarly, for any ordered pair (x_i, x_j) of distinct variables in ϕ, let $RES_{(x_i, x_j)}(\phi)$ be the pure conjunctive concept formed by the conjunction of all binary atomic formulae of ϕ involving the pair (x_i, x_j).

We claim that $S \cap h_1 = S \cap h_2$ whenever

(1) The number of variables in h_1 and h_2 are the same and can be put in 1-1 correspondence such that
(2) for each variable x of h_1 and corresponding variable $\bar{x}$ of h_2, the set of all objects in $OBJ(S)$ that satisfy $RES_x(h_1)$ is the same as the set that satisfy $RES(h_2)$ and
(3) for each ordered pair of distinct variables (x, y) of h_1 and corresponding pair $(\bar{x}, \bar{y})$ of h_2, the set of all ordered pairs of objects in $PAIR(S)$ that satisfy $RES_{(x, y)}(h_1)$ is the same as the 04set that satisfy $RES_{(\bar{x}, \bar{y})}(h_2)$.

To verify this, simply observe that under assumptions (1), (2), and (3), if a scene in S satisfies h_1 according to the definition given in Section 1 then it will also satisfy h_2 and vice versa.

To obtain an upper bound on the cardinality of $\Pi_{H_s}(S)$ we will now simply count (or upper bound) the number of ways we can

(A) choose the number of variables r in a canonical expression, possibly with some useless atomic formulae, for an $h \in H_s$ and for a given r, assuming the variables in h are $x_1, \ldots, x_r$,
(B) choose the subsets of $OBJ(S)$ consisting of all objects that satisfy $RES_{x_i}(h)$ for each i, where $1 \leq i \leq r$ and
(C) choose the subsets of $PAIR(S)$ consisting of all ordered pairs of objects that satisfy $RES_{(x_i, x_j)}(h)$ for each i, j, where $1 \leq i < j \leq r$.

As far as (A) is concerned, we can assume that the number of variables is between 1 and k. This is because any existential conjunctive expression with more than k variables is not satisfied by any scene in S, and hence these hypotheses add at most 1 to the cardinality of $\Pi_{H_s}(S)$.

As for (B) and (C), given r, where $1 \leq r \leq k$, and $h \in H_s$ with variables $x_1, \ldots, x_r$, we can assume that h has exactly s atomic formulae, since if h has less than s atomic formulae then we can add useless atomic formulae without changing the set of scenes that satisfy h. Since h has a total of $r + r(r-1) = r^2$ variables and pairs of distinct variables, there are a total of r^{2s} ways that these s atomic formulae can be assigned to variables and pairs of distinct variables. In addition, each atomic formula can be defined using any one of n relations, so these relations can be assigned to the s atomic formulae in n^s ways. Furthermore, there are many ways to assign value restrictions to the atomic formulae (i.e., pairs of values v_1 and v_2 with $v_1 \leq v_2$ for linear relations and observable or abstract values for tree-structured relations), but all that really matters is what subset of the objects in $OBJ(S)$ (or the pairs in $PAIR(S)$ if the atomic formula is binary) have values that comply with these restrictions. It is easy to see that the maximal number of such subsets occurs when the atomic formula is binary, $PAIR(S)$ is as large as possible (i.e., has size $k(k-1)m$), each pair of objects in $PAIR(S)$ has a distinct value for the relation of the atomic formula, and this relation is linear. In this case we can imagine that all $k(k-1)m$ pairs are sorted in increasing order according to their values. A restriction that the value lie between two given values amounts to selecting an interval of pairs from this ordering, which can be done in $1 + k(k-1)m + \binom{k(k-1)m}{2} \leq (k(k-1)m)^2$ ways, giving an upper bound of $(k(k-1)m)^{2s}$ on the distinct (w.r.t. S) ways that the atomic formulae in h can be assigned value restrictions. Finally, the order of the atomic formulae is immaterial, so this gives an upper bound of $\frac{r^{2s} n^s (k(k-1)m)^{2s}}{s!}$ on the number of ways we can specify (B) and (C) for a given choice r for (A).

It follows that the cardinality of $\Pi_{H_s}(S)$ is at most

$$1 + \sum_{r=1}^{k} \frac{(r^2 n\,(k\,(k-1)m)^2)^s}{s!},$$

which is certainly less than $k\,(k^6 n m^2)^s/s!$. □

From Proposition 4 and Lemma 2 we get the following

Theorem 2. Let the number of objects per scene in the instance space X be at most k and both the number of attributes defined on objects and the number of relations defined between pairs of objects in scenes be at most n. Let D be an arbitrary distribution on $X \times \{+, -\}$. Then for any $1 \le s \le nk^2$, there is a sample size m that is

$$O\left(\frac{s}{\epsilon} \log \frac{kn}{\epsilon}\right)$$

that is sufficient for learning existential conjunctive concepts over X in the following sense: Given m independent random examples from D, any algorithm that succeeds in finding an existential conjunctive hypothesis with s atomic formulae that disagrees with at most $\epsilon m/2$ of these examples has, with probability at least $1 - O\,(ke^{-\epsilon m})$, found a hypothesis with error less than ϵ.

Proof. We show that this holds for

$$m = \frac{128s}{\epsilon} \ln \frac{512ek^6 n}{\epsilon},$$

where ln denotes the natural logarithm and e its base. We claim that using Lemma 2, with $H = H_s$ and this value of m, the bound given in Proposition 4 is at most $\sqrt{\frac{32}{\pi s}}\,ke^{-\epsilon m/32}$. The result follows.

To verify the claim, first note that using Stirling's approximation, $\frac{k\,(k^6 n m^2)^s}{s!} \le \frac{k}{\sqrt{2\pi s}} \left(\frac{ek^6 n m^2}{s}\right)^s$. Let $\alpha = \frac{8k}{\sqrt{2\pi s}} = \sqrt{\frac{32}{\pi s}}\,k$ and $\beta = 4ek^6 n$. Then by Lemma 2, $8\Pi_{H_s}(2m) \le \alpha(\beta m^2/s)^s$. Hence if $(\beta m^2/s)^s \le e^{\epsilon m/32}$, then the bound in Proposition 4 is at most $\alpha e^{-\epsilon m/32}$, as claimed. It is easily verified that the former inequality holds for

$$m = \frac{128s}{\epsilon} \ln \frac{128\beta}{\epsilon} = \frac{128s}{\epsilon} \ln \frac{512ek^6 n}{\epsilon},$$

using the fact that $s \le nk^2 < \beta$. □

Theorem 2 shows the sample size required grows only logarithmically as the number of objects per scene and the number of relations between attributes is

increased. Note also that the bound given is independent of the number of values that each attribute and relation can take on. See (Haussler, 1988) for further discussion of this point. Finally, note that sample size bounds that hold for any algorithm that produces a nearly consistent hypothesis, regardless of the number of atomic formulae it contains, can be obtained by setting $s = k^2n$, since this is the maximum number of atomic formulae needed in any existential conjunctive expression by Proposition 1.

The constant factors in the proof of Theorem 2 are still quite large. It is likely that with more work, including improvements to Proposition 4, smaller constants can be obtained. In the case that the hypothesis produced is completely consistent with the training examples, a sample size of

$$m = \frac{16s}{\epsilon} \log \frac{64eh^6n}{\epsilon},$$

can be used, where log denotes the logarithm base 2. The analysis is similar, but uses Proposition 3 in place of Proposition 4. However, there is a limit to how much this theorem can be improved. From Corollary 5.7 of Haussler (1988) using results from Ehrenfeucht, et al. (1988) it can also be shown that for $k = 1$ (i.e., for attribute-based instance spaces)

$$\Omega\left(\frac{\bar{s}}{\epsilon} \log \frac{n}{\bar{s}}\right)$$

is a lower bound on the sample size required by any learning algorithm for conjunctive concepts when examples are drawn according to the worst case distribution on the instance space and labeled correctly according to the worst case conjunctive target concept that has at most $\bar{s}$ atomic formulae. Clearly this lower bound also holds for larger k, and thus this shows that the bound in Theorem 2 cannot be improved by more than a logarithmic factor.

4.3. Finding good hypotheses using subset queries

The results above indicate that the only obstacle to learning existential conjunctive concepts from random examples is the computational complexity of finding a consistent or nearly consistent hypothesis. This leads us to consider learning algorithms that use other information to help them search the hypothesis space. One possibility is to allow the learning algorithm to formulate queries during learning, as in (Sammut, 1986; Subramanian, 1986; Muggleton, 1988; Valiant, 1984; Angluin, 1988).

While many types of queries have been used, we will consider only *subset queries*. In a subset query, the learning algorithm formulates a hypothesis h, here expressed as an existential conjunctive expression or concept graph, and then asks an oracle

if h is contained in (i.e., equal to or more specific than) the target concept. The oracle, representing a teacher or expert that knows the target concept, responds yes or no. We do not assume that a counterexample is returned when the answer is no, as in Angluin (1988).

We will first describe an algorithm that, for a fixed bound k on the number of objects per scene, finds a hypothesis consistent with any set of examples of an existential conjunctive target concept using a number of subset queries at most linear in the number of examples. Then by applying a greedy simplification step as in (Haussler, 1988), this hypothesis will be reduced to one that has a number of atomic formulae within a logarithmic factor of the minimum possible for any consistent hypothesis. The latter hypothesis will still be consistent with most of the examples. Using Proposition 4 above, we will show that this algorithm is a PAC learning algorithm for existential conjunctive concepts, modulo the added ability to make subset queries, and that it uses a sample size that is within a poly-logarithmic factor of optimal for any PAC learning algorithm.

The main technique used in the algorithm is that of matching positive examples with each other and with intermediate hypotheses to form maximally specific common generalizations. This is the technique used in constructing the set S in Mitchell's (1982) version space algorithm, and in many other learning algorithms for existential conjunctive concepts that have been investigated (see Dietterich, 1983). In presenting the algorithm, we will, for simplicity, assume that all examples have exactly k objects, for some $k \geq 2$.

The technique of forming maximally specific common generalizations (MSG's) can be illustrated by considering the set of examples and target concept given in Figure 5. Here the attributes and relations are those given in Figures 1a and 1b. Considering only the first two positive examples in Figure 5, two MSG's can be obtained, each from one of the two possible 1-1 matchings between the objects in the first example and the objects in the second. These MSG's are given in Figure 6.

The first is obtained by matching the triangle of the first example with the triangle of the second, and similarly for the circles. Since both circles are unshaded, we specify that the circle must be unshaded. Since one has size 1 and the other size 4, we specify a size range of 1 to 4 (Michalski's "closing interval rule" (Dietterich, 1983)). The attributes for the triangle are calculated in a like manner, except that the size range of 1 to 5 leads to a useless atomic formula, which is denoted by the value "?". To calculate the relations between the objects, note that in the first example the triangle is to the left of the circle and in the second it is to the right. Therefore, the maximally specific value for the **rel_pos** relation between the objects is *side-by-side* (Michalski's "climbing generalization tree rule"). Finally, in both examples the objects are close. The second MSG is computed in a similar manner, matching the triangle of the first example with the circle of the second.

In the same manner we can obtain the MSG's of any pair of existential conjunctive concepts, the MSG's of a pair of instances being a special case (using the single representation trick). The maximally specific generalization of two concept graphs

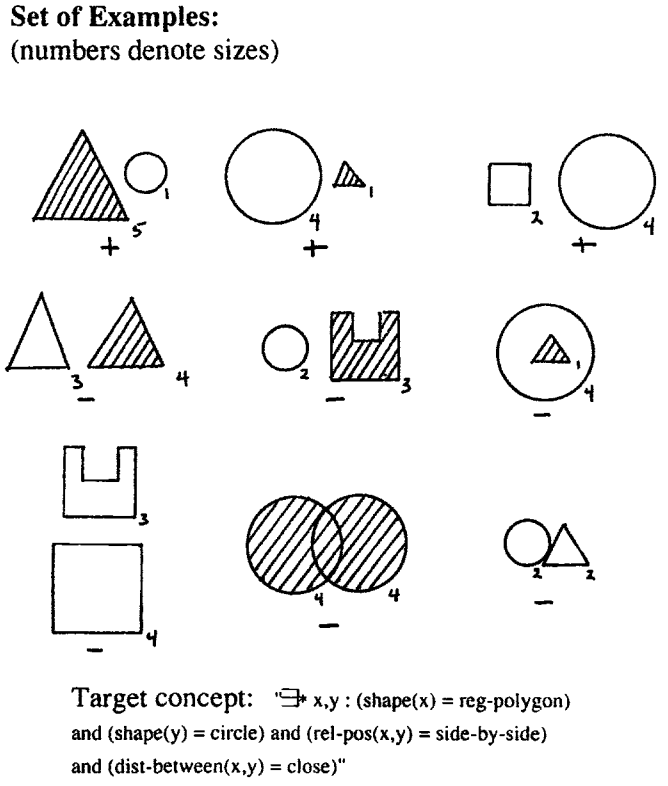


Fig. 5. Positive and negative examples of a target concept.

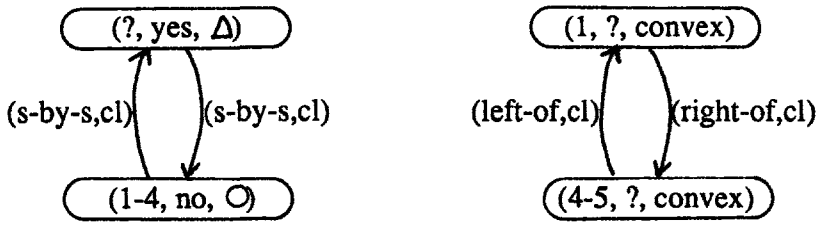


Fig. 6. The two MSG's for the first two positive examples.

under a given 1-1 matching of their nodes is defined by taking the maximally specific restrictions that include the value ranges specified in labels of the matched graphs as the value range of each attribute and relation in the resulting graph. Figure 7 shows the two MSG's obtained from the first generalization in Figure 6 and the third positive example. Each corresponds to one of the 1-1 matchings of the nodes in the generalization with the objects in the example. Note, however, that the first

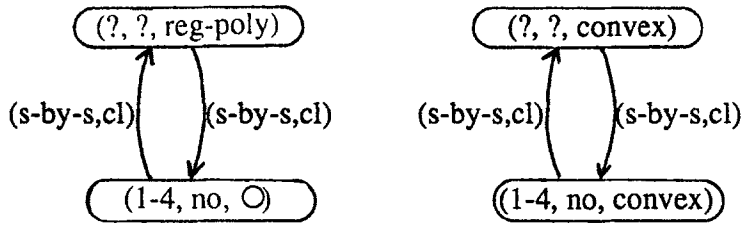


Fig. 7. MSG's obtained from the first generalization in figure 6 and the third positive example.

MSG is a specialization of the second MSG. This shows that not every matching leads to an MSG that is maximally specific among all possible MSG's of the two graphs (i.e., not every matching produces a generalization in Mitchell's S set).

The notion of the MSG's of two concept graphs extends directly to the case of more than two graphs. Given m graphs with 2 nodes each (resp. k nodes each), there are 2^{m-1} (resp. $(k!)^{m-1}$) different 1-1 matchings among the nodes of these graphs, each potentially creating a distinct MSG. The set of positive examples used to create the exponential size set S in the construction at the end of Section 2 comes close to achieving this worst case for $k = 2$. However, to find a hypothesis consistent with a set of examples, it suffices to find just one of these MSG's of the positive examples that does not include any negative examples. Instead of doing this by heuristic search, a process that is very likely exponential time in the worst case by Theorem 1, we propose to do it by making subset queries. We use the following simple lemma.

Lemma 3. Let G, G_1, G_2 be concept graphs for existential conjunctive concepts ϕ, ϕ_1, ϕ_2, respectively, where G_1 and G_2 each have k nodes. If $\phi_1 \subseteq \phi$ and $\phi_2 \subseteq \phi$ then there exists a 1-1 matching between the nodes of G_1 and the nodes of G_2 such that the MSG of G_1 and G_2 under this matching is contained in ϕ.

Proof. Since $\phi_1 \subseteq \phi$, there is a 1-1 mapping Θ_1 from the set of nodes of G into the set of nodes of G_1 such that each node in G_1 in the range of Θ_1 is labeled with a stronger tuple of restrictions than the corresponding node in G and each directed edge between two nodes in G_1 in the range of Θ_1 is labeled with a stronger tuple of restrictions then the corresponding edge in G. Since $\phi_2 \subseteq \phi$, there is a similar mapping Θ_2 from the nodes of G into the nodes of G_2. The desired matching between the nodes of G_1 and the nodes of G_2 can be defined by taking Θ_2 composed with the inverse of Θ_1 and extending it to a full matching by defining an arbitrary 1-1 matching among the nodes of G_1 and G_2 that are not in the range of Θ_1 or Θ_2, respectively. □

Theorem 3. For any fixed bound k on the number of objects per scene, existential conjunctive concepts are PAC learnable using subset queries.

Proof. Again we assume that each example contains a scene with exactly k objects; the proof of the general case with less than or equal k objects per scene is similar. We use the following simple learning algorithm:

Suppose that S is a set of examples of some existential conjunctive target concept ϕ. If S contains no positive examples, return the empty hypothesis. Otherwise, choose any positive example in S and initialize the hypothesis h to be the most specific existential conjunctive concept that contains this example. We can do this by simply interpreting the instance graph for this example as a concept graph. Now, while there are still positive examples left in S that are not contained in h, do

1. choose any such positive example,
2. for each 1-1 matching of the nodes in this example with the nodes in h, form the MSG h' and make a subset query to determine if $h' \subseteq \phi$, until a matching is found for which the answer is yes,
3. replace h by h'.

When all positive examples are contained in h, return h.

To illustrate this algorithm, assume the sample S and target ϕ are as given in Figure 5, and that the first two positive examples in that figure are used to form the first MSG h' in step (2), using the matching that leads to the first MSG in Figure 6. It is easily verified that the subset query returns yes. Hence this MSG becomes the new h. Since the third positive example is not contained in h, another MSG is formed with this example; perhaps the first MSG given in Figure 7. Again the subset query returns "yes" and h is updated. All positive examples are now included, so this hypothesis is returned.

To see that this algorithm always finds a consistent hypothesis, note that its initial hypothesis is always contained in the target ϕ, and that this property is preserved in step (3), since the answer to the previous query in step (2) was yes. Thus by induction, Lemma 3 shows that step (2) will always terminate correctly, giving a matching for which the answer to the query is yes. Since the hypothesis is generalized in each execution of step (3), it must eventually contain all positive examples. Since it is always contained in the target concept, it will never contain any negative examples. Hence the algorithm is correct.

It is clear that for fixed k this algorithm is linear in the number m of training examples and the number n of relations, assuming each call to the oracle is charged unit cost. Furthermore, by Theorem 2 with $s = k^2n$, this algorithm is a PAC learning algorithm for existential conjunctive concepts using sample size

$$O\left(\frac{k^2n}{\epsilon}\log\frac{kn}{\epsilon}\right)$$

plus some small logarithmic term involving the confidence δ (see e.g., Haussler, 1988). □

The worst case computation time for the algorithm given above is proportional to $k^2k!nm$, where n is the number of relations (in the worst case all binary). Note that this is the same as the time needed to find which of m instances are contained in a given concept by straightforward matching. The worst case number of queries used is approximately $k!m$. These bounds can probably be improved in practice by choosing examples and matchings judiciously. We do not pursue this further here.

We have also assumed that the examples are generated without noise and that the queries are always answered correctly. We can get around the former assumption to a certain extent by using queries to find an initial hypothesis that is contained in the target concept and then simply rejecting any positive example that is not included in the current hypothesis h and has no MSG with the current hypothesis that is contained in the target concept. When the fraction of such examples is small, Theorem 2 still applies. Here we rely on correct answers to the queries. We know of no way of avoiding this latter assumption.

The algorithm we have given does not make use of negative examples at all. This is similar to the classical algorithm for conjunctive concepts in attribute-based domains, which forms the unique MSG of the positive examples directly (see, e.g., the discussion in Haussler, 1988). However, as in Haussler (1988), a variant of the above algorithm that does use the negative examples can get by with fewer examples by restricting its hypothesis space to existential conjunctive concepts that are not much larger than the target concept. We briefly sketch this idea.

First note that since the algorithm of Theorem 3 always forms a maximally specific hypothesis, for smaller sample sizes, this hypothesis is likely to contain more atomic formulae than necessary to eliminate all the negative examples. For example, in the illustration of the algorithm given above, the final hypothesis contains superfluous atomic formulae indicating that the circle must be unshaded and have size between 1 and 4, among others. This means that the hypothesis h is more complex than it needs to be, in terms of the number s of atomic formulae it has. We call s the *size* of h.

Given an existential conjunctive hypothesis h that is consistent with a set of positive and negative examples, it is in general NP hard to find the smallest hypothesis which is consistent with the examples that can be obtained from h by deleting atomic formulae. This follows from the analogous result for attribute-based domains given in Haussler (1988). However, there is a greedy method that produces a simplification h' of h that is consistent and reasonably small. We describe this method now.

First note that any simplification h' of h obtained by deleting atomic formulae, since it generalizes h, will still be consistent with all the positive examples. In order to be consistent with a single negative example, all possible 1-1 matchings from the nodes of h' into the nodes of the example must conflict with at least one atomic formula in h'. When such a conflict occurs, we will say that the atomic formula *eliminates* the given matching. Only if the set of atomic formulae of h' collectively

eliminates every matching to every negative example is h' consistent with the negative examples.

The greedy method produces h' as follows: Starting with just the quantifiers, we add the atomic formula of h that eliminates the largest number of matches to negative examples. Ties are broken arbitrarily. Continuing the illustrative example from the proof of Theorem 3, using the hypothesis h given in the first graph of Figure 7, this atomic formula could either be the one specifying that the node at the top of the graph (call it x) has *shape*(x) = *regular_polygon* or the one specifying that the other node (call it y) has *shape*(y) = *circle*. The first formula eliminates 2 matchings from each of the second and fifth negative examples (counting left to right, row by row), and one matching from the third, fourth, and sixth negative examples, for a total of seven matchings eliminated. The second also eliminates seven matchings. Assume we add the former to our evolving expression. To continue with the greedy algorithm, we now add the atomic formula of h that eliminates the largest number of the remaining matches to negative examples, and continue in this way until all matches are eliminated. In our illustrative example, the next formula would be *shape*(y) = *circle*, which eliminates three of the five remaining matchings. Specifying that one object must be close to the other, but not touching or inside, e.g., by the formula *distance_between*(x, y) = *close*, eliminates the last two matchings, and we get the hypothesis

$$h' = \exists^* x, y : (\textit{shape}(x) = \textit{regular\_polygon}) \text{ and}$$

$$(\textit{shape}(y) = \textit{circle}) \text{ and } (\textit{distance\_between}(x, y) = \textit{close}).$$

Except for the fact that we are counting matchings, this greedy algorithm is the same as that used in Haussler (1988) for the attribute-based case. Now, however, we will make a slight modification to this algorithm, as suggested by M. Warmuth: Instead of continuing until all matchings are eliminated, we will continue only until the number of matchings that remain is less than $\epsilon m/2$, where m is the total sample size (positive and negative examples) and ϵ is a bound on the error we can allow in the final hypothesis. This implies that the final hypothesis will be consistent with all but at most $\epsilon m/2$ of the training examples as required by Theorem 2, since there must be an uneliminated matching for each inconsistent negative example. (Here we assume perfect consistency with the positive examples, although this can also be relaxed somewhat.) The following Lemma bounds the complexity of this hypothesis.

Lemma 4. Let $\bar{s}$ be the minimum number of atomic formulae from h needed to eliminate all matchings to the negative examples. Then the hypothesis h' produced by the above modified greedy method will have no more than $\bar{s} \ln \left(\frac{2k!}{\epsilon}\right) + 1$ atomic formulae, where k is the maximum number of objects per scene.

Proof. Assume there are m negative examples. Hence we begin with at most $k!m$ matchings. Since $\bar{s}$ atomic formula suffice to eliminate all these matchings, there is at least one formula that eliminates a fraction $1/\bar{s}$ of them. Hence, since the greedy method picks the formula that eliminates the most matchings, after the first formula is added, there will remain no more than $(1 - 1/\bar{s})k!m$ matchings. Continuing this reasoning, using the fact that some formula always eliminates a fraction $1/\bar{s}$ of the remaining matchings, we conclude that after r formulae are added, there will remain no more than $(1 - 1/\bar{s})^r k!m$ matchings. So long as there is still one more formula that will be added to h', this latter number must be at least $\epsilon m/2$. Since $1 - 1/\bar{s} \leq e^{-1/\bar{s}}$, this implies that $e^{-r/\bar{s}}k!m \geq \epsilon m/2$, i.e. $r \leq \bar{s} \ln\left(\frac{2k!}{\epsilon}\right)$. Hence h' will have no more than $\bar{s} \ln\left(\frac{2k!}{\epsilon}\right) + 1$ atomic formulae. □

By applying Theorem 2 with $s = \bar{s} \ln\left(\frac{2k!}{\epsilon}\right) + 1$, this shows that for existential conjunctive target concepts with at most $\bar{s}$ atomic formulae, the learning algorithm given in Theorem 3, followed by the modified greedy simplification method suggested by Warmuth, gives a PAC learning algorithm that requires only a sample size m that is

$$O\left(\frac{\bar{s}}{\epsilon} \log \frac{k!}{\epsilon} \log \frac{kn}{\epsilon}\right).$$

For constant k this is within a poly-logarithmic factor of the lower bound given above at the end of Section 3.2. When $\bar{s}$ is small and n is large, the savings in sample size obtained by using the greedy method could be significant. However, as the method is described above, there is no room for heuristics to reduce the number of matchings that have to be examined (unlike in the initial process of forming the hypothesis), so this method will not be practical unless k is very small.

5. Extensions and open problems

Our definition of existential conjunctive concepts is obtained by taking a class of concepts defined on attribute-based domains that is well-studied from a learning point of view, namely the pure conjunctive concepts as they are called in Haussler (1988), and generalizing them by adding either single variables or pairs of distinct variables to the individual atomic formulae and allowing these variables to bind to the objects in a scene in any 1-1 mapping, i.e., prefixing the expression with the $\exists^*$ operator. There are obviously many other variants of this scheme that yield interesting classes of concepts from a learnability point of view.

One possibility is to use the standard $\exists$ operator instead of $\exists^*$, i.e., to allow more than one variable to map to the same object. This is an easy modification of the framework given in Section 1 above; we simply insist that binary relations be defined between any object and itself and drop the restriction that the mappings be 1-1 in the definitions of satisfaction and the "more general than" partial order.

Many AI learning systems avoid these many-one mappings because they make the expressions for simple concepts more complex and add additional overhead to heuristic search algorithms. The problem is that once you allow many-one mappings, you have to take additional steps to avoid unwanted interpretations. For example, at first glance the expression $\exists x, y : (shape(x) = square)$ and $(color(y) = red)$ seems to represent all scenes that contain one object that is square and another that is red, but in fact this expression is satisfied by some scenes that contain only one object, so long as that object is a red square. To avoid this interpretation, we must specify that x and y must be different objects by adding an atomic formula with something equivalent to a binary relation of identity that has a value *different* when applied to two distinct objects and a value *identical* when applied to an object and itself. (This relation is sometimes called "equals," but this invites confusion with the Boolean-valued binary relation that has value 1 if both objects have the same values for all attributes and 0 otherwise.) In general, if there are r variables then up to $r^2 - r$ additional atomic formulae must be added to the expression to force a 1-1 correspondence between variables and objects. However, the other side of this is that allowing many-one mappings increases the class of concepts we can represent. Several examples are given in Hayes-Roth (1978). Hence it would certainly be of interest to extend the results given here to cover this case.

Another direction of generalization is to allow some disjunction among the atomic formulae of the expression, i.e, begin with a class of expressions more general than the pure conjunctive concepts and then add variables and prefix the expression with either $\exists$ or $\exists^*$. Again, restricted cases that have been studied in the attribute-based domains would be logical choices. Among these are the internal disjunctive expressions of (Michalski, 1983) and the k-CNF and k-DNF expressions of Valiant (1984) (see also Haussler, 1988). In Valiant (1985) it is shown that existentially quantified k-DNF expressions with a bounded number of variables (similar to the bound we have used on the number of objects per scene) are PAC learnable in structural domains with Boolean relations. No queries are used in this method, but the computational effort grows exponentially in the bound k on the number of atomic formulae per conjunction. Hence the method is not practical when this bound is large.

Other directions of generalization might further exploit the fact that we are using structural domains. One possibility is to enhance the system of concept representation by allowing objects to have *types* as well, and having different sets of attributes be defined for objects of different types. In most applications, these types would also be partially ordered, with the objects of one type inheriting the attributes from the types above it in addition to its own indigenous attributes (see e.g., Winston, 1984, Chapter 8). In this enhanced representation, the set of binary relations relevant to an ordered pair of objects would also depend on the types of the objects. Computational time would be saved by disallowing matchings between objects whose types conflict.

Other extensions would be to allow the values of relations to be themselves

objects or scenes, thereby creating a recursive structure (Boriga, et al., 1986; Sammut, 1986), or allowing additional axioms that constrain the values of certain relations beyond those implicit in the tree-structured or linear ordering, e.g., an axiom that states that the relation **distance_between** must be symmetric (Kodratoff, 1986; Buntine, 1986).

This latter extension leads to the following more general question: To what extent can domain knowledge be incorporated into the learning methods we have described? When can it be used to replace the queries we have used to constrain the search for a hypothesis? Clearly a domain theory (Mitchell, et al., 1986) that is capable of determining if a given hypothesis is contained in the target concept could implement the oracle we have postulated for answering subset queries. This may lead to some interesting hybrid EBG/empirical learning systems.

Apart from extensions of the model, it would be interesting to know if the results we have given could be obtained with a simpler type of query, perhaps a *membership query,* in which the learning algorithm constructs an instance and asks if it is in the target concept. Except for some very special cases (corresponding to monotone conjunctions of Boolean relations (Angluin, 1988)), we have not been able to show this.

Finally, we may be able to avoid queries altogether by showing that for some useful classes of distributions on examples, there are algorithms that, given random examples, can be proven to produce existential conjunctive hypotheses that are nearly consistent in polynomial time with high probability. Proposition 4 could then be used to show that these are effective learning algorithms under this particular class of distributions. We are not currently aware of any results of this type in AI learning domains.

Acknowledgements

This paper is in part a revised version of Technical Report UCSC-CRL-87-1, published by the Department of Computer Science, University of California, Santa Cruz. I gratefully acknowledge the support of ONR grant N00014-86-K-0454.

I would also like to thank Lenny Pitt, Les Valiant, Nick Littlestone, Manfred Warmuth and an anonymous referee for helpful comments concerning this material.

Notes

1. In some cases the labeling of the nodes and edges in the graphs is "rich and varied" enough so that most labels are incompatible, i.e. neither is stronger than the other. In this case the number of possible 1-1 mappings satisfying the requirements of a "more general than" matching may be severely limited and an exhaustive search may be feasible, even when the number of nodes is large.
2. Here only positive examples are used and the object is to find a maximally specific consistent concept meeting certain criteria.
3. Here and in Proposition 4 we are suppressing some additional measurability assumptions required in the general form of the theorem since they are not relevant for our application.

References

Angluin, D. (1988). Queries and concept learning. *Machine Learning, 2,* 319–342.

Baum, E. & Haussler, D. (1989). What size net gives valid generalization? *Neural Computation, 1,* 151–160.

Buntine, W. *Induction of Horn clauses: Methods and the plausible generalization algorithm.* (Technical Report). New South Wales, Australia: New South Wales Institute of Technology, Department of Computer Science.

Blumer, A., Ehrenfeucht, A., Haussler, D. & Warmuth, M. (1989). Learnability and the Vapnik-Chervonenkis dimension. *J. ACM, 36.*

Boriga, A., Mitchell, T. & Williamson, K. Learning improved integrity constraints and schemas from exceptions in data and knowledge bases. In M. Brodie and J. Mylopoulos, (Eds.), *On knowledge base management systems,* New York: Springer-Verlag.

Bundy, A., Silver, B. & Plummer, D. (1985). An analytical comparison of some rule-learning programs. *Artificial Intelligence, 27,* 137–181.

Cohen, P. & Feigenbaum, E. (1982). *Handbook of Artificial Intelligence* (Vol. 3). William Kaufmann.

Dietterich, T.G. & Michalski, R.S. (1983). A comparitive review of selected methods for learning from examples. In *Machine learning: An artificial intelligence approach.* Palo Alto, CA: Tioga Press.

Duda, R. & Hart, P. (1973). *Pattern classification and scene analysis.* John P. Wiley & Sons.

Ehrenfeucht, A., Haussler, D., Kearns, M. & Valiant, L. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation, 82.*

Garey, M. & Johnson, D. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* San Francisco, CA: W. H. Freeman.

Gill, J. (1977). Probabilistic Turing machines. *SIAM J Comput, 6,* 675–695.

Haussler, D. (1987). *Learning conjunctive concepts in structural domains.* (Technical Report UCSC-CRL-87-01). Santa Cruz, CA: University of California.

Haussler, D. (1988). Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence, 36,* 177–221.

Haussler, D., Kearns, M., Littlestone, N. & Warmuth, M. (1988). *Equivalence of models for polynomial learnability.* (Technical Report UCSC-CRL-88-06). Santa Cruz, CA: University of California.

Haussler, D. & Welzl, E. (1987). Epsilon nets and simplex range queries. *Discrete and Comp. Geometry, 2,* 127–151.

Hayes-Roth, F. & McDermott, J. (1978). An interference matching technique for inducing abstractions. *CACM, 21,* 401–410.

Kearns, M., Li, M., Pitt, L. & Valiant, L. (1987). Recent results in Boolean concept learning. *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 337–352). Irvine, CA.

Kodratoff, Y. & Ganascia, J. (1986). Improving the generalization step in learning. In R. Michalski, J. Carbonell & T. Mitchell (Eds.), *Machine learning II.* Los Altos, CA: Morgan Kaufmann.

Knapman, J. (1978). A critical review of Winston's learning structural descriptions from examples. *AISB Quarterly, 31,* 319–320.

Michalski, R. S. (1983). A theory and methodology of inductive learning. In *Machine learning: An artificial intelligence approach.* Palo Alto, CA: Tioga Press.

Mitchell, T.M. (1980). *The need for biases in learning generalizations.* (Technical Report CBM-TR-117). New Brunswick, NJ: Rutgers University, Department of Computer Science.

Mitchell, T.M. (1982). Generalization as search. *Artificial Intelligence, 18,* 203–226.

Mitchell, T.M., Keller, R.M. & Kedar-Cabelli, S.T. (1988). Explanation-based generalization: A unifying view. *Machine Learning, 1,* 47–80.

Muggleton, S. & Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 339–352). Ann Arbor, MI.

Natarajan, B.K. (1989). On learning sets and functions. *Machine Learning, 4,* 67–97.

Pearl, J. (1978). On the connection between the complexity and credibility of inferred models. *Int. J. Gen. Sys., 4,* 255–264.

Pitt, L. & Valiant, L.G. (1988). Computational limitations on learning from examples. *Journal of the ACM, 35,* 965–984.

Sammut, C. & Banerji, R. (1986). Learning concepts by asking questions. In R. Michalski, J. Carbonell & T. Mitchell, (Eds.), *Machine learning II.* Los Altos, CA: Morgan Kaufmann.

Stepp, R. (1987). Machine learning from structured objects. *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 353–363). Irvine, CA.

Subramanian, D. & Feigenbaum, J. (1986). Factorization in experiment generation. *Proceedings of the AAAI-86* (pp. 518–522). Philadelphia, PA.

Utgoff, P. (1986). Shift of bias for inductive concept learning. In R. Michalski, J. Carbonell & T. Mitchell, (Eds.), *Machine learning II,* Los Altos, CA: Morgan Kaufmann.

Valiant, L.G. (1984). A theory of the learnable. *CACM, 27,* 1134-1142.

Valiant, L.G. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560–566). Los Angeles, CA.

Vapnik, V.N. (1982). *Estimation of dependences based on empirical data.* New York: Springer-Verlag.

Vapnik, V.N. & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Th. Prob. and its Appl., 16,* 264–280.

Vere, S.A. (1975). Induction of concepts in the predicate calculus. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence* (pp. 281–287). Tbilisi, USSR.

Winston, P. (1975). Learning structural descriptions from examples. In P. H. Winston, (Ed.), *The psychology of computer vision.* New York: McGraw-Hill.

Winston, P. (1984). *Artificial intelligence.* Addison-Wesley.