Learning Conjunctive Concepts in Structural Domains

David Haussler

Department of Computer Science,
University of California, Santa Cruz, CA 95064 USA [1]

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 space approach and how it shows that it is unlikely that this class of concepts is polynomially learnable from random examples in the sense of Valiant. On the other hand, we show that this class is polynomially learnable if we allow a larger hypothesis space. This result holds for any fixed number of objects per scene, but the algorithm is not practical unless the number of objects per scene is very small. We also show that heuristic methods for learning from larger scenes are likely to give an accurate hypothesis if they produce a simple hypothesis consistent with a large enough random sample.

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 and McDermott, 1978; Knapman, 1978; Michalski, 1980, 1983; Dietterich and Michalski, 1983; Bundy et al., 1985; Sammut and Banerji, 1986; Kodratoff and Ganascia, 1986). Whereas much of the earlier learning work, especially that associated with the field of Pattern Recognition (see e.g. Duda and Hart, 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 ... 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 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.

Mitchell (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 and McDermott, 1978). 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, 1986) it is shown that even using the hypothesis space of conjunctive concepts in an attribute-based domain (corresponding to existential conjunctive concepts on scenes with only one object), if the number of attributes is large then the size of G can grow exponentially in the number of examples. However, in this case S never contains more than one hypothesis (see Bundy et al., 1985), so the learning task described above can still be solved efficiently by computing only S (using the

[1] The author gratefully acknowledges the support of ONR grant N00014-86-K-0454.

positive examples) and then checking to see if any negative example is contained in S in a second pass through the sample. We show that it is unlikely that such an efficient strategy exists for existential conjunctive concepts on domains with more than one object per scene. 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 two-valued (Boolean) attributes,

then using the hypothesis space of existential conjunctive concepts and letting the number of attributes grow, not only can the size of both S and G grow exponentially in the number of examples, but it is unlikely that any efficient method (version space or not) exists for solving the learning task above, since the version space emptiness problem is NP-complete, i.e. it is NP-complete to determine if there is any existential conjunctive concept consistent with a given sample (Theorem 1).

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 (by running the algorithm until the version space is either empty or reduced to one 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; Valiant, 1985). Using the techniques of (Pitt and Valiant, 1986), we show (Theorem 2) that it is also unlikely that there is an efficient learning algorithm for existential conjunctive concepts using random examples in the sense defined by Valiant, even with the same restrictions imposed above (i.e. 2 objects per scene, no binary relations, Boolean attributes).

To balance these negative learning results, we also obtain some positive results. First, we show that for any fixed maximum number k of objects per scene, existential conjunctive concepts can be efficiently learned from random examples in the sense of Valiant if we use an extended hypothesis space, i.e. if we restrict the target concept to be existential conjunctive with at most k variables but allow the hypothesis to be chosen from a larger class of concepts (Theorem 3). Similar results are given for other types of concept classes in (Pitt and Valiant, 1986). The intuition behind this type of result is that sometimes by replacing a detailed and precise hypothesis space by a larger but more crudely organized one, our search for a consistent hypothesis may become easier. However, because our algorithm uses a brute force translation from a structural domain into an attribute-based domain (considering all possible bindings of objects to variables), it is not practical for k larger than 2 or 3.

In addition to being computationally expensive when there are many objects per scene, the algorithm used in Theorem 3 also requires more random examples to obtain a given level of confidence in the accuracy of the hypothesis produced than would a method that produced consistent existential conjunctive hypotheses. This is because a "shift" to a more weakly biased hypothesis space (Utgoff, 1986) also weakens the statistical leverage we have in establishing the accuracy of the hypothesis within a given confidence interval. We can avoid both of these problems by restricting ourselves to existential conjunctive hypotheses as before, but using heuristics to prune the search for a consistent hypothesis (Vere, 1975; Hayes-Roth and McDermott, 1978[2]; Michalski, 1980; Dietterich and Michalski, 1983). From our NP completeness results, we do not expect that any efficient heuristic algorithms will always find a consistent hypothesis whenever there is one. However, we show that when a heuristic algorithm does find a simple hypothesis consistent with a large enough random sample, then this hypothesis will with high probability be a good approximation of the target concept in the sense of Valiant (Valiant, 1984), regardless of the method used to find it (Theorem 4). This theorem is established using the methodology of (Haussler, 1986), in which the bias of a hypothesis space is quantified by measuring its Vapnik-Chervonenkis dimension. Then, using a general probabilistic result (Vapnik and Chervonenkis, 1971; Blumer et al., 1986), this dimension is converted into the number of random examples required to guarantee that any consistent hypothesis is accurate with high probability.

[2] Here only positive examples are used and the object is to find a maximally specific consistent concept meeting certain criteria.

Summary of Definitions

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 1 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 are used only in logical formulae that represent concepts, as defined below. 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.

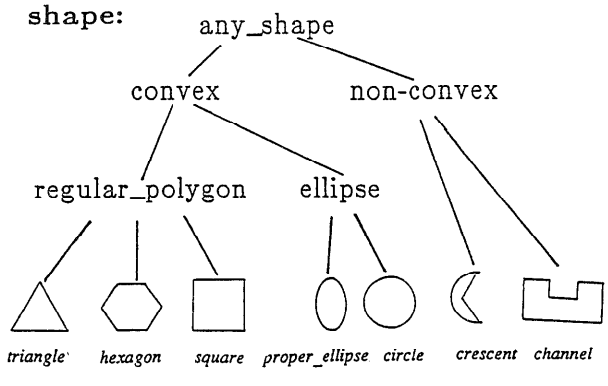

Figure 1.

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 coor-

dinates 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 2.

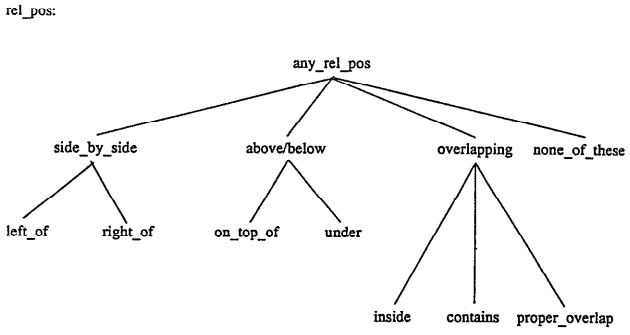

Figure 2.

Henceforth we will assume a fixed set R of relations consisting of n attributes $A_1, ..., A_n$ and l binary relations $B_1, ..., B_l$. Under this assumption, a scene with k objects is represented as a complete directed graph on k 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 a node representing obj_1 to a 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). This representation is illustrated in Figures 3a and 3b, where the triples in the nodes of Figure 3b give the values of the attributes *size, shaded* and *shape*, respectively and the pairs on the edges the values of the relations *rel_pos* and *distance_between*, respectively.

By using variables to denote unknown objects, we can define the set of (elementary) *atomic formulae* (*atoms*) over R as in (Michalski, 1983). Atomic formulae are either *unary* or *binary*. A unary atom $f(x)$, 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 atom $f(x)$ 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. In the later 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 the restrictions in $f(x)$. A binary atom $f(x,y)$, 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 atom $f(x,y)$ if the binary relation B between these objects complies with the restrictions in $f(x,y)$.

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

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

where $s \geq 1$ and each x_j, $1 \leq j \leq r$, is a variable and each f_i, $1 \leq i \leq s$, is an atom over R involving either a single variable or an ordered pair of distinct variables as defined above. We have dropped the names of the variables appearing in the individual atoms to simplify the notation. The first part of this expression (up to the colon) may be read "there exist distinct objects x_1 up to x_r such that ..." Thus a scene *satisfies* ϕ if it contains r *distinct* objects $obj_1, ..., obj_r$ 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.

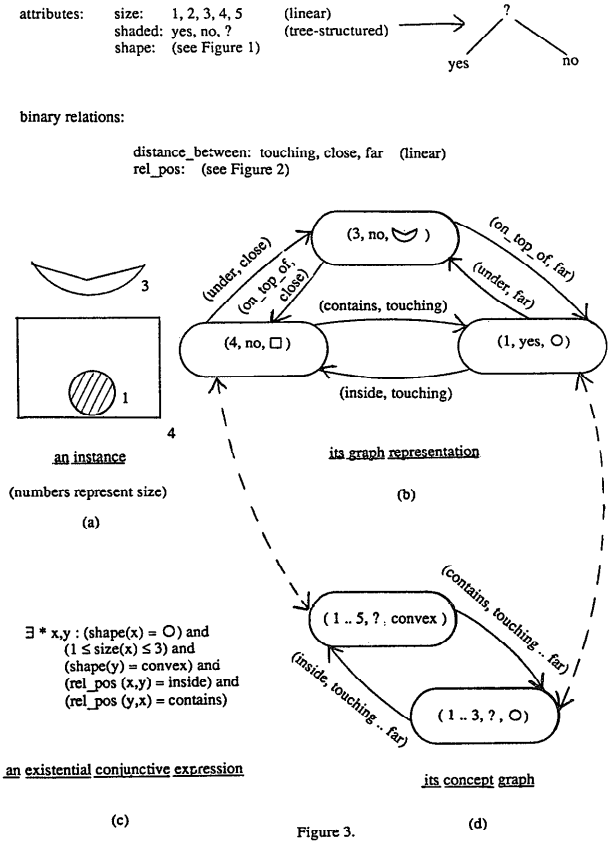

Figure 3.

The set of all scenes over R that satisfy ϕ is called the *concept* represented by ϕ, and the class of all such sets (varying ϕ) is referred to as the class of *existential conjunctive concepts*. The expression ϕ defined above can also be represented as a complete directed graph on r nodes, similar to the way a scene is represented (see Figure 3d). In this case, each node represents a variable of ϕ and the labels of nodes and edges represent restrictions imposed by the atoms 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 atom is present for a given variable or pair of variables that involves a given relation, we use the root value of a tree structured relation and the entire range of a linear relation. 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 mentioned in the introduction. 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 $\phi_2 \subseteq \phi_1$. However, since ϕ_1 and ϕ_2 are in general infinite sets, this is not a useful definition from a computational point of view. To define this relation 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, then it is easily verified that G_1 is more general than G_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 and Feigenbaum, 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 graph corresponding to the scene. The two dashed lines between the nodes in Figure 3d and the corresponding nodes in Figure 3b illustrate a mapping that shows that the scene in Figure 3a is an instance of the concept in Figure 3c.

Summary of Theorems

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. □

One sidelight of the proof of the above theorem 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 with expressions that 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, 1986), 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.

Corollary 1. The size of the sets S and G maintained in Mitchell's version space algorithm for existential conjunctive concepts can, in the worst case, be exponential in the number m of examples and the number n of attributes defined on objects in these examples, even when there are no binary relations defined, each attribute is Boolean valued, and each example contains exactly two objects. □

Theorem 1 and Corollary 1 may be taken as evidence that existential conjunctive concepts are perhaps inherently difficult to learn, even when only a few objects are involved. Following (Pitt and Valiant, 1986), can formalize this tentative conclusion.

In analogy with the class P of problems solvable in polynomial time by a deterministic algorithm, the class RP is defined as the class of problems that can be solved "probabilistically" in polynomial time by a deterministic algorithm that is also allowed to flip a fair coin to decide its next move (Gill, 1977). Here we say that the algorithm solves the problem probabilistically if whenever there is no solution, it answers truthfully, saying that there is no solution, and whenever there is a solution, it finds one (or indicates that one exists) with probability at least $1 - \delta$, where δ can be made arbitrarily small. Rabin's probabilistic algorithm for testing if an integer is composite (i.e. not prime) is a classic example of such an algorithm (Rabin, 1976).

It is easy to show that $\mathbf{RP} \subseteq \mathbf{NP}$, the class of problems solvable in polynomial time by a nondeterministic machine. Furthermore, just as it is strongly suspected that $\mathbf{P} \neq \mathbf{NP}$ it is also strongly suspected that $\mathbf{RP} \neq \mathbf{NP}$. We can show that unless $\mathbf{RP} = \mathbf{NP}$, Theorem 1 implies that existential conjunctive concepts are not polynomially learnable from random examples in the sense first defined by Valiant in (Valiant, 1984) (see (Haussler, 1987) for a formal definition of Valiant's learning framework from an AI perspective).

Theorem 2. If existential conjunctive concepts are polynomially learnable from two-object random examples then $\mathbf{RP} = \mathbf{NP}$. □

In other words, while we cannot prove that existential conjunctive concepts are not polynomially learnable from random examples in the sense of (Valiant, 1984), we can show that an efficient algorithm for learning existential conjunctive concepts from random examples would amount to a major breakthrough in complexity theory, similar to resolving the P verses NP issue. Many other results of this type, for different concept classes, are given in (Pitt and Valiant, 1986).

In contrast to this result, using other techniques from (Pitt and Valiant, 1986), we can show

Theorem 3. Existential conjunctive concepts are polynomially learnable from k-object random examples for any fixed k if we allow our learning algorithm to produce a hypothesis that is not existential conjunctive. □

The proof of this result involves transforming the problem of learning existential conjunctive concepts on an instance space with k objects per scene into the problem of learning $k!$-CNF concepts (Conjunctive Normal Form concepts with at most $k!$ atoms per clause) in an attribute-based instance space. Since only a small fraction of such CNF concepts are needed to represent existential conjunctive concepts from the original instance space, this is actually a much larger hypothesis space. Techniques of (Valiant, 1984) or (Haussler, 1986) can be used to find $k!$-CNF concepts that, with high probability, approximate the existential conjunctive target concept to any desired accuracy. The drawback is that the time required for these techniques grows exponentially in $k!$, and hence the algorithm is not really practical for k larger than 2 or 3.

For larger k, the best available general learning algorithms are still the ones that use the hypothesis space of all existential conjunctive concepts, but employ heuristics to prune the search for a consistent hypothesis in this space, as mentioned in the introduction. As in the Valiant framework, let us assume that our sample is produced by drawing random examples of an unknown existential conjunctive target concept. The *error* of a hypothesis is defined as the probability that it will misclassify a randomly drawn example.

Theorem 4. Consider k-object examples on an instance spaces defined by n relations (unary or binary). There is a sample size m that is

$$O\left[\frac{s\log(skn)}{\varepsilon}\log\frac{s\log(skn)}{\varepsilon\delta}\right],$$

such that for any target concept c, given m independent random examples of c, the probability that all consistent existential conjunctive hypotheses with at most s atoms have error less than ε is at least $1-\delta$. Moreover, this holds independent of the choice of the probability distribution on the instance space governing the generation of examples. □

Since s is a measure of simplicity for existential conjunctive hypotheses, this result essentially says that if the sample size is large enough, then all simple hypotheses that are poor approximations to the target concept will be explicitly contradicted by an example. Thus the remaining (i.e. consistent) simple hypotheses (if any) will all be good approximations to the target concept. Hence the simplicity of the hypothesis produced by a heuristic learning algorithm can have a significant effect on the confidence we have in its accuracy, a form of Occam's Razor (see also Pearl, 1978; Blumer et al., 1986).

In (Haussler, 1986) similar results were obtained for pure conjunctive concepts in an attribute-based domain, with sample size of

$$O\left[\frac{s\log(sn)}{\varepsilon}\log\frac{s\log(sn)}{\varepsilon\delta}\right]$$

In fact these results are a special case of Theorem 4 with $k = 1$, corresponding to the case when each scene contains exactly one object, and hence the structural domain is reduced to an attribute-based domain. What is significant is that in structural domains, the sample size required grows only logarithmically as the number k of objects per scene is increased.

The key problem that remains is finding the best heuristic. Theorem 2 shows that it is unlikely that we will find a heuristic that is guaranteed to work on random examples. However, it still might be the case that by the addition of queries of the type discussed in (Angluin, 1986), a polynomial learning algorithm for existential conjunctive concepts could be found that always produces simple, consistent existential conjunctive hypotheses, and whose performance does not degrade badly with increasing k like the algorithm of Theorem 3 above. The work in (Sammut and Banerji, 1986) is a step in this direction, but as yet there has been no careful performance analysis of the techniques used there.

Acknowledgements. I would like to thank Les Valiant, Nick Littlestone, Manfred Warmuth and Pat Langley for helpful conversations concerning this material.

References:

(Angluin, 1986) D. Angluin. *Types of queries for concept learning.* Technical Report YALEU/DCS/TR-479, Yale University, 1986.

(Blumer et al., 1986) A. Blumer, A. Ehrenfeucht, D. Haussler and M. Warmuth. Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension. In *18th ACM Symp. Theor. Comp.*, Berkeley, CA, 1986.

(Bundy et al., 1985) A. Bundy, B. Silver, and D. Plummer. An analytical comparison of some rule-learning programs. *Artif. Intel.*, 27: 137-181, 1985.

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

(Dietterich and Michalski, 1983) T.G. Dietterich and R.S. Michalski. A comparative review of selected methods for learning from examples. In *Machine learning: an artificial intelligence approach*, Tioga Press, Palo Alto, CA, pages 41-81, 1983.

(Duda and Hart, 1973) R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

(Gill, 1977) J. Gill. Probabilistic Turing machines. *SIAM J. Comput.*, 6 (4): 675-695, 1977.

(Haussler, 1986) D. Haussler. Quantifying the inductive bias in concept learning. In *Proc. AAAI '86*, pages 485-489, Philadelphia, PA, 1986.

(Haussler, 1987) D. Haussler. Bias, Version Spaces and Valiant's Learning Framework. In *Proc. 4th Int. Workshop on Machine Learning*, Irvine, CA, June 1987, to appear.

(Hayes-Roth and McDermott, 1978) F. Hayes-Roth and J. McDermott. An interference matching technique for inducing abstractions. In *Comm. ACM*, 21 (5): 401-410, 1978.

(Kodratoff and Ganascia, 1986) Y. Kodratoff and J. Ganascia. Improving the generalization step in learning. In *Machine Learning II*,, pages 215-244, R. Michalski, J. Carbonell and T. Mitchell, eds., Morgan Kaufmann, Los Altos, CA, 1986.

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

(Michalski, 1980) R.S. Michalski. Pattern Recognition as rule-guided inductive inference. *IEEE PAMI*, 2 (4): 349-361, 1980.

(Michalski, 1983) R.S. Michalski. A theory and methodology of inductive learning. In *Machine learning: an artificial intelligence approach*, pages 83-134. Tioga Press, Palo Alto, CA. 1983.

(Mitchell, 1982) T.M. Mitchell. Generalization as search. *Art. Intell.*, 18: 203-226, 1982.

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

(Pitt and Valiant, 1986) L. Pitt and L.G. Valiant, *Computational Limitations on Learning from Examples.* Technical Report TR-05-86, Aiken Computing Lab., Harvard University, 1986.

(Rabin, 1976) M.O. Rabin. Probabilistic Algorithms. In *Algorithms and Complexity: New Directions and Recent Results*, pages 21-39, J.F. Traub, Ed., Academic Press, New York, 1976.

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

(Utgoff, 1986) P. Utgoff. Shift of Bias for inductive Concept Learning. In *Machine Learning II.* R. Michalski, J. Carbonell and T. Mitchell, eds., Morgan Kaufmann, Los Altos, CA, 1986.

(Valiant, 1984) L.G. Valiant. A theory of the learnable. *Comm. ACM*, 27 (11): 1134-1142, 1984.

(Valiant, 1985) L.G. Valiant. Learning disjunctions of conjunctions. In *Proc. 9th IJCAI*, vol. 1, pages 560-566, Los Angeles, CA, August 1985.

(Vapnik and Chervonenkis, 1971) V.N. Vapnik and A.Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Th. Prob. and its Appl.*, 16 (2): 264-280, 1971.

(Vere, 1975) S.A. Vere. Induction of concepts in the predicate calculus. In *Proc. 4th IJCAI*, pages 281-287, Tbilisi, USSR, 1975.

(Winston, 1975) P. Winston. Learning structural descriptions from examples. In *The Psychology of Computer Vision.* McGraw-Hill, New York, 1975.