

INDUCTIVE LEARNING OF CATEGORIES FROM EXAMPLES USING MINIMUM COST REPRESENTATIONS

Steven L. Tanimoto
Department of Computer Science, FR-35
University of Washington
Seattle, Washington 98195
U.S.A.

The problem of learning categories from a sequence of examples is considered in terms of maintaining a minimum-cost representation for the set of categories. Categories here are subsets of a set of natural numbers. Computational aspects of the following problems are addressed: (1) how are the category representations updated as an incremental change is made to one category, (2) how can a minimum cost representation for a new category be obtained in terms of existing ones? and (3) how does the enlargement of the known universe of objects affect representations of known categories? We then discuss extensions of our methodology to domains which include structure in the categorical descriptions.

1. INTRODUCTION

1.1 Motivation

Machine acquisition of concepts from experience is an important goal for artificial intelligence. Knowledge bases, the primary source of limitation as well as power in "expert systems" should be able to grow and thus permit the systems to improve [2]. The automation of this growth of system knowledge is then of obvious interest. In this paper we focus on a simplified learning scenario where the concepts are restricted to be categories.

1.2 Past Work

The process of learning categories from examples has been studied from several different approaches including (a) trainable pattern classifiers working with real-valued features [6] and (b) discovery of classification rules using variable-valued logic [5], and (c) formation of structural descriptions in terms of known substructures [7]. Here we discuss the learning of categories in the simple mathematical framework of countable sets as is done by Lenat [4] so as to make clear some of the computational issues involved.

1.3 Problem Formulation

The kind of learning we wish to understand is the acquisition of new "concepts" (categories of objects) and the continual refinement of the representations for known categories as new examples of old categories are encountered.

For example, suppose a person Sue previously unexposed to computers is shown a personal computer and told "This computer, the ACME 1, is a home system". Sue should then know that there is a category "computer", a category "home system" and that the ACME 1 is an example of each. She may also be able to infer that the category "home system" is included in the category "computer" if the context supports that inference, e.g. if she had previously been told "Let's discuss types of computers". After being shown several more home systems she may understand "home system" to be the set {ACME 1, Apple 15, JETSETTER, ACME 2}. Suppose she is then told that the following are "cheapo's": {ACME 1, Apple 15, ACME 2}. Being unfamiliar with the computer domain, we would not expect Sue to be able to distinguish the cheapo's from the non-cheapo's except by using the information she has been told. If Sue is thinking, we would naturally expect her to infer that JETSETTER is an example of a home system that is not a cheapo. In fact, if she remembers which are the home systems, then she has an easy method for remembering the cheapo's in:

$$\text{cheapo's} = (\text{home systems}) - \{\text{JETSETTER}\}.$$

We would like computer programs to make similar inferences in the course of learning categories.

2. MINIMUM COST REPRESENTATIONS

2.1 Representation Scheme

Given a set S of natural numbers, a representation for S written $R(S)$ is any one of the following forms (as applicable) and $C(R(S))$, the cost of $R(S)$, is as given. (1) if S is known

to be empty, then $R(S) = \{\}$ and $C(R(S)) = 1$.
 (2) if S is a singleton, then $R(S) = \{a_j\}$ and $C(R(S)) = 1$. (3) if S is the extension of an initial predicate, then $R(S) = \{x | P_j(x)\}$ and $C(R(S)) = 1$.
 (4) if S is a union of two other sets, then $R(S) = (R(S_1) \cup R(S_2))$ and $C(R(S)) = C(R(S_1)) + C(R(S_2)) + 1$. (5) if S is an intersection, then $R(S) = (R(S_1) \cap R(S_2))$ and $C(R(S)) = C(R(S_1)) + C(R(S_2)) + 1$. (6) if S is a complement, then $R(S) = (\sim R(S_1))$ and $C(R(S)) = C(R(S_1)) + 1$.

The cost of S is the minimum cost over all representations of S . Our cost function, while arbitrarily chosen, yields cost values that roughly reflect the amount of symbology necessary to store representations.

2.2 Updating a Single Category

The amount of the change in cost for assimilation of a new fact is bounded (when only one category is represented). The following proposition describes the cost change for updating S to S' according to $S' = S \cup \{a_i\}$ where $a_i \notin S$.

Proposition 1: $C(S') - C(S) \leq 2$
 and $C(S) - C(S') \leq 3$

where $C(X)$ denotes the cost of X .

As will become clear from subsequent discussion, when several categories are simultaneously represented, an incremental change in one may affect the total cost of representation beyond the bounds of proposition 1.

3. COMPUTATIONAL CONSIDERATIONS

3.1 Computing Minimum Cost Representations

Even the problem of transforming a list representation of a category into a minimum cost representation is a nontrivial problem in combinatorial optimization. The computational complexity of this and related problems of minimum cost representation is of interest because the high cost of such processing illustrates the subtlety in learning even simple concepts.

3.2 Multiple Categories

Suppose that there are initial predicates $P_0, \dots, P_k$ and existing categories $S_0, \dots, S_m$. Given a new category S_{m+1} we wish to find a minimum cost representation for $\{S_0, \dots, S_{m+1}\}$. One approach is to find a minimum cost rep. for S_{m+1} in terms of the rep's.

for $S_0, \dots, S_m$. However it is possible that the rep's. of some or all of $S_0, \dots, S_m$ should change with the assimilation of S_{m+1} . We shall address this subsequently.

As we permit LS to handle multiple categories we need an additional representation form: $R(S) = R(S_1)$. However we now must modify our cost rules so that any given category only gets charged once.

$C(R(S))$ = the number of occurrences of all of the following in $R(S)$:

$n, u, \sim, \{a_i\}, \{\}, \{x | P_j(x)\}, R(S_1^1),$

where a_i is any integer, P_j is any predicate and S_1^1 is any category whose representation is charged separately. Note that the assimilation of new information may require changing representations for old information.

3.3 Circularity in Representation

As category representations are updated, care must be taken to avoid circular representations. If at time 1 we have S_2 represented as or in terms of $R(S_1)$, then we may not at a later time (2) change the representation of S_1 to be in terms of S_2 unless S_2 is first made independent of S_1 . Prevention of circularity may be handled by maintaining a partial order on the categories, implemented as a directed acyclic graph. The algorithm below builds the representation for a collection of categories in a non-circular way by representing the simplest (least costly) categories first and building up to "inherently costly" categories (those that would be expensive if expressed independently of other categories), thus progressing through a "topological sort" of the partial order.

3.4 Brute-force Method

The "exhaustive" computation scheme follows:
 FINISHEDREPS $\leftarrow$ set of extensions of initial predicates; $I \leftarrow 1$; WHILE categories remain to be represented DO BEGIN ADD representations for all categories having cost at most I (in terms of representations already in FINISHEDREPS; $I \leftarrow 1$; END.

In order to determine whether (and how) a category S_i can be represented in terms of the members of FINISHEDREPS with a cost of I , we can simply generate all representations (finitely many when U is finite) and test to see if any represent the category under consideration. It is not hard to see that the combinatorial explosion is severe here. If the size of the universe is N , and the number

of representations on FINISHEDREPS is q , then $O((N + q)(I + 1)/2)$ representations must be considered using this scheme.

3.5 Reducing the Computational Effort

If q , the number of complete representations is much less than n , the size of the universe, then the following approach is economical. Combinations (of the existing q categories) of successively greater cost are generated using $\sim$, $\cup$, and $\cap$ but no singleton sets. The category S_i which is to be represented, is "matched" to each of these categories as it is generated. In matching S_i with an S_j (set represented using some combination of existing Categories), compute $S_{ij1} = S_i - S_j$, the set of numbers in S_i missing from S_j and $S_{ij2} = S_j - S_i$, the set of numbers which need to be removed from S_j in forming a representation for S_i in terms of S_j . We call $d_{ij} =$

$|S_{ij1}|$ the deficiency of S_j (for S_i) and $e_{ij} = |S_{ij2}|$ the excess of S_j (for S_i). If $C(R(S_j)) + (2d_{ij} - 1) + (2e_{ij} + 1) \leq I$ then we have a representation for S_i of the form $((\dots((R(S_j) \cup \{x_1\}) \cup \{x_2\}) \cup \dots \cup \{x_{d_{ij}}\}) \cap (\{y_1\} \cup \{y_2\}) \cup \dots \cup \{y_{e_{ij}}\})$.

The operation of determining the deficiency set and the excess set is analogous to a process referred to as "interference matching" [3]. The above algorithm can take advantage of an additional short cut (without changing its solutions). From I and $C(R(S_j))$ the permissible deficiency plus excess is easily computed. As S_{ij1} and S_{ij2} are being computed it may be apparent that the limit will be exceeded. At such a point, further consideration of S_j can be aborted for the current value of I .

3.6 Supporting Categories

In some cases a collection of categories may have their cost reduced by introduction of a new category. If we let our system construct one, we term it a supporting category. While we have heuristics to generate supporting categories that reduce costs in specific situations, the problem of finding them in general is an open one.

3.7 Extensions

Our study has primarily been concerned with sets of natural numbers. There exist straight-

forward means to represent and to compute costs for structured objects such as n -tuples, labelled graphs and strings. Once again, computational complexity is a problem as one confronts the problem of subgraph isomorphism [1].

4. IMPLEMENTATION

A preliminary implementation has shown that additional heuristics and/or parallel processing are needed to make minimization of cost computationally feasible. Approximate solutions would be acceptable in most AI applications and this area is open for future work.

5. CONCLUSION

The maintainance of a minimum cost representation for a set of categories is desirable because relationships among categories are clarified and space requirements for the stored information are low. However, the complexity of computing the representations is high. Some heuristics have been given here and more are needed to simplify such computations.

REFERENCES

- [1] Aho, A.V., J.E. Hopcroft, and J.D. Ullman. The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.
- [2] Feigenbaum, E.A. "The art of artificial intelligence: themes and case studies of knowledge engineering." Proc. IJCAI-77. MIT, Cambridge, MA, 1977, pp. 1014-1029.
- [3] Hayes-Roth, F. and McDermott, J. "An interference matching technique for inducing abstractions." CACM Vol. 21, No. 5 (May 1978), pp. 401-411.
- [4] Lenat, D.B. "Automated theory formation in mathematics." Proc. IJCAI-77. MIT, Cambridge, MA, 1977, pp. 833-842.
- [5] Michalski, R.S. "Discovering classification rules using variable-valued logic system VL." Proc. Third Int. Joint Conf. on Artificial Intelligence, Stanford Univ, 1973, pp. 162-172.
- [6] Nilsson, N.J. Learning Machines. NY: McGraw-Hill (1965).
- [7] Winston, P. "Learning structural descriptions from examples." Ph.D. Dissertation, Dept. of Elect. Eng. TR-76, Project MAC TR-231, Sept. 1970.