Acquisition of Appropriate Bias for Inductive Concept Learning *

Paul E. Utgoff
Tom M. Mitchell

Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903

Abstract

Current approaches to inductive concept learning suffer from a fundamental difficulty; if a fixed language is chosen in which to represent concepts, then in cases where that language is inappropriate, the new concept may be impossible to describe (and therefore to learn). We suggest a framework for automatically extending the language in which concepts are to be expressed. This framework includes multiple sources of knowledge for recommending plausible language extensions.

I Introduction

We consider concept learning problems in which there is a domain of instances over which concepts (generalizations) are to be learned. A trainer presents a sequence of training instances, each labelled as a positive or negative instance of the concept. The task of the learner is to acquire the ability to correctly state, for every instance in the domain, whether that instance is an example of the concept. For any instance which has not been presented by the trainer, the learner must therefore inductively infer whether the instance is an example of the concept.

II Problem and Related Work

The inductive inference process is driven by two kinds of information. The first kind of information is classifications of training instances given by the trainer. The second kind of information, which we call *bias*, is broadly defined as anything which influences how the concept learner draws inductive inferences based on the observed training instances. Without such bias, the inductive inference process can perform no better at classifying unobserved instances than random guessing [5]. We are interested here in the issue of how an "appropriate" bias for inductive learning can be acquired automatically.

Bias can be built into a learning system in many ways. For example, Mitchell [4] and Vere [10] encode bias via an incomplete concept description language in which only certain partitions of the domain of instances are expressible. Waterman [11] encodes bias both in his concept description language and in his learning algorithm. Michalski [2, 3] encodes bias, not by limiting his concept description language, but by having a human user state the bias as rules for preferring one concept description to another.

*This work was supported by National Science Foundation grant GMCS80-08889.

III Approach

In this paper, we consider concept learning where the bias resides *solely* in the concept description language. To do this, we impose the constraint that a hypothesis can be described if an only if there exists a corresponding concept description in the learner's concept description language. Thus, we represent a particular bias by a particular set of describable concepts: the concept description language. This equivalence between the language and bias allows us to operate on bias by operating on the concept description language. Figure III-1 illustrates a framework for revising bias by revising the concept description language.

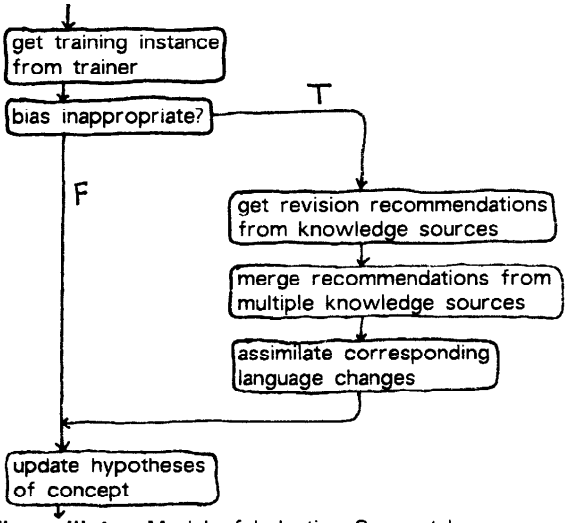

Figure III-1: Model of Inductive Concept Learner

A. Detecting Inappropriate Bias

Bias is *appropriate* to the extent that it causes the concept learner to draw correct inductive inferences. Bias is *strong* to the extent that the set of hypotheses to be considered by the concept learner is small. We are examining the process of revising bias by adding new concept descriptions to the learner's concept description language. Thus the revision process can be viewed as weakening strong bias in an appropriate manner.

Our method of detecting inappropriate bias is to detect incorrect inductive inferences which could have been drawn as a result of the bias. Accordingly, to detect that no consistent description of the observed instances exists in the concept description language is to prove that the bias is inappropriate, assuming that the training instances have been correctly classified by the trainer.

B. Revising Inappropriate Bias

When the existing concept description language has been identified as providing inappropriate bias, the bias must be revised by adding a new concept description to the language. For the purpose of induction, and to avoid the need for future revisions of bias, it is desirable to formulate a revision which corrects the discovered error and correctly anticipates the classifications of the unobserved instances.

1. Knowledge Sources for Recommending Revisions

We define a knowledge source as a procedure which uses available information (e.g. training instances, existing concept description language, context of concept learning problem) to recommend revisions to the concept description language which will render the concept description language more appropriate. Below, we consider two classes of knowledge sources.

One class of knowledge source for recommending language extensions is characterized as "syntactic" because the proposed recommendations are derived only by considering boolean combinations of currently describable concept descriptions. For example, it would be possible to specify a new concept description by creating a disjunction of concept descriptions (A ∨ B ∨ ...) which correctly classifies the training instances. Various researchers have discussed data-driven methods for describing such disjunctive concepts [1, 4, 2]. Similarly, it may be possible to specify a new concept description by creating a counterfactual, as per Vere [10], of concept descriptions (A ∧ ~(B ∧ ~...)) which correctly classifies the training instances.

derivable via the grammar is a concept description in the concept description language and describes (i.e. matches) the set of terminal strings derivable from the sentential form. We refer to the grammar as the generalization

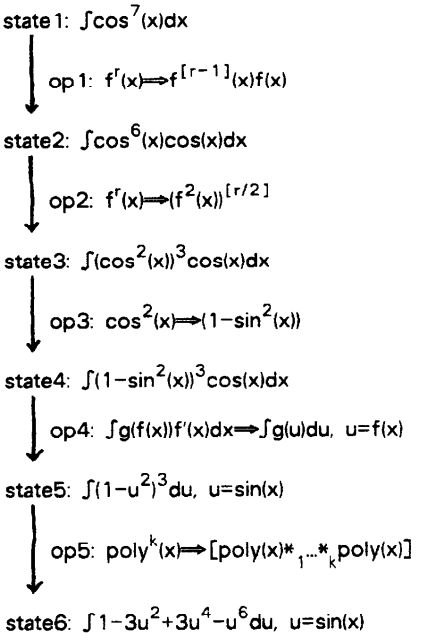

Figure III-3: Solution Path for $\int \cos^7(x)dx$

hierarchy. Suppose the concept to be learned is "situations for which the solution method shown in figure III-3 should be applied to solve a symbolic integration problem". If $\int \cos^5(x)dx$ and $\int \cos^7(x)dx$ are positive instances of this

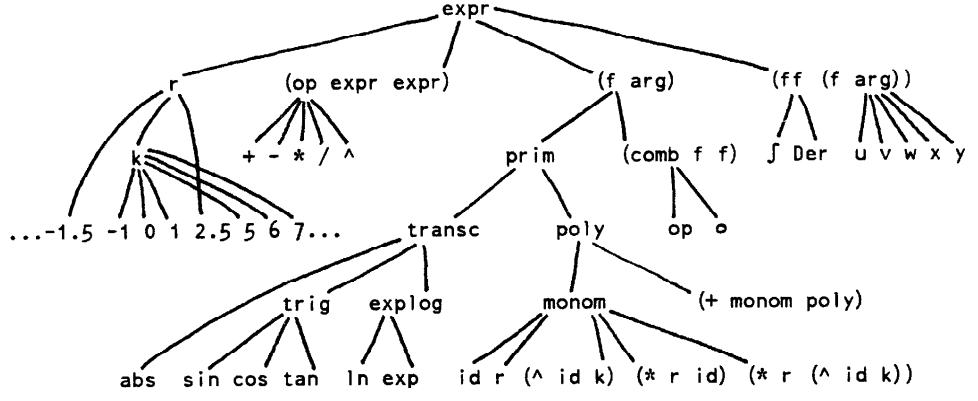

Figure III-2: A grammar for a concept description language for symbolic integration

Let us consider a hand-generated example of a concept learning problem from the task of learning problem-solving heuristics in the domain of symbolic integration [6]. The concept description language is specified by the formal grammar shown in figure III-2. Every sentential form

concept, and $\int \cos^6(x)dx$ is a negative instance, then there does not exist a concept description in the concept description language which is consistent with these observed instances. In this situation, a knowledge source KS_1 which produces succinct disjunctions may recommend

that the concept description language be revised to allow describing the disjunction $\int\cos^5 \vee {}^7(x)dx$. Similarly, a knowledge source KS_2 which searches for consistent counterfactuals may suggest revising the language to allow describing $\int\cos^k \wedge \sim^6(x)dx$.

A second class of knowledge sources for suggesting language revisions is characterized as "analytic" because the proposed recommendations are derived by analyzing the context in which the concept learning problem appears. For example, when the concepts to be learned are problem-solving heuristics, then a knowledge source based on analyzing solution traces (e.g. as suggested recently in [7]) is relevant to the task of generating new language terms. This knowledge source KS_3 uses knowledge about the set of available problem-solving operators, an explicit definition of the expected performance of heuristics, knowledge about heuristic search in general (e.g., definitions of concepts such as "solvable"), and knowledge about the task domain itself.

The solution trace analysis from KS_3 first produces a set of statements about various nodes in the search tree, which characterize reasons why the training instance is positive. These statements are then propagated through the problem-solving operators in the search tree to determine which features of the training instance were necessary to satisfy these statements. It is during this propagation and combination of constraints that new descriptive terms may be suggested.

For example, in the case of the solution path shown in figure III-3, suppose that the analysis determines that the solution path leads to a solution because state6 is of the form

$$\int poly(x)*_1 \ldots *_k poly(x)dx$$

which, in turn, satisfies the learner's definition of a solvable state. Then we can compute the set of states S_1 for which application of op5 leads to such a solvable state as

$$S_1 \leftarrow op5^{-1}(\int poly(x)*_1 \ldots *_k poly(x)dx)$$

giving

$$S_1 = \int poly^k(x)dx.$$

In turn, we can compute the set of states S_2 for which application of op4 leads to a state in S_1 as

$$S_2 \leftarrow op4^{-1}(intersection(range(op4), S_1))$$

giving

$$S_2 = \int poly^k(f(x))f'(x)dx.$$

By this repeated backward propagation of constraints through the solution tree, it can be determined that application of the solution method of figure III-3 leads to a solvable state when the initial state (in this case state1) is of the form $\int\cos^c(x)dx$ where c is constrained to satisfy the predicate "real(c) $\wedge$ integer((c-1)/2)", better known as "odd integer". The complete derivation is shown in [9]. Because the current language has no term corresponding to this predicate, the predicate can be used as the definition of a useful term to be added.

The approach sketched here is discussed further in [9, 7]. We believe this kind of analysis can provide strong guidance for revising the concept description language, and expect that studying this kind of knowledge source will continue to be a major focus of our future work.

2. Using Multiple Knowledge Sources

A knowledge source may produce any number of recommendations. Notice that a knowledge source such as KS_1 which can recommend a disjunctive description of a given set of positive training instances will always produce at least one recommendation. When several recommendations are available from one or more knowledge sources, a merging process must compare and weigh the recommendations against each other. This merging process may be based on many factors, including 1) the probability that a recommendation from a given source is correct, 2) whether the set of recommendations from one knowledge source intersects the set of recommendations from another source, and 3) the strength of the justification provided by the knowledge source for the recommended change (e.g. whether any of the concepts recommended for inclusion in the concept description language has been proven to be a sufficient condition for (subset of) the concept being learned).

For the example considered above, it is known that the analysis of the analytical knowledge source KS_3 computes a sufficient condition for concept membership. Therefore the merging routine can reject the recommendation from KS_1 in favor of the recommendation from KS_3 because the recommendation from KS_3 is a sufficient condition for concept membership and it is more general than that recommended by KS_1.

3. Assimilating a Proposed Change

When assimilating recommended changes to the concept description language, revisions of varying strengths may be possible. For example, to add a concept description which describes $\int\cos^5(x)dx \vee \int\cos^7(x)dx$, the disjunction itself could be added to the language as a permitted concept description. Alternatively, a term "k-new" could instead be added which represents the disjunction 5 $\vee$ 7. This latter revision causes the disjunction 5 $\vee$ 7 to be considered by the concept learner in contexts other than $\int\cos^r(x)dx$, and thus provides a more sweeping change to the language.

A major difficulty in assimilating new language terms lies in determining where a new term fits relative to existing terms in the generalization hierarchy. The ease with which a new term can be assimilated depends strongly on the form of its definition. It is relatively easy to assimilate a new term which is defined as a boolean combination of existing terms. For example, the new term "trig $\wedge$ ~tan" could be assimilated by placing it into the generalization hierarchy below "trig", and above all descendants of trig except "tan". As long as the number of descendants of "trig" and "tan" is finite, such expressions can be easily assimilated. In contrast, terms such as "odd integer" present considerable difficulty. While it is clear from the form of the definition that this term is a specialization of the term "r" (real number), it is *not* readily apparent that it is a specialization of "k" (integer). In order to determine this fact, the assimilator will require some knowledge about the meanings of the terms in this domain, and about the plus

and times functions used to define the new term, so that it can determine that every instance of the new term must also be an instance of k.

If it is not possible to determine the precise location where the new term belongs in the generalization hierarchy, then it may be possible, as a subproblem, to empirically learn the correct location of the new term in the generalization hierarchy. Surprisingly, certain kinds of errors in assimilating a new term into the generalization hierarchy do *not* prevent correct usage of the new term for inductive learning of subsequent concepts. For example, if "odd-integer" is added to the generalization hierarchy as a sibling of "integer", rather than as a specialization of "integer", each remains nevertheless a correctly defined term. Only their relationship to each other remains imprecisely defined. Thus, it is possible to use new terms in certain cases even before they have been correctly assimilated.

IV Summary and Open Issues

In the above discussion we have suggested a framework for extending the concept description language which drives inductive learning. While boolean combinations of existing terms sometimes lead to appropriate language revisions, they are not likely to produce radically new language terms such as "odd-integer" or "twice-integrable-function". A knowledge source such as KS_3 based on knowledge of heuristic search can, on the other hand, lead to such new terms.

The framework presented here for revising bias represents our initial approach to this problem. One major issue that we have not discussed is that the recommendations for language change produced by the knowledge sources will often be incorrect. To keep the concept description language from becoming "cluttered", a control strategy is needed for tentatively adding new terms to the language, and later backtracking on a revision if necessary. One interesting possibility is that control over the generation of training instances may provide a method for resolving ambiguities about appropriate changes to the language. Where an ambiguity exists, a training instance can be generated which, when classified by the trainer [8] will resolve the ambiguity.

Given the relative strength of the analytical knowledge source described here, one may wonder whether the entire learning process could be based on this knowledge source, removing the need for inductive inference based on syntactic comparisons of training instances. From our initial case studies, it is clear that while an analytical knowledge source, such as KS_3, of the learner is an important component, the analysis is generally too difficult to be relied upon exclusively for inferring heuristics. For that reason, we are currently working on combining analytical and empirical methods [7] for concept learning. Thus, we believe that even when such knowledge sources are available to guide learning, empirical induction based upon bias will still be a necessary element, and that the kind of revision of bias discussed here will be necessary for effective inductive inference.

Acknowledgments

We thank N.S. Sridharan, Ranan Banerji, Saul Amarel, Robert L. Smith Jr., Rich Keller, and Pat Schooley for the many discussions which led to ideas presented here. We thank Alex Borgida, John Kastner, Van Kelly, and Peter Spool for helpful suggestions and reading of earlier drafts.

References

[1] Iba, G. A., ``Learning disjunctive concepts from examples,'' Master's thesis, M.I.T., 1979, also AI memo 548.

[2] Michalski, R. S., ``Pattern recognition as rule-guided inductive inference,'' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, 1980, pp. 349-361.

[3] Michalski, R. S. and Chilausky, R. L., ``Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis,'' *Policy Analysis and Information Systems*, Vol. 4, No. 2, June 1980, Special issue on knowledge acquisition and induction.

[4] Mitchell, T. M., *Version Spaces: An approach to concept learning*, Ph.D. dissertation, Stanford University, December 1978, also Stanford CS report STAN-CS-78-711, HPP-79-2.

[5] Mitchell, T. M., ``The need for biases in learning generalizations'', Technical Report CBM-TR-117, Department of Computer Science, Rutgers University, May 1980.

[6] Mitchell, T. M., Utgoff, P. E., Nudel, B. and Banerji, R., ``Learning problem-solving heuristics through practice,'' *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, August 1981, pp. 127-134.

[7] Mitchell, T. M., ``Toward Combining Empirical and Analytic Methods for Learning Heuristics,'' *Human and Artificial Intelligence*, Elithorn, A. and Banerji, R. (Eds.), Erlbaum, 1982.

[8] Sammut, C., *Learning Concepts by Performing Experiments*, Ph.D. dissertation, University of New South Wales, November 1981.

[9] Utgoff, P. E., ``Acquisition of Appropriate Bias for Inductive Concept Learning'', Thesis Proposal, Department of Computer Science, Rutgers University, May 1982.

[10] Vere, S. A., ``Multilevel counterfactuals for generalizations of relational concepts and productions,'' *Artificial Intelligence*, Vol. 14, No. 2, September 1980, pp. 138-164.

[11] Waterman, D. A., ``Generalization learning techniques for automating the learning of heuristics,'' *Artificial Intelligence*, Vol. 1, No. 1/2, 1970, pp. 121-170.