GENERALIZED RECURSIVE SPLITTING ALGORITHMS FOR LEARNING HYBRID CONCEPTS

Bruce Lambert
Speech Communication
University of Illinois
Urbana, Il 61801

David Tcheng
Computer Science
University of Illinois
Urbana, Il 61801
tcheng@m.cs.uiuc.edu

Stephen C-Y Lu
Mechanical Engineering
University of Illinois
Urbana, Il 61801

ABSTRACT

This paper describes the Competitive Relation Learner (CRL), a generalized recursive splitting algorithm capable of producing a wide range of hybrid concept representations through the competitive application of multiple learning strategies, multiple decomposition strategies, and multiple decomposition evaluation strategies. Experimental results are reported that demonstrate CRL's ability to outperform several well known fixed-bias strategies.

INTRODUCTION

Research on methods for learning concepts from examples occupies a central position in the discipline of machine learning (Michalski, Carbonell, & Mitchell, 1983). Among those who study the problem of learning from examples, it is now widely recognized that each concept description language and search strategy has an inherent *inductive bias,* that is, an extra-evidential preference for some hypotheses over others. Furthermore, no single inductive bias will yield optimal performance on all problems (Mitchell, 1980; Utgoff, 1986). Thus, a central concern of machine learning theorists is to discover methods for intelligently selecting the best inductive bias for a particular problem relative to a given set of user objectives. One way to address this problem is to develop inductive systems capable of producing hybrid concept representations which simultaneously capitalize on the strengths and minimize the weaknesses of two or more distinct inductive biases (Schlimmer, 1987; Utgoff, 1988). Utgoff's (1988) perceptron trees exemplify this approach by combining decision-trees with networks of linear threshold units.

However, the desire to design systems to learn continuous functions in the engineering domain (Lu & Chen, 1987) has motivated us to extend Utgoff's insight and to develop hybrid systems capable of learning continuous, real-to-real mappings. The inductive system we have developed generalizes the methodology of decision-tree building algorithms to include multiple problem decomposition strategies, multiple decomposition evaluation functions, and multiple learning strategies.

In what follows, we first suggest that well known decision-tree building algorithms like ID3 (Quinlan, 1986), PLS (Rendell, 1983), and CART (Breiman, Friedman, Olshen, & Stone, 1984) as well as recently developed hybrids like Utgoff's perceptron trees (1988) can all be viewed as more or less partial instantiations of an abstract class of algorithms we call *recursive splitting algorithms*. Second, we describe the Competitive Relation Learner (CRL), our implementation of a generalized recursive splitter. Finally, we present experimental results that compare the performance of CRL to several well known strategies.

GENERALIZED RECURSIVE SPLITTING ALGORITHMS

The principle factor motivating the design of CRL (aside from the need to learn piecewise continuous functions) was the observation that the behavior of a recursive splitting algorithm depends on three factors: (1) how candidate decompositions are generated; (2) how candidate decompositions are evaluated; and (3) how predictions are made within subregions (see also Breiman, *et al.*, 1984). Analysis of traditional recursive splitting algorithms reveals that each method possesses only one decomposition strategy, one decomposition evaluation function, and one learning strategy (i.e., a method for making predictions in each subregion). For example, ID3 (Quinlan, 1983) creates n-way splits

----

* This research was supported in part by the Applied Intelligent Systems Group of Digital Equipment Corporation and by the National Science Foundation (DMC-8657116).

on nominal feature dimensions, evaluates these splits with an entropy minimization function, and assigns the most frequently occurring class label to each subregion (i.e., leaf node). PLS creates binary splits perpendicular to scalar feature dimensions, chooses the decomposition that maximizes its dissimilarity metric, and attaches the mean utility to each subregion. Utgoff's (1988) novel contribution lay in realizing that performance could be improved by putting more powerful predictors (that is, learning strategies) at the leaf nodes. His perceptron trees first attempt to classify all instances with a single network of threshold logic units. Failing that, they impose n-way splits along nominal attribute dimensions, evaluate splits by entropy minimization, and insert perceptrons at the leaf nodes.

Clearly, any fixed combination of decomposition, evaluation, and prediction strategies may be ideal for a particular class of problems but will fail to provide optimal performance on others. The CRL system was designed to achieve more robust performance across problem domains. CRL is a general system for synthesizing and selecting hybrid concept representations. The current implementation contains four classes of learning strategies (mean, regression, neural net, and exemplar), three decomposition strategies (distance-based, population-based, and centroid-based), and a range of decomposition evaluation functions (for more detailed description of CRL's various strategies, see Tcheng, Lambert, Lu, & Rendell, in press). The design is modular, and new strategies can be added incrementally as long as they adhere to CRL standard input-output specifications.

THE CRL ALGORITHM

CRL is a straightforward generalization of the simple recursive splitting algorithm— the difference being that CRL evaluates multiple learning and decomposition strategy combinations in parallel. CRL begins with a single input space region containing every example and estimates the error in the region. The error of a region is determined by applying each active learning strategy to the examples and recording the error of the most accurate hypothesis.

Next, it must be determined whether or not further decomposition will reduce the overall hypothesis error. To do this, CRL applies all active decomposition strategies and evaluates the resulting candidate decompositions by computing the error of the resulting regions in the manner described above. The most valuable decomposition, the one that brings about the greatest overall error reduction, is used to create new subregions. This process is recursively applied to each subregion until one of the following three stopping criteria is met: (1) the error of the overall hypothesis ceases to decrease more than a specified threshold; (2) the number of examples in a candidate subregion falls below a specified threshold; or (3) the time consumed exceeds a specified threshold.

EXPERIMENTAL RESULTS

CRL's task in this example is to predict the surface roughness of a machined part based on the control parameters of the cutting tool. Examples were generated by a mechanistic simulator for the intermittent turning process (Zhang, 1989). The simulator mapped three input variables—feed rate, depth of cut, and revolutions per minute to one output value—surface roughness.

For this problem, the user's objective was defined in terms of hypothesis accuracy. Accuracy was measured by training CRL on 100 examples and testing on 100 different examples. For each trial, both training and testing examples were randomly selected. Hypothesis formation time was controlled by an CRL control parameter which placed an upper limit on the amount of CPU time that could be used to form any single hypothesis. For the results reported below, the time limit was 60 CPU seconds (on a SUN/3 180 with 24 Meg).

The best hypothesis produced by CRL took the form of a PLS-like decision tree with linear functions at the leaves and had an error of 145 (variance between predicted and actual SR). For comparison, several familiar biases were also tested on the same problem. The error of the best hypothesis of each of these strategies was as follows: averaging–2125, perceptron–855, linear regression–827, PLS–750, nearest-neighbor lookup–684. Strategies with parameterized bias received roughly equal optimization resources.

CONCLUSIONS

In this essay, we have stressed the representational advantages of generalized recursive splitting algorithms like CRL. The experimental results presented here suggest that a generalized recursive splitting algorithm can outperform most other methods on this problem (where the goal is to maximize predictive accuracy). We believe that CRL's performance advantage is a direct consequence of its ability to select from a diverse set of available strategies (via competition) that decomposition strategy, learning strategy, and decomposition evaluation function which results in the most accurate hypothesis.

The power of a generalized recursive splitting algorithm lies in the scope of its representational capabilities. With more than 100 tunable biases of its own, however, a system like CRL is difficult to use. Rather than viewing CRL as a stand-alone learning system, it is more usefully conceived as defining a huge inductive bias space which is then searched by an independent optimization system. Instead of specifying obscure CRL control parameters, the user would specify his performance objectives (e.g., hypothesis accuracy, formation time, evaluation time, comprehensibility) to the optimizer which would, it turn, select the biases for him. Finally, such a system ought to learn metarules for the optimizer which relate problem characteristics (e.g., number of examples, problem domain, number and type of features, and user objectives) to desirable regions of inductive bias space (Rendell, Seshu, & Tcheng, 1987).

Acknowledgements

Special thanks to Dr. Guangming Zhang for help in running the turning simulator and to Dr. Larry Rendell for valuable insight and help in organizing this paper. Thanks also to David Lambert and Dr. Barbara O'Keefe for helpful comments on an earlier draft of this paper.

References

Box, G., Hunter, W., and Hunter, J. (1978). *Statistics for experimenters*. New York, NY: John Wiley and Sons.

Breiman, L., Friedman, J., Olshen, R.A., and Stone, C. J. (1984). *Classification and regression trees.* Belmont, CA: Wadsworth.

Buchanan, T. (1986). Multiple objective mathematical programming: A review. *New Zealand Operational Research*, 14: 1, (pp. 1-27).

Lu, S. C-Y., and Chen, K. (1987). A machine learning approach to the automatic synthesis of mechanistic knowledge for engineering decision making. In *Journal of Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 1:2, (pp. 109-118)

Michalski, R., Carbonell, J., and Mitchell, T. (Eds.). (1983). *Machine learning: an artificial intelligence approach.* Palo Alto, CA: Tioga Publishing.

Mitchell, T. (1980). The need for bias in learning generalizations. Technical report CBM-TR-117. Dept. of Computer Science, Rutgers University.

Quinlan, R. (1983). Induction of decision trees. In *Machine Learning.* 1:1, (pp. 81-106).

Rendell, L. (1983). A new basis for state-space learning systems and a successful implementation. *Artificial Intelligence.* 20:4, (pp. 369-392).

Rendell, L., Seshu, R., Tcheng, D. (1987). Layered concept learning and dynamically-variable bias management. In *Proc. IJCAI 87.* (pp. 308-314). Cambridge, MA: Morgan Kaufmann.

Schlimmer, J. (1987). Learning and representation change. In *Proc. AAAI 87.* (pp. 511-515). Cambridge, MA: Morgan Kaufmann.

Tcheng, D., Lambert, B., Lu, S. C-Y., & Rendell, L. (in press). Building robust learning systems by combining optimization and induction. *In Proc. IJCAI '89.* San Mateo, CA: Morgan Kaufmann.

Utgoff, P. (1986). *Machine learning of inductive bias.* Dordrecht, Netherlands: Kluwer Academic Publishers.

Utgoff, P. (1988). Perceptron trees: a case study in hybrid concept representation. In *Proc. AAAI 88.* (pp. 601-606). San Mateo, CA: Morgan Kaufmann.

Zhang, G. (1989). Simulating the intermittent turning process. Unpublished manuscript. Knowledge Based Engineering Systems Research Lab. University of Illinois, Urbana, Il.

PROCEEDINGS OF THE SIXTH INTERNATIONAL WORKSHOP ON MACHINE LEARNING

Cornell University
Ithaca, New York

June 26–27, 1989

EDITOR/WORKSHOP CHAIR

Alberto Maria Segre, *Cornell University*

SPONSORS

Office of Naval Research, Artificial Intelligence and Robotics Program
Office of Naval Research, Cognitive Science Program
National Science Foundation, Knowledge Models and Cognitive Systems Program
Cornell University Department of Computer Science

MORGAN KAUFMANN PUBLISHERS, INC.
SAN MATEO, CALIFORNIA