ONE-SIDED ALGORITHMS FOR INTEGRATING EMPIRICAL AND EXPLANATION-BASED LEARNING

Wendy E. Sarrett and Michael J. Pazzani
Department of Information and Computer Science
University of California,
Irvine, CA 92717
714-856-4196
sarrett@ics.uci.edu, pazzani@ics.uci.edu

A FRAMEWORK FOR INTEGRATED LEARNING

The purpose of this paper is to describe a framework for integrating empirical learning with explanation-based learning (EBL)[DeJong & Mooney 1986; Mitchell, Keller & Kedar-Cabelli 1986] and to present an algorithm which does this with both pure conjunctive concepts and k-CNF concepts. Our framework involves using an empirical and an explanation-based method to form separate hypotheses and then combining the hypotheses from the separate sources to form a composite hypothesis. An additional important complication arises because the system is required to learn the domain theory (via an empirical method) at the same time it is using the domain theory to support the explanation-based method. The empirical methods that we use are one-sided algorithms that never generate a hypothesis that is more general than the correct hypothesis (assuming that the hypothesis can be represented in the hypothesis representation language). In addition, the empirical algorithms that we consider maintain a single hypothesis that is generalized as little as possible to accommodate positive examples.

The hypotheses produced by explanation-based learning with a domain theory acquired with such a one-sided empirical learning method will also never be more general than the correct hypothesis. Since both the empirical and explanation-based hypotheses are not more general than the correct hypothesis, they can be combined by finding the least general hypothesis consistent with both hypotheses. In this manner, the integrated hypothesis will be the least general hypothesis that is consistent with both the observed data and the domain knowledge. This hypothesis may be more general than either the empirical or explanation-based hypotheses. Some regularities may be ruled out because they are not consistent with the data. Other regularities may be ruled out because they are not consistent with the domain theory, although they may be supported by the data.

PERFORMANCE AND FOUNDATIONAL EXAMPLES

In order to understand the framework for integrated learning, it is important to note that there are two different kinds of training examples. *Performance examples* are training examples of the complete performance task. For example, imagine the task is for a small child to learn when other people will become angry. Performance examples here would be examples of people becoming angry when the child performs some action (e.g., breaking Daddy's watch). *Foundational examples* are training examples from which background domain knowledge can be learned. These differ from performance examples in that they can be viewed as a subproblem of the performance task. The performance task of predicting a way that a person will get angry can be broken into two subproblems, predicting what actions will cause an object to break, and predicting what sort of objects that become broken will anger what sort of people. More formally, assume the background knowledge is of the form: A and $X_{A,B} \rightarrow B$ B and $X_{B,C} \rightarrow C$ and we wish to acquire a predictive relationship: A and $X_{A,C} \rightarrow C$.

The goal is to learn the relationship "If you play with an expensive, glass object, the owner will become angry." The background knowledge might be acquired when a mother tells a child "Don't play with Daddy's watch; if you drop it, it will break and Daddy will get angry." Here, A represents dropping an object, B represents an object breaking, and C stands for a person getting angry. The term $X_{A,B}$ represents a number of unspecified conditions that restrict the class of objects that are broken when dropped (e.g., objects composed of glass), $X_{B,C}$ refers to additional conditions which are needed to determine what class of persons will be come angry when what class of objects breaks (e.g., the owner becomes angry when an expensive object breaks). These conditions are not specified in the domain theory; they must be acquired empirically from foundational examples. The goal of learning is to acquire $X_{A,C}$. This can be learned empirically from performance examples, or analytically ($X_{A,C} = X_{A,B}$ and $X_{B,C}$) from a domain theory acquired from foundational examples. In the next section, we give an algorithm that combines empirical and explanation-based methods in solving this problem.

THE IOSC and k-IOSCNF ALGORITHM

The first empirical learning strategy we consider is the Wholist strategy [Bruner, Goodnow & Austin 1956]. This strategy has also been called the One-Sided Algorithm for Pure Conjunctive Concepts [Haussler 1987]. Wholist works as follows: when a positive instance of the concept is seen but the current concept definition would classify it as a negative instance, the concept definition is redefined to be the intersection of the current concept definition and the instance. This process removes from the definition any features which are not in the instance and therefore can not be in the true definition of the concept. In Bruner's work, the initial hypothesis was a conjunction of all features in the first positive example. Here, we initialize the hypothesis to be the conjunction of all features in the example description language.

The first algorithm we will present is the integrated One-Sided Algorithm for Pure Conjunctive Concepts (IOSC). In IOSC, the empirical algorithm is used for two purposes. First, it is used as the only learning algorithm to acquire the domain theory from foundational examples. Second, it is used to form one hypothesis for the performance concept from the performance examples. Explanation-based learning produces a second hypothesis for the performance example. These hypotheses are combined to form IOSC's hypothesis. When there is no domain theory, IOSC is equivalent to Wholist. In this manner, IOSC can be used on both performance and foundational examples. For simple conjunctive concepts, EBL merely finds the conjunction of the features in the antecedents of the rules in the domain theory. Since the regularities present in the performance examples may differ from the regularities present in the foundational example, the hypotheses produced by empirical and explanation-based means will differ. In IOSC, there are two reasons that are sufficient for dropping a feature from a hypothesis: either the feature was not present in all positive performance examples, or the feature is not needed to explain why a performance example is an instance of the performance concept.

Input:

- H - the current hypothesis of the concept definition to be learned (current concept definition – initialized to the entire list of features).
- B - the current background theory
- E - a training example (instance of either the goal concept (performance example) or background concepts (foundational example).
- *Member*? - Boolean value indicating whether or not E is a positive example.

Output: An updated concept definition.

The algorithm:

- iosc($H, B, E, Member?$)
  1. C = *Classify*(H, E) (this classifies the instance as positive or negative based on the current concept definition).
  2. If (C = *negative* and $Member? = True$) Then
     (a) If domain-theory-explains(E,B) THEN (if you can use EBL)
        i. $H1$ = "remove features from H which are not in E";
        ii. $H2$ = "create hypothesis with EBL";
        iii. "update H by removing those features of $H1$ which are not in $H2$."
     (b) ELSE "update H by removing features from H which are not in E."

We have extended this algorithm to the case of k-CNF by reformulating the definition of a feature. In IOSC, features are simply the surface features of the training examples and hypotheses are conjunctions of these features. For k-CNF, composite features can be constructed that are disjunctions of the surface features of length k or less. Hypotheses in k-IOSCNF are conjunctions of these composite features. Instances are defined as a list of Boolean variable settings (e.g., L = 1, M = 0, etc.) which satisfy the true definition of the concept (either goal-concept or background concept) being learned with this instance. The empirical component for k-CNF from Valiant (1984) is used instead of Wholist.

Both Wholist and Valiant's k-CNF algorithm have two important properties that enable the combination of the empirical and analytic hypothesis. First, both algorithms are one-sided (i.e., the hypothesis is never more general than the true concept definition). Thus, the hypothesis formed by EBL with such a domain theory, learned by these algorithms, will also never be more general than the true concept definition. Second, the hypothesis representation language is closed under conjunction. Therefore, EBL will produce a hypothesis in the same representation language as the empirical component, permitting the two hypothesis to be combined.

CONCLUSION

In this paper we have discussed a framework for learning k-CNF by combining empirical learning with EBL. This algorithm can easily be extended to include truth maintenance. Finally, although IOSC is limited in applicability by its constrained representation language, k-CNF expressions are powerful enough to be used in describing application domains such as medical diagnoses.

References

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A Study of Thinking*, New York: Wiley.

DeJong, G., & Mooney, R. (1986). Explanation-Based Learning: An Alternative View. *Machine Learning*, 1(2), pp. 145-176.

Haussler, D. (1987). *Applying Valiant's Learning Framework To AI Concept Learning Problems* (Technical Report No. UCSC-CRL-87- 11). Santa Cruz: University of California. Department of Computer Science.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1), pp. 47-80.

Valiant, L. G. (1984). A Theory of the Learnable. *Communications of the ACM*, 27(11), pp. 1134-1142.

Valiant, L. G. (1985). Learning Disjunctions of Conjunctions. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 560- 566. Los Angeles, CA: Morgan Kaufmann.