Learning From Ambiguous Examples

Stephen V. Kowalski
University of Southern California
Department of Electrical Engineering Systems
EEB 232, MC 2562
Los Angeles, California 90089-2562
skowalsk@pollux.usc.edu

Current inductive learning systems are not well suited to learning from ambiguous examples. We say that an example is ambiguous if it has multiple interpretations, only one of which may be valid. Some domains in which ambiguous learning problems can be found are natural language processing (NLP) and computer vision. An example of an ambiguous training instance with two interpretations is shown below, where $\oplus$ is the Exclusive-OR function and each interpretation is a conjunction of attribute values.

$$\begin{aligned} E_1 &= i_1 \oplus i_2 \\ i_1 &= [(\text{cat} = \text{verb}) \wedge (\text{agree} = \text{n3sg})] \\ i_2 &= [(\text{cat} = \text{noun}) \wedge (\text{num} = \text{singular})] \end{aligned}$$

Our first thought is to transform this example into disjunctive normal form (DNF). Each conjunction would then become a new example described in a representation that can be understood by most existing inductive learners. There are several problems with this approach, two of which are described below.

First, there would be a combinatorial explosion in the number of training examples and thus the complexity of the learning algorithm. A second problem arises from the introduction of negated attribute values[1] in the training instances which some learners (e.g. ID3 (Quinlan 1986)) are ill-equipped to handle. We also note that an ambiguous example may take multiple paths down a decision tree during classification. These paths may terminate at leaf nodes that are labeled with different classes.

A system that could learn directly from ambiguous examples would broaden the use of inductive learning in the previously mentioned domains.

One problem in NLP is sentence classification. Each word in a sentence has multiple interpretations corresponding to different dictionary meanings. For example, E_1 might be a representation for the word *plant*. A sentence could then be represented as a list of these expressions, one for each word.

Our approach to learning from ambiguous examples is to represent the training instances and the hypotheses with the same language. This language, at its highest level, employs a form of regular expression to match patterns of text. This expression consists of an ordered list of *items* which are either Kleene stars or expressions that consist of an exclusive disjunction[2] of interpretations. Each interpretation is a conjunction of attribute values, and internal disjunctions are permitted (e.g. category = noun $\vee$ verb). An example hypothesis is shown below.

$$\begin{aligned} H_1 &= \{*, t_1\} \\ t_1 &= [(\text{cat} = \text{noun}) \wedge (\text{num} = \text{singular})] \oplus \\ &\quad [(\text{cat} = \text{adverb})] \end{aligned}$$

This hypothesis, H_1, would match any sentence where the last word can be interpreted as either a singular noun or an adverb.

Our inductive learning algorithm performs a beam search from general to more specific expressions and relies heavily on a formal definition of subsumption. Since we are using the same language for the training examples and the hypotheses that are learned, we say that a hypothesis *matches* a training example if the expression for the hypothesis *subsumes* the expression for the example. A subsumption operator is a binary Boolean operator that takes as arguments two expressions, and evaluates true if the expression on the left is more general or has the same generality as the expression on the right, and false otherwise.

Experiments have been conducted in one of the domains of the Message Understanding Conference (MUC). Text sentences were classified by the type of information that they contained, and a dictionary-based pre-processor was used to generate the ambiguous representation. The learning algorithm was shown to successfully learn concepts that could be used to form information extraction rules. Our current work is on integrating the learning system with a performance element which is a rule-based information extraction system.

References

Quinlan, J. R. 1986. Induction of Decision Trees. *Machine Learning* 1:81–106.

[1] $a \oplus b = (a \wedge \neg b) \vee (\neg a \wedge b)$.

[2] An exclusive disjunction is of the form $a \oplus b \oplus c$.