Graph Based Concept Learning

Jesus A. Gonzalez, Lawrence B. Holder, Diane J. Cook

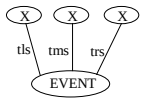
University of Texas at Arlington, Department of Computer Science and Engineering
Box 19015, Arlington, TX 76019-0015
{gonzalez,holder,cook}@cse.uta.edu
URL: http://cygnus.uta.edu/subdue/ConceptLearning

Concept Learning is a Machine Learning technique in which the learning process is driven by providing positive and negative examples to the learner. From those examples, the learner builds a hypothesis (concept) that describes the positive examples and excludes the negative examples. Inductive Logic Programming (ILP) systems have successfully been used as concept learners. Examples of those are Foil (Quinlan and Cameron 1993) and Progol (Muggleton 1995). The main engine of these systems is based in first order logic. In this research we introduce a graph based relational concept learning system called SubdueCL, which through the experiments has shown that it is competitive with ILP systems in different types of domains. SubdueCL is an extension made to the Subdue (Cook and Holder 1994) system, which is an unsupervised graph based learner.

The Subdue system takes as input a labeled graph and discovers substructures (sub-graphs) that compress the input graph, according to the minimum description length principle and represent structural concepts in the data. The main discovery algorithm is a computationally constrained beam search. The algorithm begins with the substructure matching a single vertex in the graph. Each iteration the algorithm selects the best substructure and incrementally expands the instances of the substructure. The algorithm searches for the best substructure until all possible substructures have been considered or the total amount of computation exceeds a given limit. Evaluation of each substructure is determined by how well the substructure compresses the description length of the input graph. The best substructure found by Subdue can be used to compress the input graph, which can then be input to another iteration of Subdue. After several iterations, Subdue builds a hierarchical description of the input data where later substructures are defined in terms of substructures discovered on previous iterations. Subdue has been applied to several domains including image analysis, CAD circuit analysis, chemical reaction chains, and artificially-generated databases (Cook and Holder 1994).

The SubdueCL system is an extension to Subdue. It uses Subdue's core functions to perform graph operations, but the learning process is different because it works as a supervised learner by differentiating positive and negative examples using a set-covering approach. The hypothesis found by SubdueCL is a disjunction of conjunctions. SubdueCL forms one of these conjunctions (rules) in each iteration. Positive example graphs covered in a previous iteration are removed from the graph for subsequent iterations. SubdueCL evaluates the generated substructures (or rules) according to how well they describe the positive examples without describing the negative examples. For this evaluation, the positive examples that are not covered and the negative examples covered by the substructure are considered errors because the ideal substructure would be one covering all the positive examples without covering any negative example. SubdueCL allows hypotheses where some of the rules may cover negative examples. We are working on a version of SubdueCL that produces only consistent hypotheses, and the application of the PAC learning framework to SubdueCL.

A comparison of SubdueCL with the two ILP systems Foil and Progol using attribute-value databases showed that SubdueCL performs better for non-numeric domains and Progol for numeric domains. The goal now is to show how SubdueCL performs with relational domains where ILP systems have produced very good results. Some preliminary results using this type of domains show that SubdueCL is able to learn the tic-tac-toe domain and produce a perfect output just as Foil and Progol do. The figure shows a rule learned in the tic-tac-toe domain that describes a winning board with X's in the top line (tls=top-left-side, tms=top-medium-side and trs=top-right-side).



References

Cook, D. J. and Holder, L. B. 1994. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research* 1:231-255.

Muggleton, S. 1995. Inverse entailment and Progol. *New Generation Computing* 13:245-286.

Quinlan, J. R. and Cameron-Jones, R. M. 1993. FOIL: A Midterm Report. *In Proceedings of the European Conference on Machine Learning*. Vienna, Austria 3-20.