Declarative bias in ATRIS rule induction shell

Dunja Mladenić
J. Stefan Institute
Artificial Intelligence Laboratory
Jamova 39, 61000 Ljubljana, Slovenia
Phone: (+386)(61) 1259 199, Fax: (+386)(61) 1258 058
E-mail: Dunja.Mladenic@ijs.si

Abstract

The paper presents ATRIS rule induction shell in the light of its declarative biases. ATRIS was developed to enable simple use of different combinatorial optimisation algorithms and noise-handling mechanisms, to enable simple "cross validation" experiments, experiments with different attributes considered as class and different attribute subsets used for learning. ATRIS currently incorporates seven local optimisation algorithms and four heuristic estimates, enables use of simple built-in background knowledge, use of pruning parameters and validation of the induced hypothesis.

1 Introduction

The paper presents inductive concept learner that learns in propositional logic. ATRIS examples are described by attribute value tuples and induced knowledge is represented by a language of *if-then* rules. Learning is performed inside the covering paradigm where for a single rule induction, local combinatorial optimisation algorithm is used. ATRIS currently incorporates stochastic local optimisation algorithms (based on *Simulated annealing* [6]) and deterministic local optimisation algorithms (*k-opt*, known from solving the "travelling salesman" problem [7]). Stochastic mechanism used at the beginning of each rule induction enables ATRIS to escape from the unpromising parts of the search space. The whole search process is guided by a heuristic function. There are four different heuristic functions that can be used in ATRIS for the evaluation of the expected classification accuracy.

In Section 2 ATRIS is described in the light of each of its declarative bias. Section 3 concludes with summary and some ideas for the further work on declarative bias in ATRIS.

2 Declarative bias in ATRIS

The input to ATRIS is domain described in FUN format [4] that enables multidimensional visualisation of domain data and, representation of control knowledge in a declarative way. In a domain definition, experiments are described by means of training and testing examples, class and attributes, value types and declarative biases. Next three sections describe three types of declarative bias as implemented in ATRIS shell.

2.1 Language bias

Language bias is the least flexible bias in the ATRIS rule induction shell. ATRIS learns in propositional logic. Its hypothesis space is defined as a set of *if-then* rules. Rules are of the form **if** *Complex* **then** *Class*, where *Complex* is a conjunction of selectors and *Selector* relates an attribute to a value or a disjunction (so-called *internal disjunction*) of values. Using the idea from the LINUS relation learning system [2], ATRIS enables use of simple built-in background knowledge. Eg. equality: generate new attribute from pair of attributes of the same or compatible type using equality of attribute values.

2.2 Search bias

Inductive learning can be defined as search of the space of concept descriptions guided by some heuristic function. ATRIS perform this search by some local combinatorial optimisation algorithm. It currently incorporates 7 local combinatorial optimisation algorithms; two stochastic and five deterministic. Both stochastic algorithms are some sort of *Simulated annealing* [6]. The method of *Simulated annealing* is based on the analogy with thermodynamics; (for slowly cooled systems, nature is able to find the minimum energy state). For a selected cost function [5] which is to be minimised, the method performs stochastic search in a discrete state space. All five incorporated deterministic algorithms are variants of *k-opt* algorithm [7]. In general, *k-opt* algorithm have some sort of k-size window inside which deterministic search is performed.

ATRIS learning process can be divided into two phases: construction of a hypothesis and its post-processing. For the first phase, each of the seven optimisation algorithms can be selected. For the second phase, non or one of the five deterministic optimisation algorithms can be selected. Selection is made by user in the domain definition. When combining two optimisation algorithms one should be aware of their characteristics. Stochastic algorithms have good performances on difficult problems with hilly search space where their ability of escaping from local minima is beneficial. Deterministic algorithms more or less systematically search neighbourhood, always moving to a better position and thus can be used for "fine-tuning" of the current solution. In our experiments [3] the best results were achieved by the combinations of stochastic algorithm (learning phase) and deterministic algorithm (post-processing phase).

Each local combinatorial optimisation algorithm uses some cost function to guide the search. ATRIS incorporates 4 heuristic estimates that are used as *cost functions* in the above algorithms. The cost function we defined with the quality (the classification accuracy) of a rule and can be calculated by several probability estimates, the *relative frequency* being the most common one. In the both learning phases ATRIS uses the same heuristic estimate i.e. one of the four currently implemented heuristics selected by the user.

2.3 Validation bias

ATRIS learns inside the covering paradigm. The covering algorithm induces set of rules for each class separately. Covering algorithm loop is controlled by *stopping criterion*, used to decide when to stop the induction of rules for the current class. This criterion can be referred to as the *sufficiency criterion*. For the validation of sufficiency of the current class hypothesis, ATRIS uses parameter given by the user that represents allowed percentage of uncovered positive examples. The other well known *stopping criterion*, usually referred to as the *necessity criterion* can be included in the ATRIS heuristic estimates and applied during the induction and post-processing of each rule. There is additional validation bias applied on each rule at the end of its construction that is similar to the informativity pre-pruning parameter in the Assistant algorithm [1]. This bias is given by the user in the form of the lowest bound of an accepted rule quality. The main difference between informativity pre-pruning parameter in Assistant and the rule quality used by ATRIS is that Assistant stops subtree construction while ATRIS just rejects currently induced rule and trys to induce a better one. Stochastic mechanism used at the beginning of rule induction makes it possible for ATRIS.

Validation of the induced hypothesis ATRIS performs by calculating classification accuracy on training, testing, or all (training and testing) examples [1]. Example set for this validation is specified by the user in the domain definition.

3 Conclusion

ATRIS rule induction shell enables user to represent several control information in a declarative way. The first type of declarative bias, language bias, can be just slightly influenced by the user through the use of simple built-in background knowledge. The hypothesis language is fixed and defined with a set of *if-then* rules. The second, search bias, is the one user have the greatest influence to. In experiment design, user can select search algorithm for learning and the other for post-processing. Each search algorithm uses some heuristic to guide the search. Selection of this heuristic is also left to the user. The third type of declarative bias, validation bias, we divided into two categories. One used to validate induced hypothesis for a specific class during the learning process and the other used to validate induced hypothesis at the end of the learning process.

The first category is influenced by the parameter used for stopping the hypothesis induction and the parameter used to accept/reject a single rule. The second category is influenced only by the user's selection of the example set for validation (training, testing or all examples).

At the moment ATRIS declarative biases can be shifted by the user (ML expert). Shifting declarative bias by the system is also interesting, specially when the learning problem is specified as "goal driven" with different, possibly combined goals. One of the most direct definition of the learning goal is maximisation of classification accuracy of the induced hypothesis. We plan to investigate this possibility inside the ATRIS and/or some other inductive learner.

References

[1] Cestnik, B., Kononenko, I. and Bratko, I. (1987) ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In Bratko, I. and Lavrač, N. (eds.) *Progress in machine learning*, pp. 31-45. Wilmslow, Sigma Press.

[2] Lavrač, N., Džeroski, S. and Grobelnik, M. (1991) Learning nonrecursive definitions of relations with LINUS. *Proc. of EWSL 91*, pp. 265-281. Porto, Portugal, Springer-Verlag.

[3] Mladenić, D. (1993) Combinatorial optimization in inductive concept learning. In *Machine Learning Proceedings of the Tenth International Conference*, (P.E.Utgoff, ed.), pp. 205–211. Morgan Kaufmann Pubishers, San Mateo, CA.

[4] Mladenić, D. (1993) FUN: UNiversal domain Formalism (In Slovene), Technical report, J. Stefan Institute, *IJS DP-6845*, Ljubljana, 1993.

[5] Papadimitriou, C.H. and Steiglitz, K. (1982) Optimization problems. In *Combinatorial optimization: Algorithms and complexity*, pp. 1-24. Prentice-Hall, Inc. Englewood Cliffs

[6] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1987) Combinatorial minimization: Method of simulated annealing. In *Numerical Recipes: The art of scientific computing*, pp. 326-334. Cambridge University Press.

[7] Syslo, M.M., Deo, N. and Kowalik, J.S. (1983) Local search heuristics. In *Discrete optimization algorithms*, pp. 368-380. Prentice-Hall. Inc. Englewood Cliffs