Using phenotypic sharing in a classifier tool

Mozart Hasse

Departamento de Informática
Universidade Federal do Paraná
Curitiba, Brasil
mozart.hasse@usa.net

Aurora R. Pozo

Departamento de Informática
Universidade Federal do Paraná
Curitiba, Brasil
aurora@inf.ufpr.br

Abstract

This paper describes a classifier tool that uses a genetic algorithm to make rule induction. The genetic algorithm uses the Michigan approach, is domain independent and is able to process continuous and discrete attributes. Some optimizations include the use of phenotypic sharing (with linear complexity) to direct the search. The results of accuracy are compared with other 33 algorithms in 32 datasets. The difference of accuracy is not statistically significant at the 10% level when compared with the best of the other 33 algorithms. The implementation allows the configuration of many parameters, and intends to be improved with the inclusion of new operators.

1 OVERVIEW

This paper describes a classifier tool that uses a genetic algorithm to make rule induction. The genetic algorithm uses the Michigan [Holland, 1986] approach, is domain-independent and can process discrete and continuous attributes.

The tool works as follows. One independent genetic algorithm is created for every possible class. Each population of rules is based only on positive and negative instances, and all of them are used to create a rule set. A simple heuristic joins the rules sorted by importance.

The genetic algorithm folows the basic format in [Mitchell 97]. It evolves a population of fixed-length rules, and each rule is a set of conjuntions in the form r_1 ^ r_2 ^ ... ^ $r_n \Rightarrow C$ where n is the number of attributes, r_i is the constraint for values in attribute A_i, i in [1,n], and C is the predicted class. Any constraint of a rule can be empty, indicating that the corresponding attribute can assume any value.

The fitness function for this problem must be able to qualify the rules as *partial* classifiers, so the accuracy of a rule is more important than its ability to cover all training instances.

The genetic algorithm uses a sharing scheme to force the creation of subpopulations of rules. Sharing is based on the reduction of the fitness of an individual in a proportion to the presence of similar individuals in the population. Similarity between rules can be measured in genotypic or phenotypic space. Phenotypic sharing allows a more useful comparision between similarity, because the measure is taken from the number of common examples covered by the rules. The sharing measure for each rule is based on the positive training instances correctly classified by the whole population.

2 RESULTS AND CONCLUSIONS

In order to allow comparisions, the same methodology of [Lim et al, 1999] was used. In that paper, 33 classification algorithms were compared in 32 different datasets in terms of classification accuracy, running time and number of rules. The tool has been tested on all 32 datasets, obtaining a mean error rate of 0.2303. This result is not statistically significant (at the 10% level) from the best of the other 33 classifiers. The classifier tool did not get the worst result in any dataset. Moreover, its accuracy was very good (better or very close to the best) in 7 datasets.

Phenotypic Sharing performed very well and improved accuracy significantly. The results obtained until now show that the algorithm possibly can achieve better results with the inclusion of new genetic operators and other improvements.

References

Holland, J. H. (1986). *Escaping Brittleness : The possibilities of general purpose learning algorithms applied to parallel rule-based systems.* in R. Michalski, J. Carbonell & T. Mitchell (Eds.), *Machine Learning : an AI Approach, vol II.* Morgan Kaufmann, Los Altos, CA, pp. 593-623.

T.-S. Lim, W.-Y. Loh and Y.-S. Shih (1999). *A comparision of prediction accuracy, complexity and training time of 33 old and new classification algorithms.* in Machine Learning Journal, Kluwer Academic Publishers, Boston.

M. Mitchell. (1997) *An introduction to genetic algorithms.* Cambridge. MIT Press. 207p.