Evolution of adaptive discretization intervals for rule-based genetic learning system

Jaume Bacardit
Enginyeria i Arquitectura La Salle (URL)
Psg. Bonanova 8, 08022-Barcelona,
Catalonia, Spain, Europe.
jbacardit@salleURL.edu

Josep M. Garrell
Enginyeria i Arquitectura La Salle (URL)
Psg. Bonanova 8, 08022-Barcelona,
Catalonia, Spain, Europe.
josepmg@salleURL.edu

Abstract

The traditional classifier rules evolved in the genetic classifier systems need a discretization process to handle well problems with real attributes. A good discretization is needed to generate a solution with good accuracy because the alternative of a high number of simple uniform-width intervals is bad due to the big search space being hardly explorable in a reasonable time.

The work being summarized here deals with a rule representation with adaptive discrete intervals which split or merge through the evolution process, finding the correct discretization intervals as the learning process is done with a reasonable computational cost.

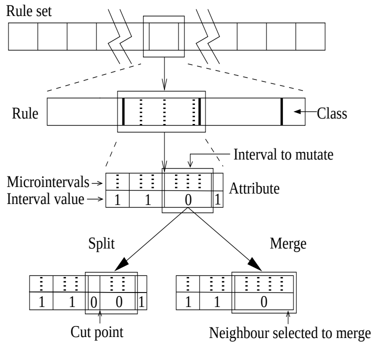
Summary

This representation is used inside a Pittsburgh approach genetic classifier system derived from GABIL [DeJong et al., 1993], and the rule structure is taken directly from GABIL, using Conjunctive Normal Form (CNF) predicates.

To make this representation have a reasonable computational cost and to control well the search space, it has to follow the next constraints:

- We define a fixed number of "low level" intervals which we call *micro-intervals*.
- The adaptive intervals are built joining *micro-intervals*.
- When we split an interval, we select a random point in its *micro-intervals* to do the process.

Figure 1: Diagram of the adaptive intervals representation and the split and merge operations.



- When we merge two intervals, the value of the resulting interval is taken from the one which has more *micro-intervals*. If both have the same number, the value is chosen randomly.

The representations and split and merge operations are represented in figure 1

To make the interval splitting and merging part of the evolution, we have to make it part of one of the recombination phases of the GA. The easiest part of the GA which can handle well the new operators is the mutation. So, we redefine the mutation operator with p_{split} and p_{merge} as the probabilities of splitting or merging an intervals selected to mutate, instead of the classic bit-inversion operator.

References

[DeJong et al., 1993] DeJong, K. A., Spears, W. M., and Gordon, D. F. (1993). Using genetic algorithms for concept learning. *Machine Learning*, 13(2/3):161–188.