Improving the Discovery Component of Classifier Systems by the application of Estimation of Distribution Algorithms

Joaquin Rivera , Roberto Santana*
* Center of Mathematics and Theoretical Physics, ICIMAF,
Calle 15 e/ C y D, Vedado, Havana, Cuba
Rsantana@cidet.icmf.inf.cu

Since Holland and Reitman [2] published their CS-1 a great amount of studies concerning classifiers systems (CSs) have appeared. The needs of studying crucial matters in the functioning of CS determined that researchers paid little attention to the evolutionary component of these systems. Nevertheless in the last years GA's have experienced a considerable development that makes necessary to revise an update current applications of EA's to CSs. The goal of our current research is to incorporate to the CS's machinery new Evolutionary Techniques that make of them more powerful machine learning tools. In this paper we introduce an Estimation Distribution Algorithm to perform as the discovery component of the classifier system XCS.

Estimation of Distribution Algorithms [1][6][5.5] are based on probability theory and statistics. They are population based optimization methods that use selection. Instead of applying genetic operators to the selected population these algorithm estimate probability distribution of individuals and use this information to generate new points. EDA have shown to be superior [5.5] to classical GA's in the optimization of a wide set of functions. The XCS classifier system[8] has been successful in solving different learning problems. It has deserved a great attention in recent years and settled theoretical basis for the study of generalizations. For the interest reader it may be important to revise [9], [4] and [5] for recent developments concerning XCS.

The rules in XCS store three main parameters : prediction p_j, prediction error ε_j, and fitness F_j. The prediction parameter means the reward the rule expects to gain. The prediction error is an estimate of the error present in the prediction of the rule. The fitness gives a measure of how precise is the prediction of the rule, and it is a function based on the prediction error.

Every time an input signal is sent by the environment to the system, a match set [M] is formed with the rules in the population [P] whose condition part match the input signal. In case the match set [M] is empty a new rule that matches the input is created through an operator called *covering*. Then a system prediction $P(a_i)$ is computed for every action a_i in [M]. This system prediction is used to select the action to be carried out by the system. The selection can be made stochastically or deterministic. Once the selection is made, an action set [A] is formed with the rules in [M] advocating the selected action a_i (rules with its action part equal to a_i). The selected action (system action) is then performed. As a result the environment sends a new input signal together with an immediate reward (r_{imm}). This reward is used to adjust the parameters in [A]-1, the action set of the previous time step. In the XCS the adjustments are made using a Q-learning-like technique that have some similarities with bucket-brigade.

The discovery component in XCS (GAs, usually) is performed in niches (action sets). It generates new rules, based on the ones present in the niche. Every time an offspring is going to be inserted in the population, the rules in the niche are checked out to see if any rule's condition contains the condition of the new generated rule. If so, the new rule is not inserted, instead a numerosity counter of the old rule is increased. This technique is known as subsumption, and using it permits the insertion of only those rules in [P] which are more general than the existing ones. The rules numerosity 'at born' is one. The rules with numerosity greater than one are known as macroclassifiers. The combined use of macroclassifiers together with the conception of fitness, result in rules mapping accurately the environment, and in niches having one dominating rule.

The environment employed in this paper is an animat one, the woods2. This environment is the one used in [8] and [9] to test the performance of the XCS. The binary decoding of blank B is 000, of tree T is 010, of predator animat A is 011, of food F is 100, and food G decodes 110. The animat in woods2 is supposed to learn to avoid trees and predators, and reach foods. It also must reach food at a 1.7 step average. The initial position of the animat is a random blank cell. The GAs are the evolutionary algorithms always employed in classifier systems. This paper is the first attempt of applying another EA to these systems.

Estimation of Distribution Algorithm for the XCS:

STEP 0 : Build the population of individuals to evolve from the rules present in [A]-1.

STEP 1 : Select M<N individuals using the truncation selection method. Estimate the distribution of the rules in the selected set using the 3 order marginal factorization of the distribution.
STEP 2 : Generate K new rules according to the distribution estimated.

Different environment would need different factorizations of the distribution. In the Animat environment the marginals are calculated for the three-alleles blocks. In the woods2 there are 27 possible blocks, with four of them illegal (111, 11#, 001, and 110). We obtain a mutation like effect by making the illegal marginals zero, then dividing the desired mutation rate between the legal three-alleles blocks that are not present in the niche, and sharing the remaining probability (1-mutation rate) between the three-alleles blocks present in the niche proportional to their marginals. This will result in a kind of heuristic mutation for the EDA.

In the XCS the prediction of the offspring is usually the mean of the parents predictions. In our EDA implementation this genetic concept of parent does not exists. What the Estimation Distribution Algorithms do is to utilize the information about the alleles distribution of the best individuals to generate the offspring. We introduce a novel technique[7] that allows a more sophisticated and exact offspring predictions calculation, that can accelerate the desired accurate mapping of the offspring, improving the system performance.

In the aim of comparing this two evolutionary algorithms, and their influence in the performance of the XCS new measures able to give a detailed idea of the system's dynamics were defined. There were more than 12 different measures of performance which are described in [7]. Statistics mentioned here demonstrate that learning is faster when the EDA method is employed.

For both algorithm we counted the average of:

- Adjustments needed to : become a macroclassifier for first time.
- Adjustments needed to : kj equals one.
- Adjustments needed to : error equals zero.
- Absolute difference of every offspring prediction to the predictions of its two parents (GA).
- Absolute difference of every offspring prediction to the predictions of every rule in pool (EDA).

Results of the experiments show that rules generated by the EDA become adjusted more times than those generated by GA's. The superiority is in the range of 20% to 40% of the average number of rules created for both algorithms. Moreover, rules created with EDA need a less number of adjustments in order to make its error equal to zero. This fact means that learning is faster. The absolute difference between the prediction values of descendant and parents is higher in EDA, this could be explained by the way we estimate the prediction of new created rules, this estimation seems to be more accurately that the simple average of both parents prediction used by traditional GA.

This paper reports preliminary results of a Classifier Systems that uses an EDA as its discovery component. We have introduced a new method to estimate the prediction of descendant, and explained measures that help to a deeply understanding of the XCS performance. Further work includes the analysis of other classifier systems for more complex environments that would need discovery components able to exploit in a greater measure particular characteristics of the environment.

References

[1] Baluja, S. & Caruana, R. (1995). Removing the Genetics from the Standard Genetic Algorithm. Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, CA.

[2] Holland, J. H. & Rietman, J. S. (1978). Cognitive systems based on adaptive algorithms.

[3] Kovacs, T. (1996). Evolving optimal populations with XCS classifier systems. Technical Report CSR-96-17 and CSRP-96-17, School of Computer Science, University of Birmingham, Birmingham, U. K.

[4] Lanzy, P. L. (1997). Solving Problems in Partially Observable Environments with Classifier Systems (Experiments on Adding Memory to XCS). Technical Report N. 97.45.

[5] Lanzy, P. L. (1997). A Model of the Environment to Avoid Local Learning (An Analysis of the Generalization Mechanism of XCS). Technical Report N. 97.46

[5.5] Mühlenbein, H., Mahnig, T. Och

[6] Pelikan, M.; Goldber D. E., Cantú-Paz E.(1998). Linkage problem, Distribution estimation and Bayesian Estimation Networks. Illegal Report 98013. November 1998

[7] Rivera J.; Santana R. (1999) The discovery component of the XCS Classifier System. Diploma thesis in preparation (In Spanish)., Faculty of Mathematics, Havana University.

[8] Wilson, S. W. (1995). Classifier fitness based on accuracy. Evolutionary Computacion, 3 (2)

[9] Wilson, S. W. (1998). Generalization in the XCS classifier system. From Genetic Programming 1998 : Proceedings of the Third Annual Conference, J. Koza et al. (Eds.). San Francisco, CA : Morgan Kaufmann.