A Further Look at UCS Classifier System

Albert Orriols-Puig
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull
Quatre Camins, 2. 08022 Barcelona, Spain.
aorriols@salleurl.edu

Ester Bernadó-Mansilla
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull
Quatre Camins, 2. 08022 Barcelona, Spain.
esterb@salleurl.edu

ABSTRACT

This paper studies UCS, a learning classifier system (LCS) derived from XCS that works under a supervised learning scheme. A complete description of the system is given. Besides, we introduce a fitness sharing scheme to UCS and analyze UCS both with fitness sharing and without fitness sharing. Results show the benefits of fitness sharing in all the tested problems, specially those with class imbalances. We also compare UCS with XCS and analyze how the fitness pressure works under each approach.

Categories and Subject Descriptors

I.2.6 [**Learning**]: concept learning, knowledge adquisition

General Terms

Experimentation

Keywords

Evolutionary Computation, Genetic Algorithms, Machine Learning, Learning Classifier Systems, Fitness Sharing

1. INTRODUCTION

UCS [1] is a learning classifier system (LCS) derived from XCS [10, 11] that works under a supervised learning scheme. UCS inherits the main components and structure of XCS, which are adapted for supervised learning. The main differences between both systems are related to 1) classifier's parameters and their update, and to 2) the lack of a prediction array in UCS. UCS's fitness is based on accuracy, computed as the percentage of correct classifications. This makes UCS to explore the consistently correct classifiers and thus evolve only best action maps.

In previous work [1], UCS's lack of fitness sharing was identified as a potential weakness. Thus, the aim of this paper is to introduce a fitness sharing scheme and analyze its benefits compared to raw UCS. We also compare both UCS with and without fitness sharing with XCS. The testbed consists of two binary-input classification problems selected from a larger set (see [7]) that highlight the differences between both systems: the decoder and the imbalanced multiplexer problems. We show that XCS suffers from fitness dilemma [3] in the decoder problem, whose effect is a misleading pressure that tends to guide the genetic search in the wrong direction. The way in which accuracy is computed allows UCS to overcome the problem.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

2. DESCRIPTION OF UCS

2.1 UCS Components

UCS is an accuracy-based learning classifier system introduced in [1]. It inherits the features of XCS, but specialize them for supervised learning tasks. UCS mainly differs from XCS in two perspectives. Firstly, the performance component is adjusted to a supervised learning scheme. As the class is provided with each new example, UCS only explores the correct class. This implies that UCS only evolves high-rewarded classifiers, that is, the best action map [B]. Secondly, accuracy is computed differently in both systems. UCS computes accuracy as the percentage of correct classifications instead of computing it from the prediction error.

In the following, we give a deeper insight into UCS by explaining each component of the system.

Classifier's Parameters

In UCS, classifier's parameters are the following: a) accuracy acc; b) fitness F; c) correct set size cs; d) numerosity num; and e) experience exp. Accuracy and fitness are a measure of the quality of the classifier. The correct set size is the estimated average size of all the correct sets where the classifier participates. Numerosity is the number of copies of the classifier, and experience is the number of times that a classifier has belonged to a match set.

Performance Component

UCS is an online learner that receives a new input example $x = (x_1, ..., x_n)$ at each learning iteration. As it works under a supervised learning scheme, also the class c of the example is provided. Then, the system creates a match set [M] that contains all classifiers in the population [P] whose condition matches x. From that, the correct set [C] is created, which consists of the classifiers in [M] that predict the correct class. If [C] is empty, the covering operator is activated, creating a new classifier with a generalized condition matching x,

and predicting class c. The remaining classifiers form the incorrect set ![C].

In test mode, a new input example x is provided, and UCS must predict the associated class. To do that, the match set [M] is created. All classifiers in [M] emit a vote, weighted by their fitness, for the class they predict. The most-voted class is chosen. Under test mode, the population of UCS does not suffer any change. All update and search mechanisms are disabled.

Parameter Updates

Each time a classifier participates in a match set, its experience, accuracy and fitness are updated. Firstly, the experience is increased. Then, the accuracy is computed as the percentage of correct classifications:

$$acc = \frac{number\ of\ correct\ classifications}{experience} \quad (1)$$

Thus, accuracy is a cumulative average of correct classifications over all matches of the classifier. Next, fitness is updated according to the following formula:

$$F = (acc)^{\nu} \quad (2)$$

where ν is a constant set by the user (a common value is 10). Thus, fitness is calculated individually for each microclassifier, and it is not shared. The fitness of the whole macroclassifier is: $F_{macro} = num * F$.

Finally, each time the classifier participates in [C], the correct set size cs is updated. cs is computed as the arithmetic average of all sizes of the correct sets in which the classifier has taken part.

Discovery Component

The genetic algorithm (GA) is used as the primary search mechanism to discover new promising rules. The GA is applied to [C], following the same procedure as in XCS. It selects two parents from [C] with a probability that depends on classifier's fitness. The same selection schemes applied in XCS can be used in UCS, such as proportional selection or tournament selection. The two parents are copied, creating two new children, which are recombined and mutated with probabilities χ and μ respectively.

Finally, both children are introduced into the population. First, each offspring is checked for subsumption with its parents (the subsumption mechanism is adapted from XCS). If the offspring cannot be subsumed, it is inserted in the population, deleting another classifier if the population is full. The deletion probability is computed in the same way as in XCS (see [5]).

Parameter Initialization

UCS is very robust to parameter initialization since the initial value of most of the parameters is lost the first time that the classifier participates in a match set. When a classifier is created by covering, its parameters are set to: $exp = 1$, $num = 1$, $cs = 1$, $acc = 1$ and $F = 1$. If a classifier is created by the GA, its parameters are initialized to: $exp = 0$, $num = 1$, $cs = (cs_{p_1} + cs_{p_2})/2$ (where p_1 and p_2 denote each of the parents), $acc = 1$ and $F = 1$.

2.2 Why do not share fitness?

We introduce a new fitness computation scheme that shares fitness, similarly to XCS, with the aim of comparing its advantages and disadvantages with a non sharing scheme. In the remainder of the paper, UCS without sharing is referred as UCSns, and UCS with sharing as UCSs.

Parameters update with fitness sharing works as follows. Experience, correct set size and accuracy are computed as in UCSns. However, fitness is shared among all classifiers in [M]. Firstly, a new accuracy k is calculated, which discriminates between accurate and inaccurate classifiers. For classifiers belonging to ![C], $k_{cl\in![C]} = 0$. For classifiers belonging to [C], k is computed as follows:

$$k_{cl\in[C]} = \begin{cases} 1 & if \quad acc > acc_0 \\ \\ \alpha(acc/acc_0)^{\nu} & otherwise \end{cases}$$

Then, a relative accuracy k' is calculated:

$$k'_{cl} = \frac{k_{cl} \cdot num_{cl}}{\sum_{cl_i\in[M]} k_{cl_i} \cdot num_{cl_i}} \quad (3)$$

And fitness is updated from k':

$$F = F + \beta \cdot (k' - F) \quad (4)$$

Let's note that, under this scheme, the computed fitness corresponds to the macroclassifier's fitness, as numerosities are involved in the formulas.

3. XCS AND UCS IN BINARY-INPUT PROBLEMS

3.1 Methodology

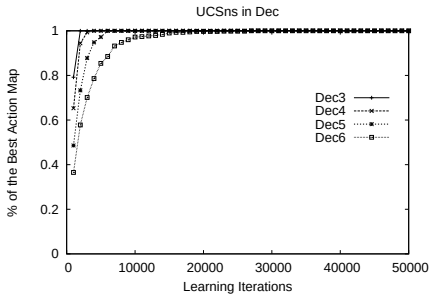
The aim of this section is to analyze the behavior in different facets of UCS and XCS. We base our analysis on two artificial problems that gather some complexity factors said to affect the performance of LCSs [1, 6]: a) the decoder [1], a multiclass problem; and b) the imbalanced multiplexed [8], an imbalanced binary-class problem.

The proportion of the optimal action map achieved was proposed as an accurate measure of the progress of the genetic search in XCS [6]. However, XCS and UCS evolve different types of action maps. To permit a fair comparison between both systems, we used the proportion of the best action map achieved %[B] as the metric of performance.

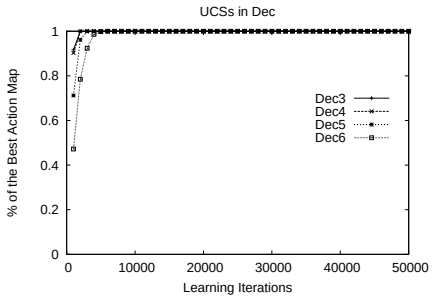
3.2 Results

We ran XCS, UCSns and UCSs with the decoder and the 11-bit imbalanced multiplexer. Parameters were set as follows. For XCS, we set: $\beta = 0.2$, $\alpha = 0.1$, $\nu = 5$, $\theta_{GA} = 25$, $selection$ = tournament, $\chi = 0.8$, $\mu = 0.04$, $\theta_{del} = 20$, $\delta = 0.1$, GA_{sub} = true, $[A]_{sub}$ = false, $\theta_{sub} = 20$. Parameters for UCS had the same values as in XCS, with acc_0=0.999 and ν=10. Population size was set to $N = 25 \cdot |[O]|$ in XCS, and to $N = 25 \cdot |[B]|$ in UCS[1]. Besides, in the 11-bit imbalanced multiplexer problem, parameters of both XCS and UCS were tuned following the guidelines proposed in [8]. For XCS we set β = {0.04, 0.02, 0.01, 0.005} and θ_{GA} = {200, 400, 800, 1600} for i={6, 7, 8, 9} respectively. UCS appeared to be less sensitive to parameters' settings in previous experiments (not reported here). We only set θ_{GA} = 50 and $\beta = 0.02$ for i≥6.

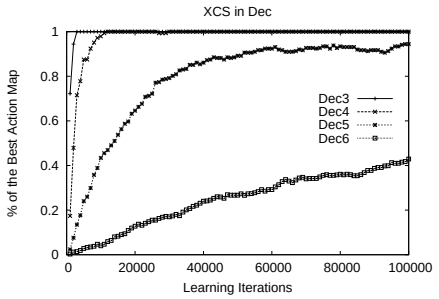
[1]Optimal population sizes $|[O]|$ in XCS and UCS are different since XCS evolves the complete action map and UCS only evolves the best action map.



(a) UCSns

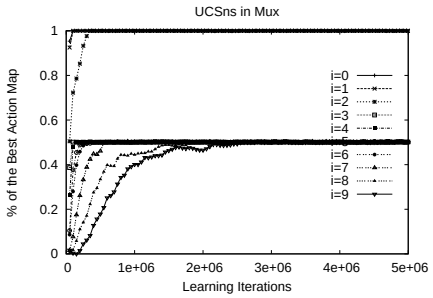


(b) UCSs

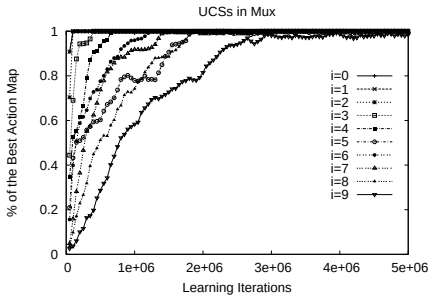


(c) XCS

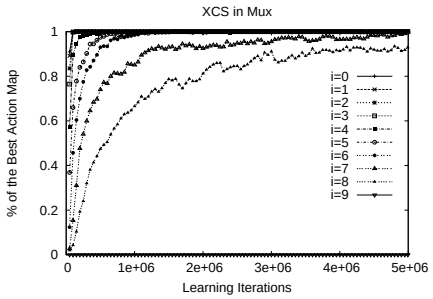
Figure 1: Proportion of the best action map achieved by UCSns (a), UCSs (b) and XCS (c) in the decoder problem with condition lengths from l=3 to l=6. Note that UCS is shown for 50,000 explore trials, while XCS is shown for 100,000 trials.



(a) UCSns



(b) UCSs



(c) XCS

Figure 2: Proportion of the best action map achieved by UCSns (a), UCSs (b) and XCS (c) in the 11-bit multiplexer problem with imbalance levels from i=0 to i=9.

Figure 1 shows the proportion of the best action map achieved by UCSns, UCSs and XCS in the decoder problem with condition length from 3 to 6. Figure 2 shows the same metric for the 11-bit imbalanced multiplexer problem with imbalance levels from i=0 to i=9. All curves were averaged over ten runs with different seeds. The differences observed between the three systems are summed up in the following.

Explore Regime

Exploring only the class of the input instance (as UCS does) is beneficial in the decoder problem. In general, such an explore regime is advantageous in problems with high number of classes. Moreover, it helps to solve the imbalanced multiplexer up to one imbalance level higher than XCS, in an extreme low supply of minority class instances.

Accuracy Guidance

The results of XCS in the decoder problem shows the lack of fitness guidance toward accurate classifiers. This problem, already observed in previous studies [1, 3], was termed as the fitness dilemma in [3]. The problem does not exist in UCS since accuracy is computed directly as the percentage of correct classifications. The results show that XCS strongly suffers from fitness dilemma in the decoder. In this

case, UCS clearly outperforms XCS. To alleviate this effect, bilateral accuracy was proposed for XCS [3]. As a future work, we aim to investigate how this approach compares with UCS.

Fitness Sharing

Fitness sharing speeds up the convergence in all problems tested. Specially, it appears to be crucial in highly imbalanced datasets to deter overgeneral classifiers from overtaking the population.

Figure 2(a) shows that, without fitness sharing, UCS fails at solving the imbalanced multiplexer problem for high imbalance levels. For i$\geq$4, UCSns evolves only half of the best action maps. Looking at the populations evolved (not shown for brevity) we observed that UCSns discovered all optimal classifiers predicting the minority class (class 1). However, optimal classifiers predicting the majority class were replaced by the most overgeneral classifier (with all the bits set to '#') covering the majority class. This behavior is related to 1) the low difference in fitness of the most overgeneral rule an the maximally general and accurate rules predicting the majority class when both coexist in the population; and 2) the increasing number of genetic opportunities that the most overgeneral classifier receives respect to the maximally general ones.

Population Size

In the tested problems, UCS evolved best action maps with less learning iterations. Also smaller population sizes were used in UCS in all the tested problems. The population evolved by XCS is generally larger, but comparable to that of UCS in terms of legibility. In fact, by removing low-rewarded classifiers from XCS's final population, we get a set of rules similar to that of UCS (not shown for brevity). Thus, the advantage derived from having smaller populations in UCS is that we need less computational resources to solve the problem.

4. CONCLUSIONS

This paper provided a brief analysis of the UCS learning classifier system (see [7] for more details). We improved the original UCS system as introduced in [1] by including fitness sharing. A fitness sharing scheme appeared to be beneficial in both problems tested. Specially, it was crucial in the imbalanced multiplexer problem. Using sharing, we allow overgeneral classifiers until optimal classifiers start to evolve. When this happens, fitness of overgeneral classifiers decreases fast by the effect of sharing fitness with better competing solutions. We suspect that this behavior can be also generalizable to other imbalanced problems, where overgeneral classifier can easily become strong.

Comparison with XCS allowed for better understanding of the differences between two approaches of accuracy-based classifier systems. There were two key differences between XCS and UCS that provided UCS with better results in the classification domains tested: exploration focused on best action maps and correct fitness pressure toward accuracy. XCS's convergence could be improved by using search regimes with more exploitation guidance. Some methods such as those based on ϵ-greedy action-selection or softmax action-selection [9] have already been tested on reinforcement learners. Their introduction in XCS could lead to similar performance to UCS. To avoid the effects of fitness dilemma in XCS, the use of bilateral accuracy was proposed [3].

Acknowledgements

The authors thank the support of *Enginyeria i Arquitectura La Salle*, Ramon Llull University, as well as the support of *Ministerio de Ciencia y Tecnología* under project TIN2005-08386-C05-04, and *Generalitat de Catalunya* under Grants 2005FI-00252 and 2005SGR-00302.

5. REFERENCES

[1] Bernadó-Mansilla, E. and Garrell, J.M. Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks. *Evolutionary Computation*, 11(3):209–238, 2003.

[2] Bull, L. and Hurst, J. ZCS Redux. *Evolutionary Computation*, 10(2):185–205, 2002.

[3] Butz, M.V., Goldberg, D., and Tharankunnel, K. Analysis and improvement of fitness exploration in XCS: bounding models, tournament selection, and bilateral accuracy. *Evolutionary Computation*, 11(3):239–277, 2003.

[4] Harik, G. Finding Multiple Solutions in Problems of Bounded Difficulty. Technical report, IlliGAL Report No. 94002, Urbana-Champaign IL 61801, USA, May 1994.

[5] Kovacs, T. Deletion Schemes for Classifier Systems. In W. Banzhaf, J. Daida, A. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, (GECCO-99)*, pages 329–336. Morgan Kaufmann, 1999.

[6] Kovacs, T. and Kerber, M. What makes a problem hard for XCS. In *Lanzi, P. L., Stolzmann, W., & Wilson, S. W. (Eds.), Advances in Learning Classifier Systems: Third International Workshop, IWLCS*, pages 80–99. Springer-Verlag, 2000.

[7] Orriols-Puig, A. and Bernadó-Mansilla, E. A Further Look at UCS Classifier System. Technical report, Enginyeria i Arquitectura La Salle - Ramon Llull University. http://www.salle.url.edu/∼aorriols, January 2006.

[8] Orriols-Puig, A. and Bernadó-Mansilla, E. Bounding XCS Parameters for Unbalanced Datasets. In *2006 Genetic and Evolutionary Computation Conference*, 2006 (accepted).

[9] Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998.

[10] Wilson, S.W. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

[11] Wilson, S.W. Generalization in the XCS Classifier System. In J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H. Iba, and R. Riolo, editors, *Genetic Programming: Proceedings of the Third Annual Conference*, pages 665–674. Morgan Kaufmann, 1998.