MENTORING: AN INTELLIGENTLY BIASED SELECTION POLICY FOR COLLECTIVE LEARNING AUTOMATA

ALICE J. ARMSTRONG
The George Washington University
Department of Computer Science
Washington, DC

PETER BOCK
The George Washington University
Department of Computer Science
Washington, DC

ABSTRACT

Mentoring is a new selection policy for statistical learning systems that has been initially tested with a Collective Learning Automaton that solves a simple, but representative problem. To respond to an immature stimulus that does not yet have a high-confidence response associated with it, current selection policies usually select their response randomly. Albeit unbiased, this policy ignores any confident information already acquired for *other* well-trained stimuli. To exploit this confident information, mentoring hypothesizes that *in the absence of a sufficiently confident response to a given stimulus*, *selecting a confident response to a different, but nonetheless well-trained stimulus is a better strategy than selecting a random response*. Mentoring does not require distance metrics to find "similar" stimuli (feature vectors) in search of an appropriate response. Preliminary results show that mentoring significantly accelerates learning and reduces error, especially when several stimuli share the same response, *i.e.*, when broad domain generalization is possible. Although mentoring does not provide a significant performance improvement when such generalization is not possible, neither does it incur significant costs.

INTRODUCTION

In Collective Learning Systems (CLS), a Collective Learning Automaton (CLA) learns the appropriate response for each stimulus by selecting responses until one of them emerges as statistically optimal, guided by feedback from an evaluating Environment (Bock 1976). Currently, CLS theory ignores what has already been learned by other stimuli when making decisions about a new stimulus. However, once some reliable knowledge is available for one situation, it seems reasonable to incorporate it into learning the solutions for other situations, even if they are largely unrelated. Many psychologists agree that applying successful solutions for old problems to new and often unrelated problems is a useful learning strategy (Piaget, 1977 [1933], Pulaski, 1980, Berk, 2003). Although the experiments reported in this paper do not attempt to replicate human behavior at any level, biologically and psychologically inspired mechanisms and methods can often provide useful insights and hints for AI methods (Heckman, 2004).

The research reported in this paper proposes a new selection policy for CLAs, called **mentoring**, which applies the knowledge about one well-learned situation to other situations without comparing their feature vectors to accelerate learning. Although many machine learning algorithms can achieve excellent results by identifying similar feature vectors (explicit domain generalization), they all require postulating a sensible and computable distance metric. For example, the *k*-Nearest Neighbor algorithm (Mitchell, 1997; Moore & Lee, 1994) computes similarity using the Euclidean distance between vectors in an ordered *n*-dimensional space. On the other hand, although case-based reasoning (Sycara *et al.*, 1992) allows feature vectors to be categorical, a distance metric of some kind must be postulated to identify similar cases. For many problem domains, however, it is simply not possible to postulate a meaningful distance metric. For example, in Natural Language Processing there is no direct way to compute the distance between the meanings of words so other methods to compute similarity must be devised (Portnoy, 2005). The **mentoring** method reported in this paper, however, does not compare feature factors at all, and is thus applicable to a wide problem domain.

BACKGROUND

To understand mentoring in the context of Collective Learning Systems, a brief introduction to this statistical learning paradigm may be helpful. In a Collective Learning System (CLS) each agent is a Collective Learning Automaton (**CLA**) that learns how to respond to stimuli appropriately using the algedonic cycle (Beer, 1966), as illustrated in Fig. 1. The CLA is embedded in an Environment that sends a stream of stimuli to the CLA and periodically issues evaluations of the CLA's responses to these stimuli. A **stimulus** is a vector of several features that describes some state of the Environment. The CLA uses a State Transition Matrix (**STM**) to store each unique stimuli that has been received, along with its occurrence count (sample size) and an estimate of the probability that each possible response is valid for this stimulus. For each stimulus that is received, the CLA uses these probabilities to select a **response**, which is then sent to the Environment. These selection probabilities are updated based on periodic evaluations issued to the CLA by the Environment at the end of a **stage**, which is a sequence of responses by the CLA.

In current CLS theory, for a given stimulus the **Standard CLA** selects the response with the highest statistical confidence, if the confidence is sufficiently high; otherwise, a response is selected at random. All responses are sent to the Environment, and at the end of each stage, the Environment evaluates their collective performance. This **evaluation** is issued to the CLA, where the **compensation** function converts the evaluation into an **update**. The update is applied to all the elements of the probability vectors in the STM that were used to generate the CLA's responses since the last evaluation (the **history** of the stage). (Bock 1993)

In modern CLS engines Bayes' Rule and the standard difference of two proportions are often used to compute the statistical confidence of each response for every stimulus, which is called the **selection confidence** of a response.

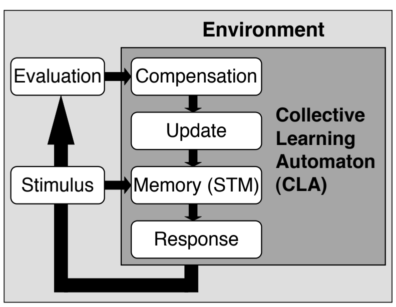

Figure 1: Algedonic cycle of a CLS

PROPOSED SOLUTION

Mentoring is an algorithm that can override the **standard selection policy** used by a Standard CLA with the objective of improving learning performance. A **Mentored CLA** follows the standard selection policy until one stimulus is sufficiently well trained to become eligible as a **`Mentor`**. A stimulus becomes a `Mentor` when its selection

ic cycle of a CLS

confidence is very high (98% for these experiments). A Mentor identifies the response it would select for its stimulus, and mentored stimuli (**Students**) simply use this response (**mentored response**), assuming it is better than a random response. However, each Student tracks the effectiveness of the Mentor's response and uses it only as long as it remains effective. When a new Mentor becomes available, all stimuli that do not yet have an effective Mentor will try out the new Mentor.

The lifecycle of a hypothetical stimulus in a Mentored CLA is described in Fig. 2. When there are no Mentors in a CLA, all stimuli follow the standard selection policy and are called **standard stimuli**. As soon as the first Mentor appears, all standard stimuli will investigate it. When a stimulus selects a Mentor, it becomes a Student of that Mentor. As long as a Mentor remains effective for a Student, the Student will continue to use the Mentor's responses. However, if a Mentor proves ineffective after some time (a parameter), the Student **drops** this Mentor, ceases to use the mentored response, and looks for another Mentor. If no other effective Mentors are available, the stimulus reverts to the standard selection policy and becomes a **freelance stimulus**. After a Student has attained a specified selection confidence (90% for these experiments), the Student becomes an **Independent** and reverts to the standard selection policy. Dropping the Mentor allows the Independent to explore its response range. Exploration is useful because it is possible that there is an even better response for the Independent than the mentored response. An Independent will either lose confidence in its response and revert to being a Student (or a freelance stimulus), or will become confident enough to become a Mentor itself. An Independent is allowed some latitude while exploring its own response range, and it will only revert to being a Student if its selection confidence falls below a threshold (75% for these experiments).

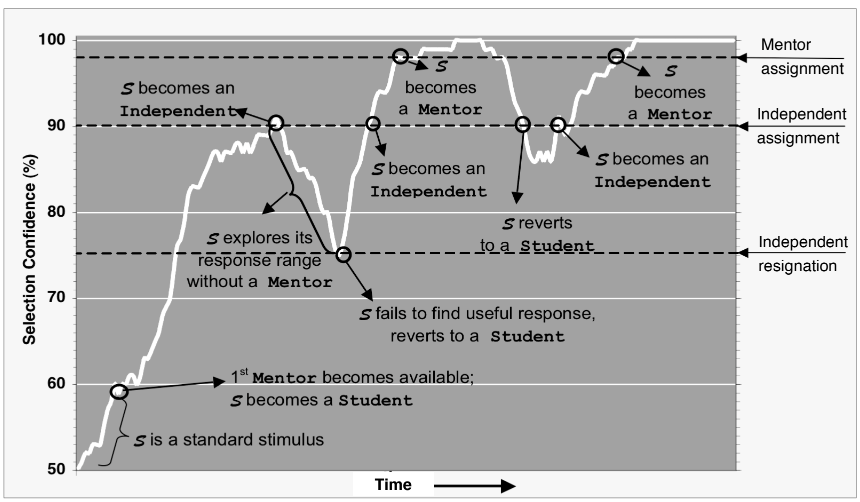

Figure 2: Lifecycle of a hypothetical stimulus *S* in a Mentored CLA

In the event that a Mentor loses confidence in its response (this is common in the early stages of a CLA's learning), the Mentor **resigns** and notifies all of its Students. All Students using this Mentor remove it from their list of possible Mentors, and any new stimuli received by the CLA will not be assigned to the resigned Mentor. A resigned `Mentor` reverts to being a `Student` or freelance stimulus and, as such, will itself seek a `Mentor`. Note that the resigned `Mentor` may become a `Mentor` again if it regains sufficient confidence in a response

It is important to note that mentoring relies solely on the response selection confidence and not on any analysis of the feature vectors of the stimuli or attributes of the responses. This avoids some of the thorny problems associated with devising reasonable similarity and distance metrics. All mentoring requires are unique identifiers for all stimuli and all responses. Mentoring relies on the pigeonhole principle, which says that if there are more stimuli than responses, then some stimuli must share the same response.

Another major advantage of mentoring is that it reduces the interference of the dissociative player: the random number generator that must be used to make random choices. However, the use of a random number generator is not eliminated entirely. Random decisions must still be made in several situations: when there are no `Mentors`, when a stimulus has become `Independent`, and when there are freelance stimuli. However, when there are many `Students`, fewer random choices are being made, and the interference of the dissociative player is minimized.

EXPERIMENT DESIGN

For this paper, CLAs were trained and tested on simple function-learning problems. The experiments included two cases: Categorical functions and Ordered functions. In the Categorical case, no inherent order among the available responses was assumed. This case can be thought of as a classification problem with no direct way to order the classes (*e.g.,* class 1 = tanks, class 2 = small rocks, class 3 = shopping malls, class 4 = spaniels, *etc.*), and the evaluation issued to the CLA by the Environment had only two values: correct or incorrect. In the Ordered case, however, the evaluation issued to the CLA by the Environment was the integer distance between the selected responses and the correct response, yielding a multi-state evaluation. It is important to consider these two cases separately, because they represent two very different, important, and common classification problems.

Figure 3 shows the functions used in these experiments. The first two rows show selections from the Categorical functions, and the second two rows show selections from the Ordered functions. In the Categorical case, the effect of the distribution of correct responses in the range was investigated. Two piece-wise linear functions were tested which shared the same number of correct responses, but whose responses were either distributed approximately as a sigmoid (**Smooth1**) or as a straight line (**Smooth2**). In the Smooth1 function, as many stimuli as possible shared the same correct response. In the Smooth2 function, the correct responses were distributed evenly among the stimuli.

In the Ordered case, the effect of two different distributions of the correct responses was investigated. In both functions, the correct responses were distributed evenly among the stimuli, but the distance between the elements of the sets of the correct responses was varied. In the **Rough** function, correct responses were placed as far apart as possible from each other. In the **Smooth2** function the distance between responses was minimized.

Each function had a discrete range and domain of 40 values. All functions varied from being one-to-one ($f(x) = x$) with forty correct responses, to a constant ($f(x) = 20$) with only one correct response. Each of the 40 possible discrete instantiations of the function is called a **step**, labeled from $f(x) = 20$ as step 1 to $f(x) = x$ as step 40.

All experiment trials were run as follows: a Mentored CLA and a Standard CLA were each given the same random stimulus, and the response of each was immediately evaluated. This CLS stage (stimulus → response → evaluation) was defined as a **contest.**

The CLAs were all trained for 40,000 contests. During training, learning was turned off every 400 contests, and the CLAs were tested. During these **test samples**, the CLAs were presented with all 40 possible stimuli, which were repeated 6 times to acquire reasonable estimates of the averages and standard deviations of performance during the test sample.

The matches of 40,000 contests and their periodic test samples were Monte Carlo'd 30 times using 30 different random-number-generator seeds to facilitate formal statistical analysis under controlled conditions.

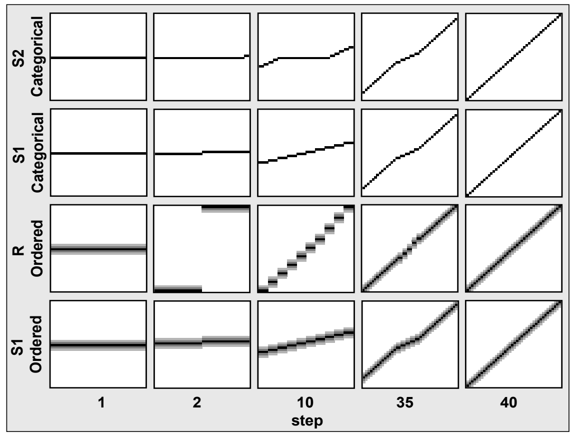

Figure 3: Steps of the categorical functions (S2 = Smooth2 in the first row and S1 = Smooth1 in the second row) and the ordered functions (R = Rough in the third row and S1 = Smooth1 in the last row). Each function transitions from $f(x) = 20$ to $f(x) = x$ in 40 steps. The step number is the number of correct responses. Stimuli lie along the horizontal axes; responses lie along the vertical axes.

RESULTS

The results of these experiments are reported in three sections: Categorical results, Ordered results, and a comparison of the performance with the different kinds of functions (Smooth 1, Smooth2, and Rough). A few samples from Categorical and Ordered results are presented in Fig. 4 and 5. It is important to note that first and last steps of the Categorical and Ordered functions are exactly the same (see Fig. 3). Additionally, the Categorical functions Smooth1 and Smooth2 are exactly the same from step 20 to 40.

Categorical Results

For the Categorical functions, the CLA either selected the correct response or not, so performance was measured as the average percentage of correct selections made during each test sample. The standard difference of two proportions was used to compute the single-tailed confidences that the performance of the Mentored CLAs exceeded the performance of the Standard CLAs. Figure 4 shows a selection of these Categorical results. Step 1 presented the most difficult learning task for both CLAs, but the Mentored CLA's performance improved very quickly starting around contest 20,000, and the difference in performance is statistically significant thereafter. Both CLAs learned step 40 very quickly and there is almost no significant difference in their performances. At step 10, the Mentored CLAs' performance is significantly better than the Standard CLAs' on both the Smooth1 and the Smooth2 functions. It is interesting to note that performance of the Mentored CLAs is almost exactly the same for both functions.

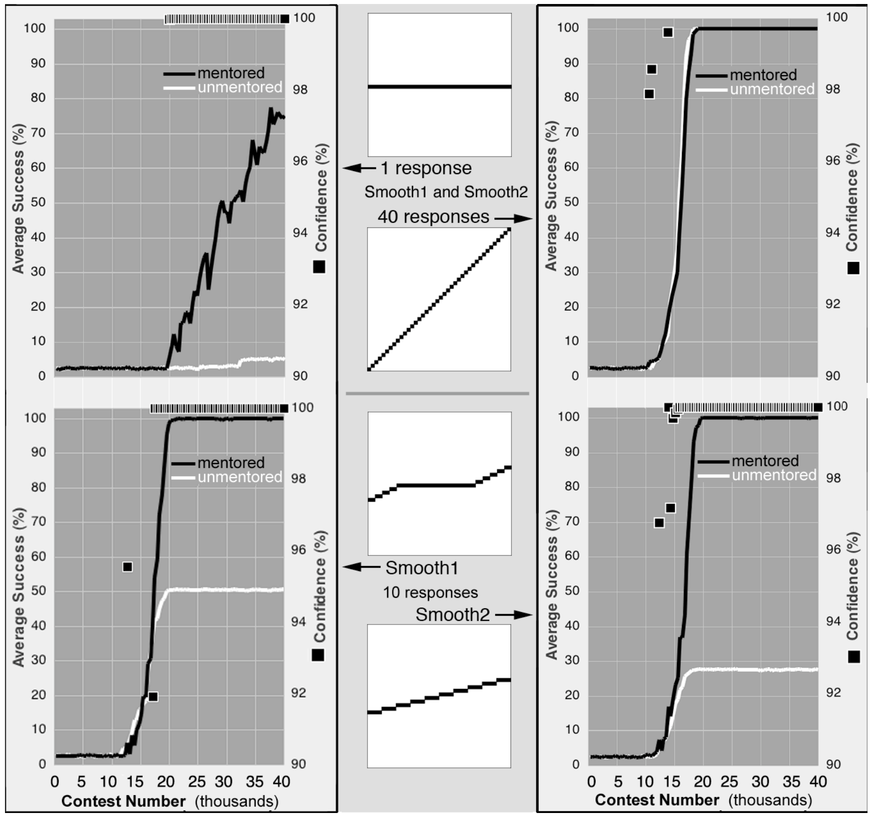

Figure 4: Categorical Results. Performance (left-hand axes) and statistical confidences (right-hand axes) that a Mentored CLA yields better performance than a Standard CLA. The charts in the center of the figure specify four examples of the functions to be learned by the CLAs, with stimuli along the horizontal axes and responses along the vertical axes.

Ordered Results

To learn Ordered functions, the evaluation issued by the Environment was the absolute distance from the CLA's response to the correct response, *i.e.*, the absolute error. For these experiments, an error less than or equal to 2 was considered acceptable. An error of zero was rewarded most highly, an error of 1 less highly, and an error of 2 was rewarded even less highly. As the error became greater than 2, a CLA was given negative evaluations that scaled with the error. For some steps of the Rough and Smooth2 functions it was possible to receive harsher punishments than in others because there was a greater possible error range. For example, if the correct answer was response 1 and the CLA chose response 40, then the maximum possible error was 39. On the other hand, if the correct response was 20, then the maximum error was only 20.

In the Ordered case, the performance metric is error, so a lower score means better performance. Note that the results for step 1 and step 40 of the Rough and Smooth2 functions are exactly the same (Fig. 5). For the Ordered results, the standard difference of two proportions was used to compute the single-tailed confidence that the difference of the two results for each of the test samples in the match was statistically significant.

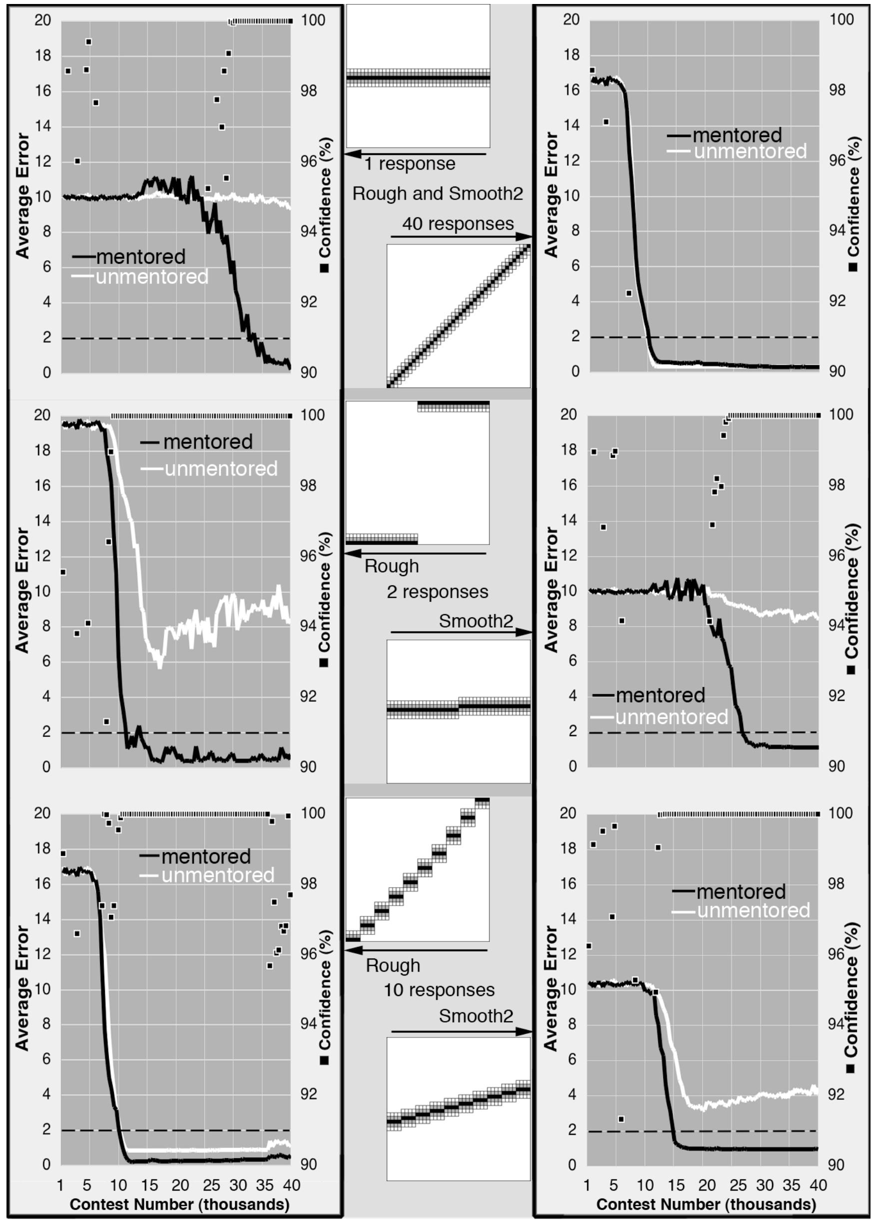

Figure 5: Ordered Results. Performance (left-hand axes) and statistical confidences (right-hand axes) that a Mentored CLA yields better performance (lower error) that a Standard CLA. The charts in the center of the figure specify six examples of the functions to be learned by the CLAs. Stimuli lie on the horizontal axes and responses on the vertical axes.

Figure 5 shows results for selected steps. The results for steps 1 and 40 are very similar to those in the Categorical case. In step 1, the Standard CLA makes almost no improvement in performance, but at about halfway through the match, the Mentored CLA's performance improves dramatically. By the end of the match, the Mentored CLA is performing with in the range of acceptable error. In step 40, there is little difference in performance between the two CLAs. Both learn very quickly and both reach near zero error by contest 30,000.

Step 2 shows the biggest disparity between the results of the two functions. In the Rough function, the Mentored CLA's performance is significantly and consistently better than the Standard CLA's after about contest 12,000. In the Smooth2 function, the results look very similar to those in step 1. This difference in performance on the two functions suggests that the dispersion in the domain has a significant effect on the learning process in both Mentored and Standard CLAs.

Step 10 also shows some of the effect that the dispersion in the domain has on learning and mentoring. In the Rough function, both the Mentored and Standard CLA learn at approximately the same rate, but the Mentored CLA's performance is still statistically different than that of the Standard CLA. In the Smooth2 function, the Standard CLA is still struggling to learn the function. Its error never goes below the threshold of 2. Again, the Mentored CLA quickly learns the function and outperforms the Standard CLA.

Comparison of Performance with Different Functions

In each match, there were 100 test samples where learning was turned off and the CLA was tested. The 30 results from the 30 matches were averaged, and the confidence that the mentored performance was better than the unmentored performance was used as a metric for assessing overall performance. Figures 6 and 7 show all forty steps on the horizontal axis and the average confidence on the vertical axis.

In an attempt to summarize the change in performance over all 40 steps of the Categorical and Ordered functions, the confidences that the Mentored CLA outperformed the Unmentored CLA were averaged over the test samples in which the most rapid changes in performance occurred. In both classes of functions and over most of the steps in each function, Mentored and Unmentored CLAs did not make any significant improvement in performance before the first 40 test samples were completed, therefore these results are not considered in Fig. 6 and 7.

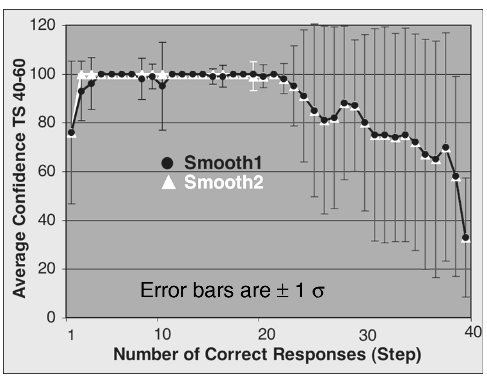

Figure 6: Categorical functions. The average confidence that the Mentored CLA outperformed the Unmentored CLA in test samples 40 through 60 on each step of the categorical functions.

In the Categorical functions, the most dramatic change in performance occurred between test samples 40 and 60. The average confidence that the Mentored CLA's performance was better than the Unmentored CLA's performance between test samples 40 and 60 is shown in Fig. 6. Mentoring holds a strong advantage in the first 20 steps of both the Smooth1 and Smooth2 Categorical functions midway through training. In the last 20 steps, there is no significant advantage to mentoring midway through training.

In the Categorical case, the average confidence that the Mentored CLA performed better than the Unmentored CLA is 100% for test samples 41 through 100 for steps 1 through 38. Only in the last two steps of the Categorical functions, where there are only two stimuli that share the same response (step 39) or no stimuli that share the same response (step 40) is there no significant advantage to mentoring by the end of the match. The results for the Smooth1 function in Fig. 6 show a slightly higher variance that can be attributed to the fact that when more stimuli share the same response, the dissociative player has greater influence until a `Mentor` is found that is effective for the majority of stimuli. From these results it is concluded that mentoring is reasonably insensitive to the dispersion of the responses among stimuli.

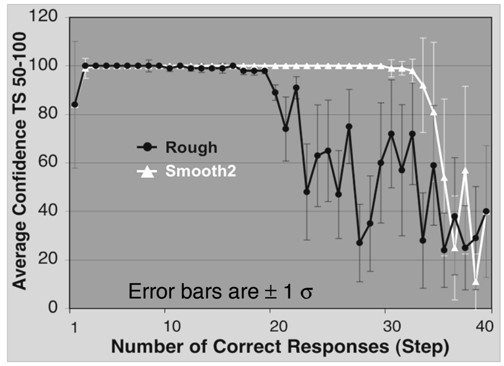

Figure 7: Ordered functions. The average confidence that the Mentored CLA outperformed the Unmentored CLA in test samples 50 through 100 on each step of the categorical functions.

In the Ordered functions, improvements in performance occurred over the second half of testing, so Fig. 7 shows to average confidence that the Mentored CLA's error was lower than that of the Unmentored CLA in test samples 50 to 100. The Ordered functions were designed to investigate the impact of the dispersion of the responses in the range and, as Fig. 7 shows, there is a strong impact as more and more correct responses are introduced. The Mentored CLAs perform significantly better than the Unmentored CLAs in step 2 to 18. After step 18, the Mentored CLA shows no significant improvement over the Unmentored CLA on the Rough function, but the Mentored CLA significantly outperforms the Unmentored CLA until step 30.

The Ordered results need to be put into context in order to be fully understood. In the Ordered functions, an error less than or equal to 2 was considered acceptable. In the Rough function, both Mentored and Unmentored CLAs were averaging error rates less than 2 in the second half of the match by step 6. In contrast, on the Smooth2 function the Unmentored CLA did not consistently average error rates less than 2 until step 32. Before step 32, there were several functions where the Unmentored CLA *never* achieved an acceptable error rate. On the other hand, the Mentored CLA *always* achieved an acceptable error rate by the end of the match and in most steps this was achieved well before the end of the match.

For the Ordered functions, it is concluded that increased dispersion in the range (Rough) makes learning easier for all CLAs and that decreased dispersion (Smooth2) makes learning harder for Unmentored CLAs, but not Mentored CLAs.

CONCLUSIONS AND FUTURE DIRECTIONS

This work presented a new response-selection policy for a Collective Learning Automata: **mentoring**. An initial set of formal experiments showed that the new policy is an effective strategy for accelerating and improving learning performance for two broad classes of learning problems: Categorical and Ordered. These experiments also showed that even when mentoring does not provide a significant improvement in performance, it does not significantly hinder performance and incurs no significant costs.

These conclusions suggest that mentoring is an effective strategy to apply even when the exact nature of the problem is not well understood. To pursue this possibility, several aspects of the method will be explored in future work, notably:

- varying the collection length.
- using a much larger range of stimuli.
- using multi-valued functions both in the Categorical and Ordered cases.
- exploring possible explanations for the apparent degradation of performance in the Standard CLA. Informal observations suggest that using the Bayesian posterior probability may be forcing Standard CLAs into suboptimal solutions and that mentoring does not suffer from the same problem.
- including mentoring in the ALISA classifier (Bock, 1998).
- investigating the parallels between mentoring and known biological and psychological phenomena.

REFERENCES

Beer, S., 1966, *Decision and Control: The Meaning of Operational Research and Management Cybernetics*, West Sussex, England.

Berk, L., 2003, *Child Development*, Allyn & Bacon, Boston, MA.

Bock, P., 1998, "ALISA: Adaptive Learning Image and Signal Analysis", *Proceedings of the SPIE Applied Imagery Pattern Recognition Conference*, Washington DC.

Bock, P., 1993, *The Emergence of Artificial Cognition: An Introduction to Collective Learning*, World Scientific, New Jersey.

Bock, P., 1976, “Observation of the Properties of a Collective Learning Stochastic Automaton”, *Proceedings of the International Information Sciences Symposium*, Patras, Greece.

Heckman, K., 2004, *An assessment of the effect of personality and forward context on the expertise of a collective learning system using the FIVE-FACTOR model of personality*. Doctoral Dissertation, The George Washington University Press, Washington, DC.

Mitchell, T., 1997, *Machine Learning*. McGraw-Hill, New York, NY.

Moore, A.W., Lee, M. S., 1994 , “Efficient algorithms for minimizing cross validation error”, *Proceedings of the 11th International Conference on Machine Learning*. Morgan Kauffman, San Francisco, CA.

Piaget, J., 1977 (Originally published 1936), *The Origins of intelligence in children*, Penguin, Harmondsworth, UK.

Portnoy, D., Bock, P., 2005 , “Unsupervised Fuzzy-Membership Estimation of Terms in Semantic and Syntactic Lexical Classes”, *Proceedings of the 33rd IEEE Applied Imagery Pattern Recognition Workshop*. Los Alamitos, CA.

Pulaski, M., 1980, *Understanding Piaget: An Introduction to Children’s Cognitive Development*. Harper & Row, New York, NY.

Sycara, K., Guttal, R., Koning, J., Narasimhan, S., Navinchandra, D., 1992 , “CADET: A case-based synthesis tool for engineering design”. *International Journal of Expert Systems*, 4(2), 157-188.