COBOSLAB

Cognitive Bodyspaces: Learning and Behavior

Technical Report No. CoboslabY2011N001
18th of April 2011

Extracting Knowledge with XCS on Scarce and Noisy Data

Anke Endler, Martin V. Butz & Günter Daniel Rey

COBOSLAB, Department of Psychology
University of Würzburg
Röntgenring 11
97070 Würzburg, Germany
http://www.coboslab.psychologie.uni-wuerzburg.de

Extracting Knowledge with XCS on Scarce and Noisy Data

Anke Endler* Martin V. Butz† Günter Daniel Rey‡

Abstract

This paper investigates XCS performance on scarce and noisy artificial and real-world data sets. The real-world data set is derived from an E-learning study, in which motivations were correlated with the adaptation of task difficulty. The artificial data set was generated to evaluate if XCS can be expected to mine information from the real-world data set. By progressively increasing the sparsity and noise in the artificial data set, mimicking the properties of the real-world data set, we show that XCS can handle scarce and noisy data well. We furthermore show that the extracted structure contains problem-relevant information. The analysis of the XCS rules produced in the real-world data set itself reveals structures that correspond to actual psychological learning theories. Thus, the contributions of the paper are twofold: (1) We show that XCS can mine highly scarce and noisy data; and (2) the results suggest that the current motivational state of a user of an E-learning program may be utilized to adapt the program for improving learning progress.

1 Introduction

XCS is a learning classifier system, which was introduced by Stewart Wilson (Wilson, 1995). The system has been successfully applied to various classification and datamining problems as well as reinforcement learning problems (Bernadó-Mansilla & Garrell-Guiu, 2003; Bull, 2004; Butz, 2006). Moreover, due to its rule-base representation, knowledge extraction is easy to accomplish (Wilson, 2000). That is, the system is not only suitable to yield good classification accuracy but is also suitable to extract particular feature dependencies hidden in the analyzed data.

We utilize an XCS version that processes integer-valued inputs, similar to Wilson's XCSI setup (Wilson, 2001b). Moreover, we do not use any action encoding, similar to the XCSF setup, which is typically used for function

*anke.endler@uni-wuerzburg.de
†butz@psychologie.uni-wuerzburg.de
‡rey@psychologie.uni-wuerzburg.de

approximation (Wilson, 2001a). Thus, our XCS setup specifies no action or classification and the system predicts one reward value. To avoid further name confusions, we will refer to our setup as an XCS system.

We analyze XCS performance in cases where only very scarce and noisy data is available for learning. In particular, we investigate if XCS is able to extract feature dependencies and interactions when facing a highly scarce and noisy real-world data set. This data set was extracted from an E-learning study, in which the current user motivation was correlated with task difficulty adjustments. To analyze this data, we first evaluate XCS performance on artificially generated data with a hidden structure. This data is progressively made scarce and noisy. Finally, we analyze the real-world data set. The results show that XCS is well-able to extract feature dependencies from scarce, highly noisy data. With respect to the E-learning study, the results suggest particular motivation-dependent difficulty adjustment strategies.

We now first introduce the real-world data set and the artificial data set used. Next, we specify the XCS setup in our information extraction scenario. We then analyze XCS performance first on the artificial data set, progressively making the data more scarce and noisy. Finally, we analyze XCS performance on the real-world data, extracting interesting knowledge. Summary and future work suggestions conclude the paper.

2 Data Encoding and Generation

The focus of this work was to extract adaptation strategies for E-learning programs based on the motivational state of the user. Thus, an experimental study was conducted in which an E-learning program generated random adaptations of task difficulties, effectively gathering a rather wide spectrum of data for the extraction of useful adaptation strategies. We now give a short overview on the adaptive E-learning program and the nature of the data extracted from the program.

The utilized E-learning program was a computer-based training program to solve simple mathematical puzzles. In particular, either a series of numbers, such as $1, 2, 3, 4$ was to be continued or a set of numbers and operators had to be combined to yield a target result. The user had to choose one answer to each puzzle out of five options given by the program. Both tasks were available in six levels of difficulty.

Fig. 1 shows a typical training block of the E-learning program. Each participant worked on two successive blocks, consisting of a motivational questionnaire to assess the learner's current motivation, a learning block presenting 10 tasks, and two test blocks with 6 tasks each to measure learning success. At the beginning of the learning block, adaptation of task difficulty took place, increasing, decreasing, or maintaining the previous level

of difficulty. This adaptation of difficulty was randomly applied by the E-learning program during the study to gain a broad sample of data. Within the learning block, difficulty was further adapted according to the user's performance, increasing (decreasing) the difficulty after two successive correct (incorrect) answers.

The study was conducted with 37 participants, yielding a data set of 74 data entries. Further details on the study, the participant distribution, and prior data analyses can be found in (Endler, 2010).

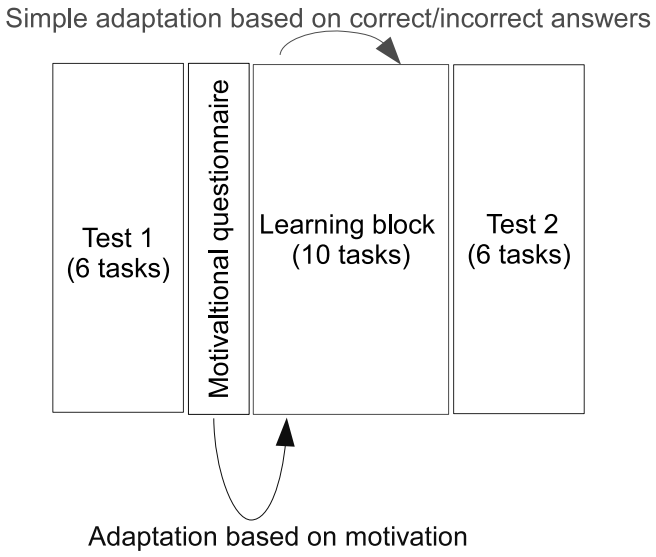

Figure 1: A training block of the E-learning program

Four motivational states were recorded, namely anxiety, probability of success, interest and challenge. These were assessed by means of a reduced form of the questionnaire introduced in (Rheinberg, Vollmeyer, & Burns, 2001). The assessed motivational states were encoded by three possible values: low, medium, or high. As the adaptation of task difficulty was applied by the E-learning program at random, it was defined as part of the condition input specifying either a decrease, increase, or no change of the level of difficulty. The prediction value was the change in performance, calculated as the percentage of correct answers in the test after a learning block minus the test before that learning block. Learning success ranged from -0.5 to $+0.5$, and was normalized to $[0, 1]$ for the knowledge extraction.

Consequently, the features of each data entry consisted of 5 nominal values, each of which could take on three actual values. This yields a problem input space size of $3^5 = 243$ possible input values. Since the study pro-

Table 1: An overview on the distribution of motivation and adaptation in the study. Every table entry shows the amount of data collected for a specific motivational state further sorted by applied adaptation. The four motivational states are anxiety (anx), challenge (chal), interest (int) and probability of success (suc).

Anx	Chal / Suc / Int	low			med.			high		
		low	med.	high	low	med.	high	low	med.	high
low	low	1,0,0	3,0,0	2,0,0	0,0,0	2,2,1	1,1,1	0,0,0	0,1,0	1,2,0
	med.	1,0,0	1,1,0	0,0,0	0,0,1	0,0,0	0,0,0	0,0,0	0,0,0	1,3,2
	high	0,0,0	0,0,0	1,2,1	0,0,0	0,0,0	0,2,0	0,1,0	0,0,0	0,0,0
med.	low	0,0,3	0,1,0	0,0,0	0,1,0	1,3,0	0,0,0	0,0,0	0,0,0	0,1,1
	med.	0,0,0	0,0,0	0,0,0	0,1,0	1,1,0	0,0,0	0,0,0	0,0,1	0,1,0
	high	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,1,0	1,0,0
high	low	2,0,0	0,0,0	0,0,0	0,0,1	0,1,0	0,0,0	0,0,0	0,1,0	0,0,0
	med.	1,0,0	0,0,0	0,0,0	0,0,0	0,1,1	0,0,0	0,0,2	1,1,1	1,0,0
	high	0,0,0	1,0,0	0,0,0	0,0,0	0,1,0	0,0,0	0,0,0	1,1,2	0,1,0

vided us with 74 data entries, maximally 30.5% of the problem space could be covered. Table 1 shows the distribution of the data derived from the study. Every table entry shows the amount of data collected for a certain motivational state, sorted further by adaptation. Within a table entry, e.g. 1,2,3 means that for this motivational state there is one data set where difficulty was decreased, two where it was not changed, and three where it was increased. Motivational states that were not covered by the data are highlighted.

3 XCS Setup

The objective was to let XCS learn rules that can predict the suitability of particular adaptations of difficulty, given the learner's motivational state. XCS is particularly suited for this task since its rule-based representation facilitates knowledge extraction. However, it was not clear to us whether XCS is able to deal with the sparsity and noise in our data set. In the following, we specify the XCS setup. Next, we explore if XCS is able to extract the structure embedded in a sparse artificial data set, before moving on to the real data set analysis.

Given the specified integer encoding of the five data entries, we use interval conditions according to (Wilson, 2001b). Mutation was also defined accordingly. Thus, the number of expressible classifiers was $(\frac{4 \cdot 3}{2})^5 = 6^5 = 7776$.

If not stated differently, parameter settings were the standard parameter settings specified in (Butz, 2006). Since there are no actions and thus no action sets, the evolutionary algorithm was applied in the match sets. Reward prediction updates were done based on the standard error-based adaptation scheme. Further information on the exact functionality of XCS can be found in the algorithmic description (Butz & Wilson, 2001).

4 Evaluation on Artificial Data

The decision tree shown in Fig. 2 was implemented to simulate an intuitively logical and easily verifiable artificial scenario. Nodes in the tree specify motivational states. Leafs specify the reward values dependent on the type of difficulty adjustment. Thus, for example, given low to medium anxiety (< 2) and low challenge, values suggest that an increase in the level of difficulty will yield a reward of 0.8, which would correspond to a solid performance improvement based on our reward definition in the real data set, as it corresponds to an increase of approximately two correct answers in the test blocks.

We now proceed with analyzing if XCS is able to identify this systematics while progressively adding noise and sparsifying the available data. First, we use artificial data with values taken from the entire input space. Next, we limit the number of different input values to the number of values we derived from the study, that is, we generate 74 data set instances and randomly sample those during the learning process. Then we add noise to imitate the inaccuracy of real data. 10-fold cross-validation is used to ensure that XCS can produce rules that are able to predict previously unknown data. Finally, XCS is evaluated with the real data from the study. Again, 10-fold cross-validation is used to ensure that the rules derived with the XCS from this data cannot only predict known input but also unknown data.

Moreover, we evaluate XCS performance dependency on three crucial parameters: its maximum population size N, the number of learning iterations T, and the start of the compaction mechanism C. Intuitively speaking, parameter N fosters competition in the population: give a large population, competition is low and learning is delayed, however, given a very small value, competition may be too strong, possibly leading to the problem of a continuous covering-deletion cycle (Butz, Kovacs, Lanzi, & Wilson, 2004) or niche loss (Butz, Goldberg, Lanzi, & Sastry, 2007). Parameter T specifies the time during which XCS can evolve appropriate rules. Enough time needs to be provided for XCS to converge - however, overly long learning may also result in overfitting. The compaction mechanism simply stops mutation and crossover operators from being applied, thus condensating the population to the most dominant, accurate classifiers at that time. All reported evaluations are done using independent learning runs, reporting the average values

for mean absolute error and number of distinct classifiers. The average error of one run is the mean of the absolute errors of all prediction values and the overall average error is the mean of the average errors of all independent runs. When we use cross-validation, the closest matching classifiers are used for reward prediction if no classifier matches a particular instance.

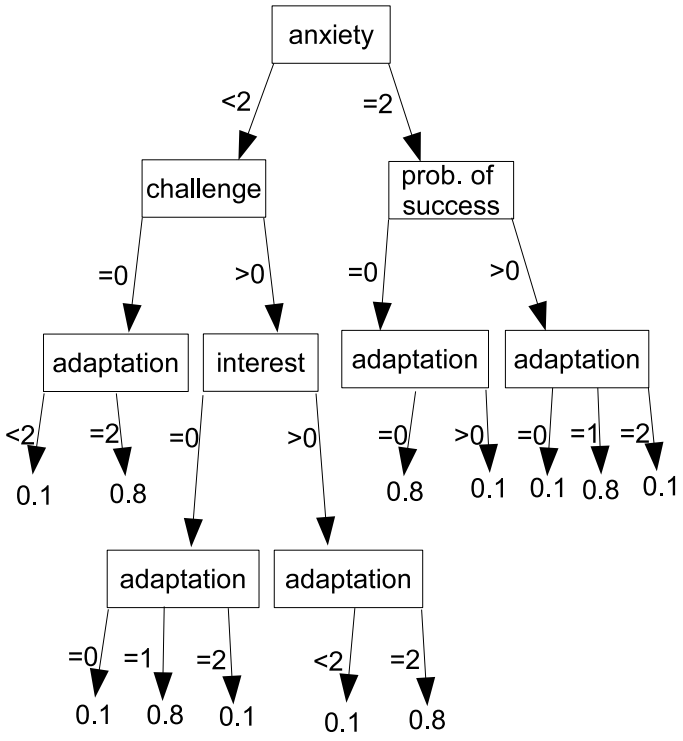

Figure 2: The hidden interdependencies of the input variables in the artificial data set

4.1 Performance on Full Data Set

Fig. 3 shows the average results for average error and number of distinct classifiers, respectively, for various choices of N and T. Each point essentially reports the results of ten independent runs with particular settings for N and T and the compaction C set to 90% of T. Input values were sampled from the entire problem space (with replacement). We can see that for $N \geq 400$ and $T \geq 40,000$ the average error stays below 0.03. For $T \geq 30,000$, T only has an insignificant influence on the number of distinct classifiers. As a general rule, the number of distinct classifiers increases with increasing N. Therefore, low N is preferable, and we conclude that $N = 400$ is a good choice for our problem. Furthermore, T should be at least 40,000

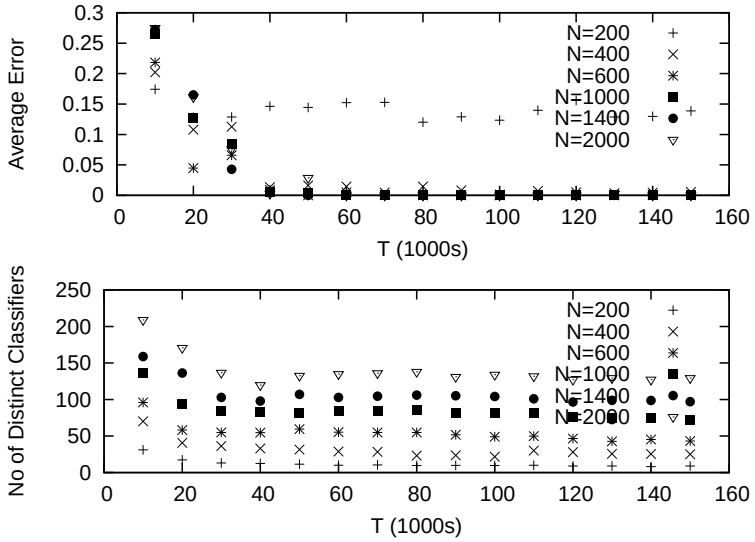

Figure 3: Artificial data with full data set and $C = 90\%$ of T

to gain acceptable results for both the average error and the number of distinct classifiers.

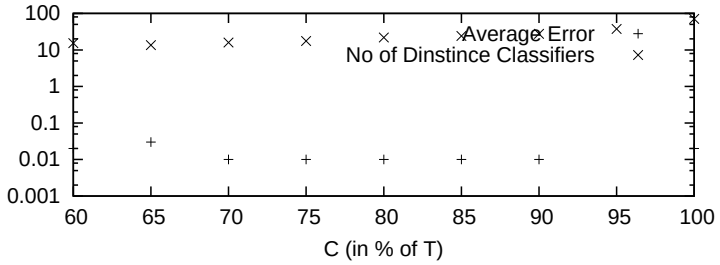

Figure 4: Artificial data with full data set with $N = 400$ and $T = 100,000$, log-scaled.

Fig. 4 shows the average error and the number of distinct classifiers for varying C ranging from 60% to 100% of $T = 100,000$. The figure shows that the later the compaction starts the more distinct classifiers remain in the population. The average error shows a slight tendency to decrease for a later start of compaction. We choose $C = 75\%$ of T for further evaluation to avoid a large number of classifiers.

4.2 Performance on Size-Limited Data Set

In the previous section, XCS was provided with input from the entire problem space. The study, however, provides only a very small amount of data

for knowledge extraction. Therefore we evaluate the system limiting the number of different input values to 74. These input values, i.e. motivational states and adaptation, are taken from the real-world data but the reward is generated artificially using the introduced function. This ensures that the data given to the system covers the same part of the problem space as the real-world data but we can still verify the quality of the rules, because the structure of the data and thus the reward values are known. The 74 available values are presented to XCS repeatedly and randomly with replacement. We choose $C = 75\%$ of T and, again, vary T and N.

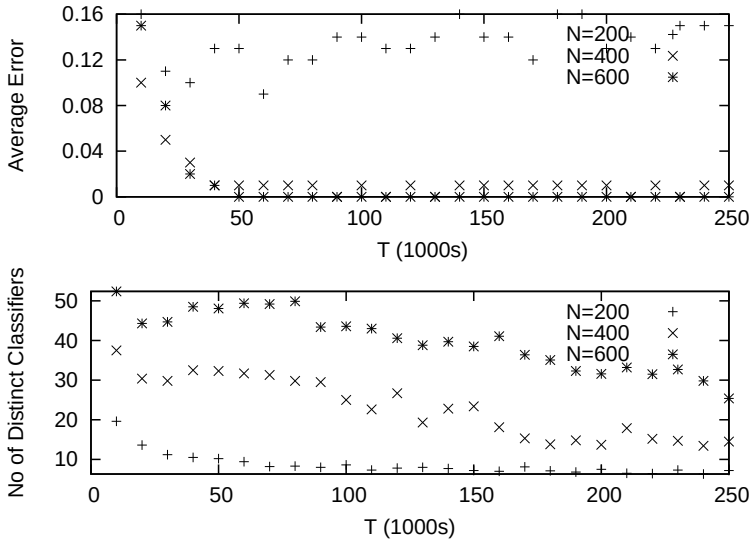

Figure 5: Size-limited artificial data set with C of 75% of T

Fig. 5 shows that for $T > 30,000$ a larger N results in a lower average error. $N = 200$ is clearly too small, resulting in a high error of more than 0.1. $N = 600$ has a slightly smaller error than $N = 400$, but the error stays under 0.02 in both cases and is therefore acceptable. A larger N results in a larger number of distinct classifiers. Therefore, $N = 400$ seems a good compromise. There is no clear indication for the best choice of T. Average error reaches an acceptable range for $T > 40,000$, and does no longer change significantly after that. A higher choice for T results, as a tendency, in a smaller number of distinct classifiers but poses the danger of overfitting. Overfitting means that the system very exactly learns the examples it has been given, so it can predict them accurately but it will be less able to predict new, unknown data correctly.

A batch of 18 classifiers, with an average error of 2.16E-7, was derived with $T = 100,000$, $C = 75\%$ of T and $N = 400$. 13 of these 18 rules predict the reward correctly for their entire coverage. These rules cover

Nr	Anx	Succ	Int	Chall	Adapt	Rew	Err	Fit	Tot	Set
1	0	0-2	0	0-2	1	0.8	0.0	0.16	0.66	1.0
2	0-1	1-2	0	0-2	2	0.1	0.0	1.0	0.66	1.0
3	0-2	1-2	0	0-2	2	0.1	0.0	0.43	0.77	1.0
4	1-2	0	0-2	0	0	0.8	0.0	0.29	0.5	1.0
5	1-2	0	0-2	0-2	0	0.8	0.0	0.71	0.5	1.0

Table 2: The incorrect rules for a batch with the size-limited artificial data set

almost the entire problem space. A rule is defined as correct, if its reward prediciton deviates from the correct value by less than 0.35, so it can still be allocated to one of the two learning success values in the test function. By this definition, five rules in this particular batch return incorrect values. Table 2 lists the incorrect rules.

Fitness is apparently no indication for the correctness of the rules, as Rule 2 has maximal fitness. Rules 1, 2 and 3 are incorrect for challenge $= 0$ but correct for challenge > 0. Rule 4 is correct for anxiety $= 2$ but incorrect for anxiety $= 1$. Rule 5, finally, is correct for anxiety $= 2$ and incorrect for anxiety $= 1$. All errors appear in the anxiety < 2 part of the test function and all errors except for those of Rule 5 appear in the anxiety < 2 and challenge $= 0$ part.

Remember that we took the motivational states and adaptations from the data collected in the study. If we look at Table 1 again we can see how these incorrect classifiers can occur. Let us take Rule 1 as an example: The motivational state of this rule covers the first row of Table 1. In this row, for 6 sets of data the value for adaptation is 1, i.e. no change. These sets all have a value for challenge that is greater than 0. Therefore the rule for anxiety $= 0$ and challenge $= 0$ with low interest and adaptation $= 1$ is not covered by the available data, which means that the system cannot learn the correct prediction for this subspace, but the condition may still be included in a rule due to both the covering operator and the variation operators.

The last two columns of Table 2 indicate what percentage of the problem space covered by a rule can still be correctly predicted by this rule, with reference to the entire problem space and the problem space covered by the data set, respectively. All five rules are able to predict the available data correctly. They are, furthermore, able to predict at least 50% of their coverage correctly.

4.3 Performance on Size-Limited and Noisy Data Set

When collecting data in an experimental study, noise has to be expected. To extract rules, the system has to be able to handle this noisy data. Therefore, we test the system by adding Gaussian noise to the artificial data. Since noise is implicitly added only once when collecting the real-world data, we

add noise to the artificial data by calculating one noisy output for each of the 74 input values. This means, that a certain data set will always return the same (noisy) output. Again, the 74 available values are presented to the XCS repeatedly and randomly with replacement.

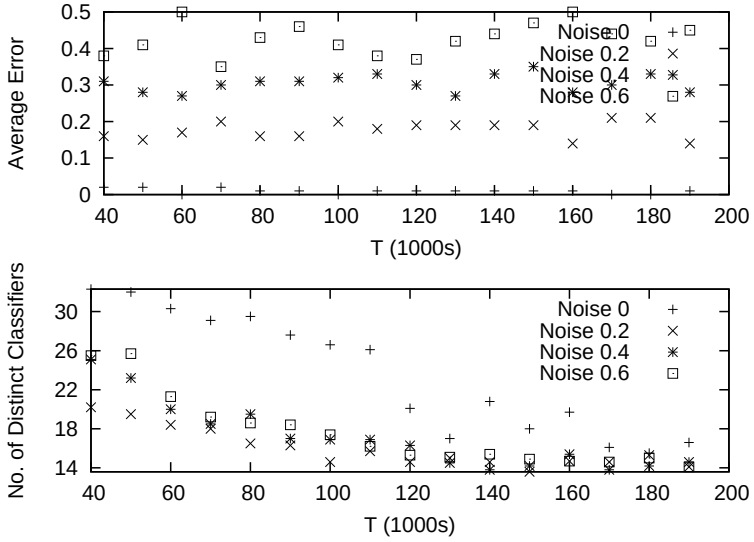

Figure 6: Size-limited, noisy artificial data set with $N = 400$ and $C = 75\%$ of T

Fig. 6 shows that noise has a major influence on the average error. The higher the noise, the higher the average error, irrespective of T. This is not surprising, as it is not possible for the system to learn the correct reward if the reward it receives is noisy and may therefore differ for identical input values. The number of distinct classifiers, on the other hand, drops when noise is added. More noise, however, does not have any significant influence and the number of distinct classifiers, which still decreases for a higher T irrespective of noise. The results suggest that $T = 100,000$ or higher is appropriate. For this setting, we can expect a number of distinct classifiers lower than 20. This is important, because with only 74 values available we need to reach a low number of rules to guarantee some generalisation of the data. A higher T has no major influence on the average error and only a small influence on the number of distinct classifiers but may pose the risk of overfitting.

Table 3 shows one batch of classifiers detected by XCS on an artificial size-limited and noisy data set, with $T = 100,000$ and noise of 0.2. This particular batch reached an average error of 0.178 with 13 classifiers. The last two columns of the table indicate what percentage of the problem space covered by a rule can still be correctly predicted by this rule, with reference to the entire problem space and the problem space covered by the data set, respectively.

Anx	Succ	Int	Chall	Adapt	Rew	Err	Fit	Tot	Set
2	1-2	1-2	0-2	1	0.597	0.35	0.98	1.0	1.0
1-2	0-2	0-2	1	2	0.153	0.28	0.21	0.66	1.0
1-2	0	1	0-2	0	0.815	0.001	0.99	0.5	1.0
1	2	0	1-2	1	1.210	0.00	1.0	1.0	1.0
0-2	1-2	0-2	0-2	0	0.007	0.06	1.0	1.0	1.0
0-1	0-1	0-2	0-2	2	0.747	0.08	1.0	0.78	1.0
1-2	0	1-2	1-2	0-2	0.021	0.03	1.0	0.67	1.0
2	0-1	0	0-1	0-1	0.606	0.13	1.0	0.5	1.0
1	1	0	1-2	1	0.863	0.13	1.0	1.0	1.0
2	0-2	0-2	2	2	0.214	0.09	1.0	1.0	1.0
0-1	0-2	0-2	0-2	0	-0.054	0.16	1.0	1.0	1.0
0	0-2	0	0-2	1	0.824	0.18	1.0	0.67	1.0
0-1	0-2	1-2	0-2	1	0.160	0.18	1.0	1.0	1.0

Table 3: A set of rules for the test function with limited input and noise of 0.2, with $N = 400$, $T = 100,000$ and $C = 75\%$ of T

All classifiers are able to predict the part of the problem space covered by the data set correctly. 7 of the 13 classifiers can predict their entire coverage correctly and all 6 of the remaining classifiers are able to predict at least 50% of their coverage correctly.

4.4 Cross-Validation with Size-Limited and Noisy Data Set

As Section 4.2 showed, the limited values may result in partially incorrect rules despite a low error, because only part of the problem space is covered by the data. To make sure we gain reliable rules, we need to test them on unseen data using e.g. cross-validation. We repeatedly let XCS learn with about $\frac{9}{10}$ of the data and test the rules on the remaining $\frac{1}{10}$, i.e. we use 64 or 65 input values for learning and the remaining 7 or 8 values for evaluation.

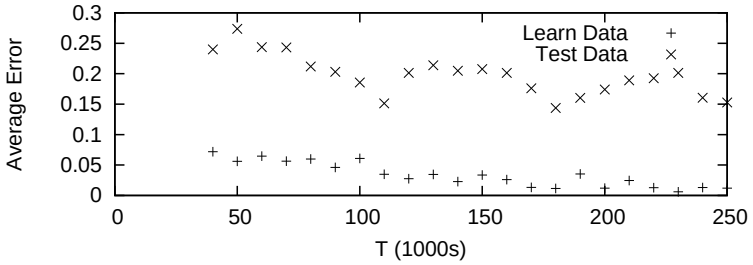

Figure 7: Size-limited artificial data set with $N = 400$ and $C = 75\%$ of T

Fig. 7 shows the average learning and test error for the artificial data without noise but with limited values and varying T. For $T > 100,000$ the

average learning error is lower than 0.05 with an average test error of around 0.2. This suggests that the amount of data available is sufficient to derive rules that can predict unknown data reasonably well.

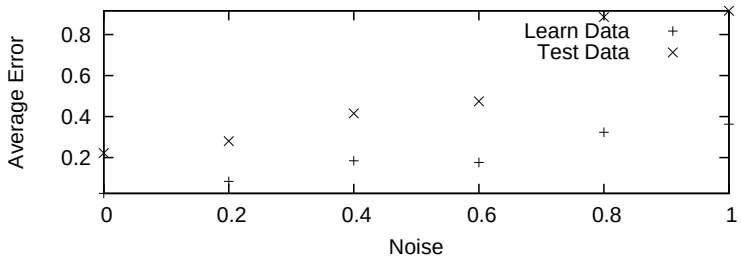

Figure 8: Cross-validation in size-limited, noisy artificial data set with $N = 400$, $T = 100,000$ and $C = 75\%$ of T

Fig. 8 shows the average learning and test error for the test function with several levels of Gaussian noise and $T = 100,000$. Both, learning and test error increase significantly with increasing noise so that for a noise of 0.4 and above the rules derived from XCS cannot feasibly predict unknown data. For a noise of 0.2 or smaller, however, the test data indicates that the system is able to reliably learn an acceptable number of rules that can predict unknown data feasibly. We can conclude that XCS, using adequate settings, is suitable for the extraction of rules from scarce data given that this data is not particularly noisy.

5 Performance on Real-World Data

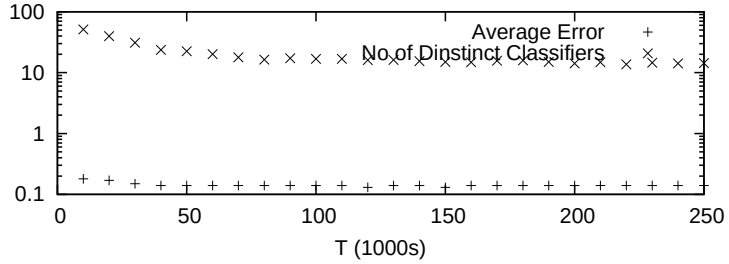

Figure 9: Real-world data set with $N = 400$ and $C = 75\%$ of T, log-scaled

Fig. 9 shows the average error and the number of distinct classifiers for the data derived from the study. The average error remains stable around a value of 0.139 for $T \geq 40,000$. This is comparable to, and even slightly lower than, the average error achieved in the artificial data set for a noise of 0.2, which is an acceptable noise value.

The number of distinct classifiers decreases significantly for up to $T = 80,000$. For higher T, it only decreases slightly, which again coincides with a noise of 0.2 in the artificial data.
Also, similar to the artificial data, there is no clear indication for the best choice of T. We can say that T should be larger than 80,000 so the average error as well as distinct number of classifiers reach an acceptable value. A larger choice of T may reach even better results, albeit probably not significantly, as both values, average error as well as distinct number of classifiers, show no major decrease for higher T. On the other hands, as stated before, a significantly higher T may result in overfitting.

5.1 Cross-Validation with Real-World Data

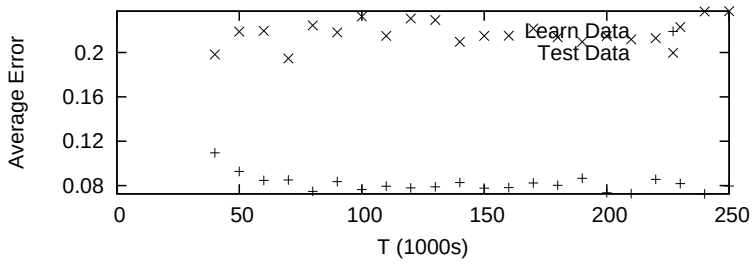

Figure 10: Cross validation with real-world data set with $N = 400$ and $C = 75\%$ of T

As XCS with the real-world data produces similar results to the artificial data with noise of 0.2, we can hope to gain acceptable results using cross validation.
Fig. 10 shows the average error for the learn and test data derived from cross-validation with the real-world data from the study for various choices of T. The learn error stays around 0.08, while the test error never exceeds 0.24. For $T < 240000$ the test error stays around 0.22, showing only a slight increase for even larger T. This increase might imply overfitting. These results show a slightly higher error than the results for artificial data without noise but are, again, comparable to, and even slightly lower than, the results for artificial data with noise of 0.2. We therefore conclude that XCS is able to extract feasible information from the real-world data.

5.2 Knowledge Extraction

Our evaluations using artificial data as well as cross-validation suggest that XCS is able to derive reasonably reliable rules from scarce and noisy data. To support these results further, we extracted a number of stable rules for real-world data using cross-validation and $T = 160,000$. Furthermore, we

No.	**Anx**	**Suc**	**Int**	**Chal**	**Adapt**	**Rew**
1	0-2	0-2	0-1	1-2	0	0.727, 0.833
2	0-2	2	0-1	0-1	0-1	0.379, 0.396, 0.423, 0.597
3	0-2	1-2	1	0-2	0-1	0.596, 0.633
4	0-2	0-2	1	0-2	0-1	0.620, 0.833
5	1-2	0-2	0-1	0	0-2	0.479, 0.538
6	1-2	1-2	0-1	1-2	0-2	0.670, 0.625, 0.634
7	1-2	0-2	0-2	0	0-2	0.454, 0.473, 0.535
8	2	0-2	0-2	0-2	1	0.704, 0.709, 0.741, 0.749, 0.763
9	0	0-1	0-2	0-2	1-2	0.462, 0.473, 0.497, 0.502
10	0-2	2	1-2	0-2	1-2	0.454, 0.517, 0.594
11	0-2	2	1-2	1-2	1-2	0.466, 0.513
12	0-2	2	2	0-2	1-2	0.437, 0.481

Table 4: Rules derived with cross validation from the study's data

chose $N = 400$ and $C = 75\%$ of T. Using these settings, the average number of distinct classifiers after each of the ten runs was 15.6.

To extract meaningful rules, we only considered classifiers with a fitness greater than 0.9 and experience greater than 5000, leaving us with an average of 6.1 classifiers per run. From these classifiers we extracted those that appeared in at least two iterations, which yielded 12 rules in total, which are shown in Tab. 4. The first column gives a rule number to every rule. The next five columns show the rule and the last column shows the reward prediction from every instance of this rule. Predictions for the same rule always show a similar value with the highest deviation being 0.218 for Rule 2. Rule 4 subsumes Rule 3, Rule 7 subsumes Rule 5 and Rule 10 subsumes Rule 11 as well as Rule 12.

5.3 Knowledge Analysis

To analyze the validity and utility of the extracted knowledge, we now interpret the extracted rules according to two well-known, widely accepted psychological learning theories that include motivational aspects.

5.3.1 Zone of Proximal Development (ZPD)

The zone of proximal development (see e.g. (Rey, 2009), (Murray & Arroyo, 2002)) predicts that the highest learning success can be expected if expertise and difficulty are on a similar level as depicted in Fig. 11.

Rules 1 and 2 predict a particularly high and particularly low learning success, respectively. They show no difference in anxiety and interest. Rule 1, however is applicable for rather high challenge and Rule 2 for rather low challenge. Both rules suggest a decrease of difficulty. Rule 2 also covers no

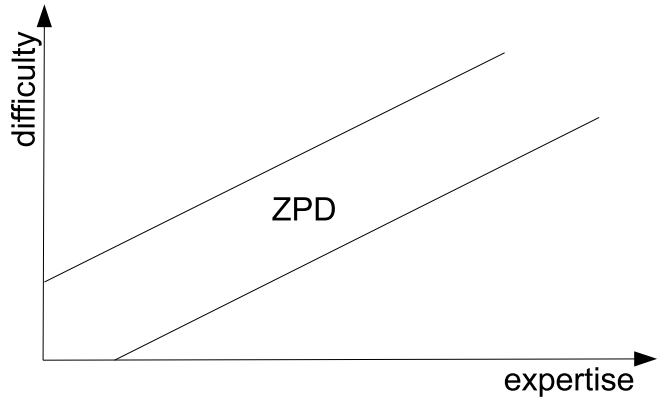

Figure 11: Zone of Proximal Development

change in difficulty.

We can interpret challenge as an indication for the interaction between task difficulty and learner's expertise. More precisely, this interpretation assumes that high challenge indicates that the difficulty exceeds the learner's expertise while low challenge indicates, vice versa, that the learner's expertise exceeds difficulty. A balance of difficulty and expertise will, in this scenario, result in medium challenge. High challenge should then indicate that difficulty is too high and the learner will leave the ZPD. In this case reducing the level of difficulty will return the learner into the ZPD and therefore, as a tendency, increase learning success (see Rule 1). If, on the other hand, difficulty is decreased further when a learner reports low to medium challenge they leave the ZPD because difficulty is too low for their expertise, resulting in low learning success (Rule 2). The same holds if the difficulty is not decreased for high challenge (Rule 11). All other rules give no indication for either challenge or adaptation.

5.3.2 Yerkes-Dodson Law

The Yerkes-Dodson Law (see e.g. (Weiner, 1985)) assumes an interrelation between arousal, difficulty of the task, and performance. The law postulates that a certain amount of arousal, i.e. motivation, is necessary to activate learning. For easy tasks, higher activating motivation is expected to result in higher performance. For difficult tasks, however, too high activating motivation may result in a decrease of performance again.
Anxiety may be such an activating motivational factor. If a learner shows no anxiety whatsoever they may not see any need to concentrate on the task. If, on the other hand, anxiety is too high they may not be able to

concentrate on the task any more.
In the study self-assessed anxiety has to be seen in the context of performance having no consequences for the participants whatsoever. Therefore we assume that even reported high anxiety does not leave the range where it is activating rather than blocking learning. Consequently high (low) anxiety should lead to a high (low) learning success. This is confirmed by Rules 8 and 9, respectively. Medium to high anxiety, which is specified in Rules 5, 6 and 7, shows a wider range in learning success but mainly within the boundaries of the learning success of Rules 8 and 9.
Interest can be analysed in the light of the same law. Interest, however, may take the full range in the scope of the study so that we expect a medium interest to result in high learning success and a low or high interest in a lower learning success. Rules 3 and 4 show an acceptable learning success for medium interest. Rules 2 and 5 show a low learning success for low to medium interest and Rules 10, 11 and 12 show a low learning success for medium to high interest. Rules 1 and 6 however contradict this theory. Rule 1 shows a very high learning success for low to medium interest. This might be due to other factors, like the ZPD explained above. Rule 6 also shows a rather high learning success for low to medium interest.
Rule 6 differs from Rule 5 mainly in challenge, suggesting that high challenge may compensate the lack in interest. Looking at the other rules, low to medium interest together with high challenge always promises a better learning success than low to medium interest with low challenge (see Rules 1, 6 and 2, 5 respectively)

6 Conclusions

In this paper we evaluated XCS performance on scarce and noisy data. With only 74 data sets, the learning material did not cover the entire problem space. Similarly, only a small number of sets where available for each covered condition, making noise a potential issue.
We used data from an intuitively logical and easily verifiable artificial scenario to evaluate XCS performance, making the input progressively scarce and noisy. First we used the artificial data with input taken from the entire problem space. Then we limited the input to a set of 74 different input values, which were presented to XCS repeatedly and randomly with replacement. Next, we added noise to imitate the inaccuracy of real-world data. With this artificial data set and these different stages, we were able to show that XCS is able to extract rules from scarce and to some extend noisy data. 10-fold cross-validation confirmed that, with these rules, XCS can not only predict known input but also unknown data.

XCS was furthermore applied to extract adaptation strategies based on the motivational state of the user for an E-learning program. This real-

world data showed a similar behavior to the artificial data with a noise ratio of 0.2, indicating that XCS could extract knowledge from the data. 10-fold cross-validation as well as consistency with psychological theories on learning strongly supported the reliability of the extracted rules.

We can therefore conclude that, using adequate parameter settings, XCS can handle scarce and, to a certain extent, noisy data reasonably well, and that it is able to extract general and reliable rules from this data.

References

Bernadó-Mansilla, E., & Garrell-Guiu, J. M. (2003). Accuracy-based learning classifier systems: Models, analysis, and applications to classification tasks. *Evolutionary Computation*, *11*, 209-238.

Bull, L. (Ed.). (2004). *Applications of learning classifier systems.* Springer-Verlag.

Butz, M. V. (2006). *Rule-based evolutionary online learning systems: A principled approach to LCS analysis and design.* Berlin Heidelberg: Springer-Verlag.

Butz, M. V., Goldberg, D. E., Lanzi, P. L., & Sastry, K. (2007). Problem solution sustenance in XCS: Markov chain analysis of niche support distributions and the impact on computational complexity. *Genetic Programming and Evolvable Machines*, *8*, 5-37.

Butz, M. V., Kovacs, T., Lanzi, P. L., & Wilson, S. W. (2004). Toward a theory of generalization and learning in XCS. *IEEE Transactions on Evolutionary Computation*, *8*, 28-46.

Butz, M. V., & Wilson, S. W. (2001). An algorithmic description of XCS. In P. L. Lanzi, W. Stolzmann, & S. W. Wilson (Eds.), *Advances in learning classifier systems: Third international workshop, IWLCS 2000 (lnai 1996)* (p. 253-272). Berlin Heidelberg: Springer-Verlag.

Endler, A. (2010). *Towards Motivation-Based Adaptation Strategies for Difficulty in E-Learning Systemy.* Diplomarbeit, Julius-Maximilans-Universität Würzburg.

Murray, T., & Arroyo, I. (2002). Toward Measuring and Maintaining the Zone of Proximal Development in Adaptive Instructional Systems. *International Conference on Intelligent Tutoring Systems.*

Rey, G. D. (2009). *E-learning. theorien, gestaltungsempfehlungen und forschung.* Huber.

Rheinberg, F., Vollmeyer, R., & Burns, B. D. (2001). *FAM: Ein Fragebogen zur Erfassung aktueller Motivation in Lern- und Leistungssituationen* (Tech. Rep.). Universitt Potsdam.

Weiner, B. (1985). *Human Motivation.* Springer.

Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation*, *3*(2), 149-175.

Wilson, S. W. (2000). Get real! XCS with continuous-valued inputs. In P. L. Lanzi, W. Stolzmann, & S. W. Wilson (Eds.), *Learning classifier systems: From foundations to applications (lnai 1813)* (pp. 209–219). Berlin Heidelberg: Springer-Verlag.

Wilson, S. W. (2001a). Function approximation with a classifier system. *Genetic and Evolutionary Computation Conference, GECCO 2001*, 974-981.

Wilson, S. W. (2001b). Mining oblique data with XCS. In P. L. Lanzi, W. Stolzmann, & S. W. Wilson (Eds.), *Advances in learning classifier systems: Third international workshop, IWLCS 2000 (lnai 1996)* (p. 158-174). Berlin Heidelberg: Springer-Verlag.