A Modified Classifier System Compaction Algorithm

Chunsheng Fu
NuTech Solutions, Inc
28 Green St,
Newbury, MA 10951

Lawrence Davis
NuTech Solutions, Inc
28 Green St,
Newbury, MA 01951

Abstract

Although classifier systems have displayed performance levels equaling or exceeding those of other techniques on a variety of benchmark classification problems, they usually solve those problems with a very large number of classifiers. In most cases, a large portion of the final classifier set is unneeded or wrong, with behavior masked by the correctly-functioning rules in the system. Wilson described a post-processing procedure for reducing the number of classifiers in an XCSI classifier system while minimizing the impact of the reduction on the performance level of the system as a whole (Wilson 2001). Wilson's procedure was designed for classifier systems that had been highly trained so that the classifiers were general in nature, and that were always correct in their classifiction of test data. In this paper, we describe some different compaction procedures that can be applied to classifier system sets that are less well-trained, that classify some instances incorrectly, or that contain classifiers that are not fully general.

1 MOTIVATION

XCS classifier systems (Wilson 1995) are competitive with other techniques on real-world classification problems and on benchmark classification problems. XCS's fitness is based on the accuracy of a classifier's payoff prediction. This gives XCS significant improvements on prior classification with respect to prediction accuracy and generality of rules. XCS's capability in both classification and knowledge abstraction makes it unique in solving a variety of real world problems.

One potential benefit of using a classifier system for classification that is frequently mentioned is the possibility that a human might inspect the rules in the system and thereby understand what the system is doing, as compared, for example, with a trained neural network, that contains procedures embedded in a network described by matrices of real-valued numbers. This potential benefit is not fully achieved when the standard approach of training a classifier system to produce hundreds or thousands of classifiers is used, for the following reasons:

- Most of the members of the final set of classifiers do not contribute to the performance of the system as a whole
- Many of those classifiers produced late in the evolutionary process have not been tested, and would degrade performance of the system as a whole, except that more experienced classifiers mask their effects
- Many of those classifiers that are less accurate or less general could be eliminated from the system without impacting performance

For these reasons, when a classifier system is trained, it is likely to contain a majority of macroclassifiers that confuse a human inspecting the system, are inferior in performance to other classifiers in the system, or are wrong but were generated through the evolutionary process and have not yet been eliminated.

Wilson addressed the need for a process that "compacts" a trained set of classifiers by specifying a procedure that could be used on an XCSI system to reduce its size from thousands of classifiers to 20-30, in the examples he considered (Wilson 2001). Wilson's procedure yielded dramatic reductions in classifier system size while resulting in low levels of performance reduction on test sets. Wilson used the Wisconsin Breast Cancer data (Blake 1998) as one of the reference problems on which he conducted his experiments, and we have followed him in the use of this problem in the experiments reported below.

Wilson's procedure works well with highly-trained classifier systems containing accurate and general classifiers. But it cannot be used to reduce the size of less well-trained classifier systems, if they produce classifications on training examples that differ from and example's "true" classification.

In this paper we consider some variant procedures that can be used in these other types of situations.

2 ALGORITHM ANALYSIS

2.1 APPROACH 1

Wilson's compact ruleset algorithm ("CRA") operates on a well-trained XCSI classifier system, and begins after the classifier system has achieved perfect performance. The reader is referred to Wilson 2000 for an explanation of that procedure. The procedures here are heavily inspired by Wilson's approach, but have some different features owing to the need to handle classifier systems that do not display 100% performance after training.

The procedure we began with is closest to Wilson's approach, although it differs in several respects. We call it Approach 1. It proceeds as follows.

Step 1: Beginning with the first classifier in the list, eliminate that classifier from the system and determine the level of performance of the resulting system on the training data. If the level of performance is worse or unchanged, delete the classifier from the system. Terminate step 1 as soon as a classifier is found whose deletion reduces the level of performance of the system as a whole. The remaining set of classifiers, including the one whose deletion reduces performance, is used as the input to step 2.

Step 2: Continuing along the list of classifiers, now eliminate each classifier, in order, and consider the performance of the remaining members of the classifier set. If the level of performance is reduced on deletion, retain this classifier. However, do not use this classifier in the subsequent tests in this step. The set of retained classifiers—those that caused performance reductions in this step—is used as the input to step 3.

Step 3: Construct a final set of classifiers (initially empty), a reference set of instances (initially equal to the training data set) and a set of trial classifiers (initially equal to the output of step 2. Repeat the following procedure until the reference set is empty or no classifier in the trial classifier set matches any member of the reference set: Determine how many members of the reference data set each member of the current trial classifier set matches; move the classifier matching the highest number of members of the reference data set to the final set; and delete the instances that it matches from the reference set. Step 3 could create a set of classifiers that match all the examples in the training set, while preferring general classifiers over specific ones. The final set of classifiers produced in this way is the output of our Approach 1 to classifier system compaction.

2.2 COMMENTS ON APPROACH 1

Approach 1 also results in dramatic levels of compaction on the Wisconsin Breast Cancer database problem. In Wilson's paper, Wilson uses classifier systems trained by presentation of 2,000,000 instances, and we suspected that an approach somewhat inspired by his might be sensitive to training levels. As we will show, a high level of training is necessary for best performance of Approach 1. The Wisconsin Breast Cancer problem can be solved to an equally high level of performance after the presentation of 40,000-80,000 instances. One question we consider below is how well Approach 1 works when training levels are in that range.

It is important to note that any rule compaction procedure has two metrics of interest: performance on the training data set (the data used to train the system), and performance on the test data set (a set of instances drawn from the same distribution that were not used in training). The more important metric is the second, and it is the second that we will primarily consider in this paper. It is well-known that classification systems working on data sets with "noisy" or inappropriate classifications can degrade performance on test data if they are overtrained—trained for extremely long periods of time, or trained so that they have the ability to "memorize" anomalous instances in the training set, resulting in reduced levels of generalization on the test set. There is a danger that a procedure requiring very high levels of training, while resulting in high performance on the training set, will actually degrade performance on the test set. We have shown that this can be the case for the Wisconsin Breast Cancer database (Fu 2001). Thus, there may be a practical as well as a performance-related need for rule compaction procedures that work well on classifier systems that are not highly trained. In addition, for a complicated real-world problem (or even a synthetic one such as the 70-multiplexer problem) there may not be enough time available to fully train a classifier system.

With regard to performance on the training set, it is worth noting that Approach 1 maintains performance levels explicitly in steps 1 and 2—no classifier is deleted whose performance reduces the level of performance of the classifier system as a whole. In step 3, performance is not used as a criterion. Instead, coverage of the training set is used. As we will show later, this causes a significant degradation in terms of prediction accuracy.

Step 3 of Approach 1 has some advantages over a performance-related criterion. The final set of classifiers produced by Approach 1 can be smaller than our performance-related criteria, as we will see.

In the remainder of this paper, our version of XCSI uses the following parameter values throughout all experiments: Population size is 3200, learning rate is 0.25, α is 0.1, Error threshold (e0) is 1, γ is 5, GAThreshold is 48, Crossover Probability is 0.8, Mutation Probability is 0.04, Deletion Threshold Experience is 50, Deletion Threshold Fitness is 0.1, Subsumption Threshold Experience is 100, Minimum Number of Actions in match set is 1, Fitness Updating Coefficient is 0.1, Error Updating Coefficient is 0.25, CoverRange ([0,r0]) is 6, Mutation Range ([1,m0]) is 2, and the reward/penalty values are 100/-100

3 RESULTS AND DISCUSSION

The Wisconsin Breast Cancer database, donated by Prof. Olvi Mangasarian, is a database of real-world data collected by Dr. William H. Wolberg to serve as a test case for classification data mining systems (Blake 1998). There are 699 records in the database, and each contains values for 9 attributes. The attribute values are integers, and each ranges between 1 and 10. The attributes have to do with properties of tissue samples, such as: clump thickness, uniformity of cell size, etc. Each record is classified as either benign or malignant. The task of a data mining system on this database is to use the attributes of records whose classification is known ("training records") to learn to predict whether an unseen case (a "test record") is benign or malignant. In other words, the task is to discover patterns and regularities in the data that allow reliable prediction of an unseen record's classification. The measure of performance of a system on this task is the system's accuracy at predicting records that it has not seen during training. It should be noted that a small number (16) of the records in the WBC database have some missing attributes. Our version of XCSI followed the procedure in Wilson's version by regarding a missing attribute as matched by any classifier.

Table 1: Performance of Approach 1 on different classifier sets

Training instances	**5K**	**50K**	**200K**	**1000K**
Initial P	0.9282	0.9207	0.9422	0.9544
Step 1 P	0.9282	0.9137	0.9422	0.9572
Step 2 P	0.9064	0.9062	0.9356	0.9529
Step 3 P	0.6982	0.8919	0.8790	0.9072
Size of CR	23.5	24.0	15.5	14.5

(P means performance; CR means compact rule set)

Approach 1 produces some reduction of performance level on both the training set and the test set as shown in Table 1 and Table 2.

In Table 1, we show the results of using Approach 1 on a run of tenfold stratification of the Wisconsin Breast Cancer (WBC) database, using our implementation of XCSI (Fu 2001). Classifiers (actually, *macroclassifiers*, many of which have numerosity greater than 1) are ordered by numerosity throughout the experiments reported in this paper. Results are presented for four levels of training of the classifier system: 5,000, 50,000, 200,000 and 1,000,000 trials.

The table shows the level of performance of the output of each of the three steps of the compaction procedure on the test data. Wilson's statement that his compaction procedure works best on highly trained classifier systems is borne out here for Approach 1. The highest levels of performance on test data, after compaction, are achieved when the compaction procedure is carried out on classifier systems that have seen the highest number of training examples—much higher numbers than those required to train the system to its optimal level of performance.

Let us consider some points related to the level of performance reduction in Table 1. As a reference, Wilson's application of XCSI without rule compaction to the Wisconsin Breast Cancer database produced results (95.5% accuracy on unseen instances, using tenfold cross-validation) that were better than any previously published results, which were in the range of 94-95% accuracy. A rough characterization of the levels of accuracy on this problem is that 93% accuracy could be achieved by nearly any technique applied to the data—decision trees and neural networks easily achieved this level of accuracy. Prior to Wilson's work on XCSI, 94.5% was state of the art, and anything higher was new ground.

Considering these levels of performance, we see that compaction of the data using Approach 1 reduces the performance of the system in each case well below the level achievable by most of the rival techniques. This might be a problem for classifier system acceptance in, for instance, the commercial arena: if compaction of a set of classifiers to a human-comprehensible size results in performance levels well below those of competing techniques, then classifier systems may not be preferred to decision trees, for example, whose classification strategy is also human-readable, but has higher performance when pruned to comparable levels of simplification.

For this reason, we did extensive experiments on Approach 1, monitoring each of its three reduction steps with regard to performance on the training data. Table 2 displays the initial prediction performance over training data, the initial rule set size, size of the compact rule set after each step, and the final compact rule set's performance on the training data.

Table 2: Performance of Approach 1 on training set during compaction

Training instances	5K	50K	200K
Initial P	0.9730	0.9952	0.9984
Initial Size	1859.0	1863.5	1381.5
CR size after S1	1188.5	585.5	252.0
CR size after S2	80.0	52.0	38.5
CR size after S3	23.5	24.0	15.5
Final Performance	0.7989	0.9793	0.9499

(P means performance; CR means compact rule set; S_i means step i)

As shown in Table 2, the size of the compact rule set decreases if the initial classifiers are trained over more instances. The more training, the less classifiers are needed to represent the system. Also, we note that the more training, the more classifiers are removed by the first and second reduction steps. Finally, note that performance was significantly degraded even over the training data. Since the first two steps of Approach 1 prevent performance degradation over the training data, the performance loss results from step 3. Thus, we considered modification to step 3 in our work on compaction algorithms. We experimented with two variations on Approach 1, which we describe below.

Two modifications to Approach 1

The first variation we implemented was incremental deletion of classifiers. Note that Wilson's original algorithm works at the macroclassifier level—each macroclassifier with numerosity greater than 1 really represents multiple classifiers, and Approach 1 follows him in this. We hypothesized that deleting microclassifiers one at a time, and testing the result on subsequent performance, might yield better "balanced" sets of classifiers. The all-or-nothing approach might produce performance degradations related to the high numerosity of the surviving macroclassifiers, or so we thought.

We applied our incremental deletion procedure to step 2 of Approach 1, yielding what we called Approach 2. We didn't consider step 1, since the result of both approaches to deletion is the same in step 1. In step 2, deleting microclassifiers has the potential to "reweight" the classifier system, yielding more appropriate strengths on the relative recommendations made by the system, after deletion of classifiers whose weight was important to the system's performance.

As we show below, microclassifier deletion does not improve the performance of the system after compaction, and it appears to slightly degrade performance over macroclassifier deletion on the WBC problem. This is very likely because of the "reweighting" effect. Since step 2 only considers the performance of M_i over M_{i-1} (not all classifiers), the reweighting may result in a problem for a slightly-favored action (Fu 2001). We believe that further study of the incremental deletion is necessary, although our experiments did not show that it is useful for the compaction approaches we tested.

The second variation we studied was a different procedure for step 3 of Approach 1, yielding Approach 3. Table 1 shows that the most significant reductions in performance of the compaction algorithm occur at step 3, where performance is not considered when the final classifier set is built. We experimented with a variant version of step 3 that was more in the spirit of steps 1 and 2. The step can be described as following. For a macrostate ordered by numerosity or experience in increasing order, delete the last classifier and check the performance of the remaining classifiers. If the performance is degraded, then the just deleted classifier is reinserted into the head of the classifier list, and so is retained and used in subsequent tests. Repeat the process until every macroclassifier has been tried by this kind of deletion.

Table 3: Performance of Approach 3 on test data

Training instances	5K	50K	200K
Initial P	0.9282	0.9282	0.9422
Step 1 P	0.9282	0.9137	0.9422
Step 2 P	0.9064	0.8921	0.9356
Step 3 P	0.8915	0.8772	0.9217
CR size	41.5	26.0	24.0

(P means performance; CR means compact rule set;)

Table 4: Performance of Approach 3 on the training data

Training instances	5K	50K	200K
Initial P	0.9730	0.9952	0.9984
Initial Size	1859	1863.5	1381.5
CR size after S1	1188.5	585.5	252.0
CR size after S2	80.0	52.5	38.5
CR size after S3	41.5	26.0	24.0
Final Performance	0.9738	0.9960	0.9992

(P means performance; CR means compact rule set; S_i means step i)

Table 3 shows the performance levels of Approach 3 on identical classifier systems trained on the Wisconsin Breast Cancer database for 5,000, 50,000, and 200,000 instances. If we contrast the data in Table 3 with that in Table 1, and if we note that both tables were constructed based on the compaction of identical initial classifier systems, we can see several differences between the behavior of Approach 1 and Approach 3.

The first is that Approach 1 yields smaller sets of classifiers—the output of step 3 is 15.5 classifiers versus 24.0, in the 200,000 case. A second difference is in the levels of performance. After step 2, the microclassifier

deletion technique shows slightly worse performance on the 50,000 case. However, after step 3, the performance-based compaction technique shows substantially higher levels of performance on the test data. We see 89% versus 70% and 92% versus 88% for the 5,000 case and the 200,000 case, although in the 50,000 case, the original procedure does better, with 89% versus 88%. (Our later experiments showed in Table 6 that the ~1% loss is created during step 2.)

As shown in Table 2 and Table 4, both original step2 and the modified step2 reduced the same number of classifiers. To summarize, we can see that lower training levels produce compact rule sets with lower levels of performance for both versions of the compaction algorithm, but performance loss is much greater for Approach 1. We see that Approach 1, however, produces rule sets that are smaller than those produced by Approach 3, and so some tradeoffs are possible when selecting compaction algorithms.

We wished to learn more about the effects of the variant versions of the three steps. To do this, we used highly-trained sets of classifiers (one million instances of training) as input to three versions of the compaction algorithm: Approach 1, Approach 2, and Approach 3. Table 5 shows the results of this study.

Table 5: Comparison of Approach 1 with Approach 2 and Approach 3 on a classifier system trained over 1,000,000 instances

Alg Names	S1S2S3	S1S2mS3	S1S2mS3m
Initial P	0.9544	same	same
Step 1 P	0.9572	same	same
Step 2 P	0.9529	0.9515	0.9515
Step 3 P	0.9072	0.9015	0.9343
Size of CR	14.5	14.5	20.9

(P means performance; CR means compact rule set; Si means Approach 1's step i; Sim means our modified procedure for step i; Si P means step i's performance)

There are several points to note concerning the data in Table 5. First, we see again that the size of the final set is larger when the performance-based version of step 3 is used, as in Approach 3. Second, we see that Approach 3 produces higher levels of performance on the test data. Third, we see that Approach 2 yields worse performance for the original step 3.

Our experimental results suggest a reduction procedure using Approach 3, unless rule set size is an important consideration. That is, we recommend in general using step 1 and step 2 of Approach 1 and our modified step 3. We implemented this reduction procedure and the results, shown in Table 6, support this recommendation.

Table 6: Performance of suggested CRA (S1S2S3m) on different classifier sets

Training instances	5K	50K	200K
Step 2 P	0.9064	0.9062	0.9356
Step 3 P	0.8915	0.8990	0.9217
CR size	41.5	26.0	24.0

(P means performance; CR means compact rule set;)

4 CONCLUSIONS

The XCS family of classifier systems already competes well with other approaches to classification. If it is to compete on problems requiring compact, human-readable solutions, then effective classifier system rule compaction procedures will be needed. In this paper we have discussed three approaches to rule set compaction that were inspired by Wilson's work, but that differ so that they can be applied to the compaction of classifier systems that do not have high levels of generalization or perfect accuracy on all test set examples. Approaches 2 and 3 yield classifier systems of compact size on the Wisconsin Breast Cancer database. They also yield higher levels of performance than Approach 1 on unseen data, and they yield lower numbers of unmatched instances. They also yield reduced sets of larger size than Approach 1.

To conclude, we know that uncompacted classifier systems are already competitive with all other classification techniques with regard to performance level, but they are not compact and are not human-comprehensible. We hope that Wilson's paper and this one will stimulate further work in classifier system compaction, in order to increase the range of real-world situations in which classifier systems are indeed the algorithm of choice for solution of classification problems, and to realize the possibility that the classifier system approach produces both high-performance results and compact sets of high-quality rules.

References

Blake, C. and C. Merz (1998). UCI repository of machine learning databases.
http://www.ics.uci.edu/~mlearn/MLRepository.html

Butz, M. V. and Wilson, S. W. (2001). An Algorithmic Description of XCS. In Lanzi, P. L. , Stolzmann, W., and S. W. Wilson (Eds.) *Proceedings of the International Workshop on Learning Classifier Systems (IWLCS-2000).* Springer-Verlag. Or see http://prediction-dynamics.com

Fu, C. S, Wilson, S. W. and Lawrence, D (2001). Studies of the XCSI classifier system on a datamining problem. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), page 95.* Morgan Kaufmann: San Francisco, CA, 2001.

Wilson, S. W. (1995). Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2), 149-175

Wilson, S. W. (2000). Mining Oblique Data with XCS. In: Technical Report No. 2000028, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign