International Journal of Intelligent Information Systems
2015; 4(6): 101-105
Published online December 14, 2015 (http://www.sciencepublishinggroup.com/j/ijiis)
doi: 10.11648/j.ijiis.20150406.12
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

An XCS-Based Algorithm for Classifying Imbalanced Datasets

Hooman Sanatkar, Saman Haratizadeh

Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

Email address:
hooman.sanatkar@ut.ac.ir (H. Sanatkar), haratizadeh@ut.ac.ir (S. Haratizadeh)

To cite this article:
Hooman Sanatkar, Saman Haratizadeh. An XCS-Based Algorithm for Classifying Imbalanced Datasets. *International Journal of Intelligent Information Systems*. Vol. 4, No. 6, 2015, pp. 101-105. doi: 10.11648/j.ijiis.20150406.12

Abstract: Imbalanced datasets are datasets with different samples distribution in which the distribution of samples in one class is scientifically more than other class samples. Learning a classification model for such imbalanced data has been shown to be a tricky task. In this paper we will focus on learning classifier systems, and will suggest a new XCS-based approach for learning classification models from imbalanced data sets. The main idea behind the suggested approach is to update the important parameters of the learning method based on the information gathered in each step of learning, in order to provide a fair situation for the minor class, to contribute in building the final model. We have also evaluated our approach by testing it with real-world known imbalanced datasets. The results show that our new algorithm has a high detection rate and a low false positive rate.

Keywords: Imbalanced Dataset, Evolutionary Algorithm, XCS

1. Introduction

Learning from imbalanced datasets are among the most important challenges in machine learning issues. In such data sets the share of one class of samples from the data is much more than that of the other classes. In such a situation, a classification algorithm will not lose much accuracy even if it completely ignores the samples from the minor class. Hence, if the learning algorithm is not used with cautions, usually the resulting model will be biased toward the major class. This can be a very serious problem, because most of the times, detecting an unlabeled sample from the minor class (like a patient being infected by a rare disease) is the main goal of the prediction. Different approaches to solve the problem of unbalanced data has been taken in statistics and machine learning domain [1] [2].

Learning classifier systems (LCS) are a set of learning methods that use evolutionary approaches like genetic algorithms to generate rule-based classifiers [10]. In these methods a rule or a set of rules is coded as chromosome and a population of chromosomes is subject of an evolutionary process. There are two classes of LCS's: Pittsburgh and Michigan. The difference between the two, lies in the way they evolve the classifiers: in Pittsburgh approach, each chromosome is a candidate rule set while in Michigan approach, each chromosome is a single rule and the whole population forms the classifier rule set [3].

Like other learning methods, LCS's are not very good at classification of imbalanced data. Some methods has been proposed for dealing with imbalanced data when using LCS's for learning, but most of them have focused on Pittsburgh methods. On the other hand, it seems that Michigan class of LCS's has some interesting features that make it a very flexible method compared to most of other learning methods [4].

In Michigan approach, the model is built gradually while it experiences the training data over and over, and has the chance to refine and update every single rule, based on its performance so far. So the learning process can focus on any classification rule or any sample in the data set that needs more attention during the learning process [5]. This is a very useful feature and we will use it to improve the performance of a basic algorithm called XCS in generating models from imbalanced data sets.

The rest of this paper is organized as follows. We will review the related works in section 2. In section 3, we will describe the architecture and implementation of our algorithm. Section 4 presents the evaluation results of the proposed system on various imbalanced datasets and we conclude in section 5.

2. Related Works

There are several approaches to constructing a prediction model based on the imbalanced datasets. One of the most popular approaches is resampling which has some different approaches, like over-sampling, under-sampling and random sampling. For using with evolutionary algorithms, some variations of resampling have been suggested [6] [7] [8] [7] [8].

In XCS-based approaches some other techniques have been suggested. For example, Orriols [9] [10] [11] has showed that using smaller learning rates on XCS will improve the performance of XCS on imbalanced datasets. That experiment, also shows that low values of GA threshold in the beginning and incrementing it slowly during the learning process can make improvements on classification quality [12].

There are some methods for handling imbalance data sets using multi objective optimization in Pittsburg evolutionary algorithms. These methods follow 2 objectives: maximizing classification accuracy for majority samples and maximizing classification accuracy for minority samples. NSGA, NPGA, and PESA are some Pittsburg evolutionary algorithms that are used in this category [13] [14] [15].

3. Minor Grabber

In this section, we illustrate a new classifier system, Minor Grabber, whose goal is to improve the classification's accuracy on imbalanced datasets using an XCS-based evolutionary algorithms.

3.1. Overview

Minor grabber is a new version of XCS, designed to improve the performance of XCS in imbalanced datasets. We will first explain the XCS algorithm briefly and discuss some of its characteristics which may cause some problems when dealing with imbalanced datasets.

3.2. XCS

XCS is the most famous Michigan evolutionary algorithm with a good performance in data classification. Like all evolutionary algorithms XCS is based on evolution of a population of chromosomes. In XCS each chromosome is a classification rule while the whole population forms a classification model. The main cycle of XCS is as follows:

1. A data sample is presented to the population
2. All rules whose "if" part match the sample are gathered in a set "M" called match set. Each rule suggests a label and predicts a reward that shows how much reward it expects to receive by suggesting that label.
3. Based on the reward prediction of the rules in "M" and the fitness of the rules, a label is selected. All the rules suggesting that label are gathered in a set "A" called action set.
4. The suggested label is compared to the real label of that data sample. If they are the same, a high reward is given to the rules in "A", and if not, the rules in "A" receive a low reward. The reward prediction of the rules are updated based on their current value and this last reward value.
5. In every few cycles, a GA process will be run over the chromosomes in the action set and new rules will be generated and added to the population, possibly causing some old and low quality rules being removed. The fitness measure used for rules, is usually a value proportional to the preciseness of the rules in predicting rewards

3.3. XCS Shortcomings for Imbalanced Datasets

As we know, minority samples are rare in the training set so they will not be presented to XCS as frequently as majority samples. This means that the action set is often containing rules that recommend the label of the majority class. So, the rules that classify a sample as the majority class have a higher chance to reproduce in the population of the rules. This means that the minority class has a little chance to form appropriate set of rules in the population. As a result the population gets biased toward the majority class. In addition if the XCS manages to keep some rules for the minority class in the population, these rules will rarely find a way in to the action set. So XCS will not have enough experiment with these rules to refine them. That means the population will find a mature set of rules suggesting the majority class early in the process while the minority class rules are still young and unable to compete with other rules. Because of this phenomenon, XCS usually can not find a good set of rules when facing a skewed data set.

Minor grabber improves XCS algorithm to deal with imbalanced datasets. In the main loop of algorithm, Minor Grabber counts the number of times that the action set belongs to each class. So, after every cycle of the algorithm, that is one pass through the training set, and at the beginning of next learning cycle, we calculate the percentage of times that the action set has suggested each class of the samples. One can expect that if the rules are formed correctly, this ratio, for each class, should be equal to the ratio of instances of that class in the training data set used in that cycle. Based on this intuition, we adaptively set the number of training sample from each class for the next run as follows:

$$SR_{t+1}(c) = SR_t(c) + \alpha\big(SR_t(c) - RR_t(c)\big) \qquad (1)$$

In this equation $SR_t(c)$ shows the ratio of samples of class "c" at cycle "t" and $RR_t(c)$ stands for the ratio of times "c" is the label suggested by the action set at cycle "t" of the algorithm.

This equation updates the number of instances of each class for every cycle of the algorithm. For the first cycle, the number of instances are set proportional to the number of them in the whole data set. If the number of needed samples of each class is more than its total number of samples, over-sampling will be applied and if this value is less than total number of samples in that class, then, under-sampling will happen. So we will have a self-adaptive system that controls the XCS's learning process by adaptively updating its input training data. Picking a small value for alpha will cause a smooth change of the number of training samples in each class, and will give the system enough time to evaluate its current population of rules. It's worth mentioning that since during the learning process Minor Grabber will use exploration steps, this method can't possibly lead to a situation that one class takes over the whole training set forever.

In essence, the adaptive resampling does the same for every class of the data and the whole process adjusts the ratio of samples in the training data in each cycle so that XCS receives more samples for the classes that have not been learnt well yet. For the special case of skewed data, adaptive resampling puts more samples from minority class in the training samples of the next run, to give the XCS the chance to experience the minority class well.

The remaining problem is that although adaptive resampling adjusts the number of samples of each class in the training data, it does not say anything about which samples in the class need to be in the training set. It is important to pick the samples wisely because the performance of the classifier for each class may not be the same in different parts of the feature space. In other words, XCS may be able to classify some samples of a class correctly and still misclassify those samples from the same class that are in a different part of space.

The intuition used for solving this problem is to give higher chances to those samples that have a higher rate of misclassification by XCS. We will do it here using a sample weighting process. In order to give a sample a resampling weight, Minor Grabber looks at the match set formed for that sample. If there is a close competition among the rules present in the match set for finding a way in to the action set, it concludes that the prediction model for this sample is still immature. So this sample will probably will need more time to be learned well. Minor Grabber will give a higher chance to this sample for being included in the training data of the next cycle by increasing its resampling weight. It is important to note that all the classifiers in the match set are in a neighboring space, because they all match the same sample. So by giving a high weight to a sample we are saying that it lies in a part of feature space in which the prediction model is not mature yet.

Based on resampling weights assigned to samples, and sample ratio of each class calculated by adaptive resampling, one can use a method like roulette-wheel to pick the sample for the training set of the next learning cycle.

Table 1. *Datasets statistics.*

Row	Dataset	Num. of Samples	Imbalanced Ratio	Minority Samples	Majority Samples	Number Of Attributes	Attributes Type
1	CAR_EVAL_4	1377	25.6	56	1321	7	Nominal
2	KR_VS_K_THREE_VS_ELEVEN	2935	35.3	81	2854	7	Nominal
3	KR_VS_K_ZERO_VS_EIGHT	1460	53.7	27	1433	7	Nominal
4	KR_VS_K_ZERO_VS_FIFTEEN	1702	80.22	21	1681	7	Nominal

4. Experiments

To evaluate the performance of Minor Grabber, we have tested it on several skewed datasets, including datasets with low, medium and high imbalanced ratio.

Table 2. *Variables default values.*

Variable	Default Value
Population Size	150
Max Iteration Number	30000
Learning Rate	0.1
Cross Over Probability	0.8
Experience Probability	0.8
Sharp Probability	0.6-0.7
Mutation Probability	0.02
Deletion Threshold	20
GA Threshold	35
Delta	0.1
µ	0.01
P0	0.01
E0	0.01
F0	0.01
alpha	0.1

Clearly generating a good prediction model from a highly imbalanced data set is usually harder and we have chosen data sets with different imbalance ratios to test the performance of the Minor Grabber against problems with different degrees of difficulty. Datasets are gathered from Georgia University repository and California Irvine machine learning repository. The chosen data sets and their specifications are summarized in Table 1.

We have implemented the Minor Grabber and evaluated the accuracy of the resulted model in prediction of the test data for each data set. Table 2 summarizes the default values of the variables in our system.

In order to show how the innovations added to the general XCS algorithm have improved the performance of this algorithm on the skewed data sets we have tested XCS with the same parameter values presented in Table 2. The classification accuracy of XCS on different data sets has been shown in Table 3.

Table 3. *XCS results.*

Dataset	Classification Accuracy
Dataset1	0.9652
Dataset2	0.9709
Dataset3	0.9793
Dataset4	0.9840

According to these results one can see that the classification accuracy on XCS on datasets is high. However, as shown in figure 1, XCS can't learn the minor (positive) class and the true positive rate does not increase as the learning process proceeds. The performance of the XCS is basically the same for all the data sets and figure 1 shows the typical convergence process.

Figures 2-5 show the learning procedure of proposed algorithm for each dataset. The results are the average of 15 runs for each dataset.

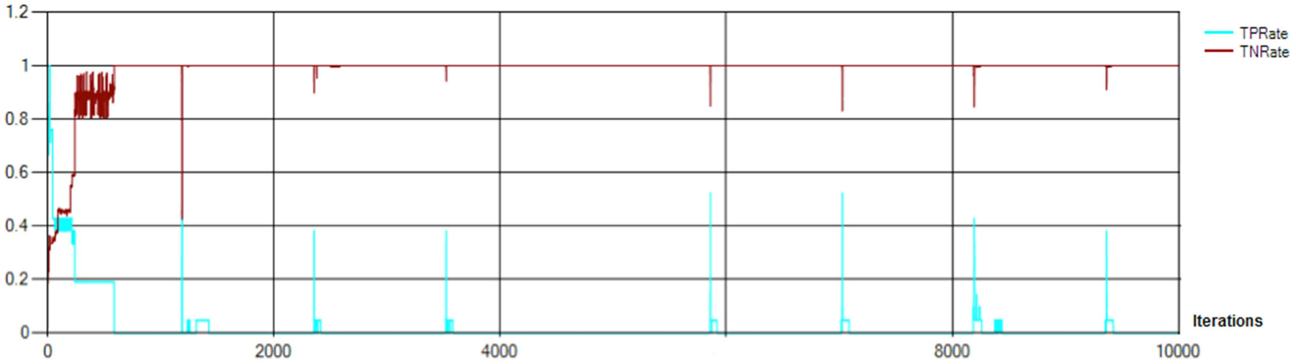

Figure 1. *Typical convergence process of XCS.*

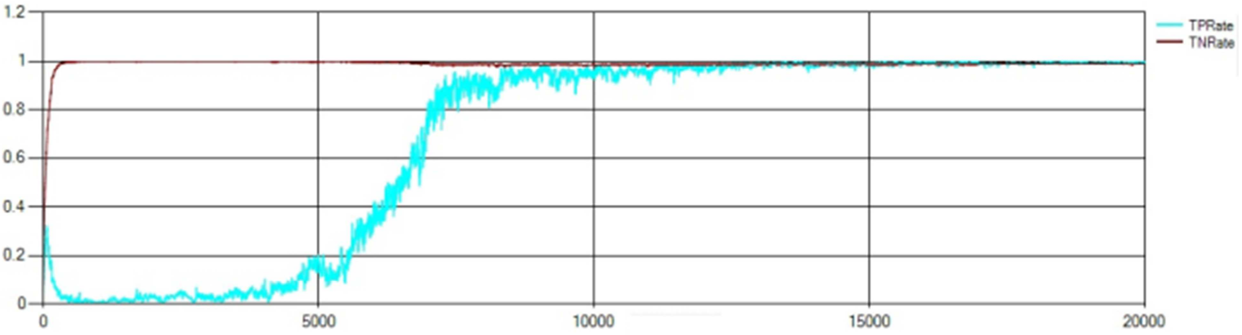

Figure 2. *Convergence process of the new method on dataset 1.*

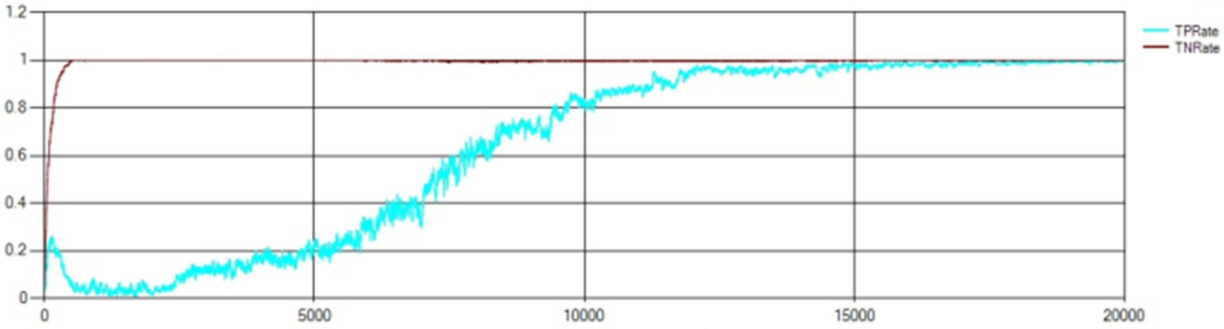

Figure 3. *Convergence process of the new method on dataset 2.*

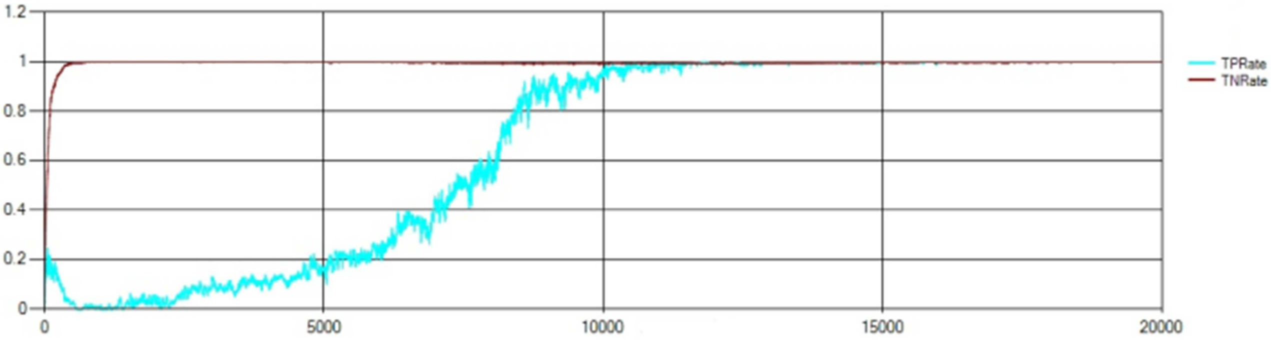

Figure 4. *Convergence process of the new method on dataset 3.*

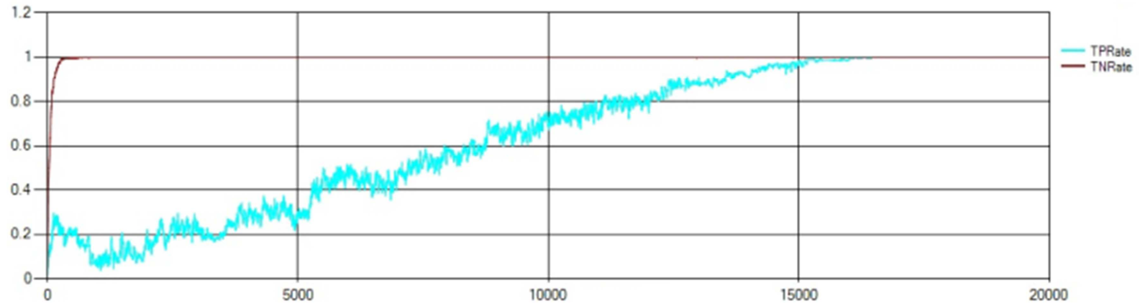

Figure 5. *Convergence process of the new method on dataset 4.*

Table 4. *Minor Grabber results.*

Dataset	TP	TN	Classification Accuracy
Dataset1	0.9960	0.9851	0.9845
Dataset2	1	0.9985	0.9984
Dataset3	0.9833	0.9982	0.9986
Dataset4	1	0.9998	0.9999

Figures 2-5 present the convergence of Minor Grabber on different skewed data sets and Table 4 compares the performance of Minor Grabber with that of XCS. As can be seen in these results, Minor Grabber not only has achieved higher classification accuracies than XCS in all data sets but also has been able to learn the minority class (positive) samples as well. As presented in figures 3-6 the true positive rate improves in all data sets as the algorithm proceeds and finally converges to 1. Although this convergence happens later for the minor class, the important fact is that Minor Grabber can eventually find the correct prediction model while the XCS is unable to do that.

5. Conclusion

In this paper, we presented Minor Grabber, an XCS-based learning classifier system improved by two techniques called adaptive resampling and sample weighting. These two methods help the algorithm to wisely choose samples for the training sets used in each learning cycle. Using this approach the algorithm can adaptively focus on classes that have not been learnt yet, and experiment suitable training instances to improve the model for those classes. The algorithm has been tested against different skewed data sets with imbalanced ratios from 11 to 80, and has shown a quite good performance in learning prediction models for those data sets.

References

[1] A. Bria, N. karssemeijer and F. Tortorella, "Learning from unbalanced data: A cascade-based approach for detecting clustered microcalcifications," *Medical image analysis,* 2014.

[2] Xue, Jing-Hao and P. Hall, "Why Does Rebalancing Class-unbalanced Data Improve AUC for Linear Discriminant Analysis," *Pattern Analysis and Machine Intelligence,IEEE Transactions on ,* pp. 1109-1112, 2015.

[3] M. Butz, "Learning classifier systems," in *Springer Handbook of Computational Intelligence*, Berlin , Springer , 2015, pp. 961-981.

[4] P. L. Lanzi, "Learning classifier systems: a gentle introduction," in *companion on Genetic and evolutionary computation companion*, 2014.

[5] L. Bull, "A brief history of learning classifier systems: from CS-1 to XCS and its variants," *Evolutionary Intelligence,* 2015.

[6] N. CHAWLA, K. Bowyer, L. Hall and W. Kegelmeye, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research,* vol. 15, pp. 321-357, 2002.

[7] N. Japcowicz and S. Stephen, "The Class Imbalance Problem:A Systematic Study," *Intelligent Data Analisis,* 2002.

[8] Wang, Shuo, L. Leandro, Minku and Xin Ya, "Resampling-Based Ensemble Methods for Online Class Imbalance Learning," *Knowledge and Data Engineering, IEEE Transactions on,* pp. 1356-1368, 2015.

[9] A. Orriols-Puig, "Facetwise Analysis of Learning Classifier Systems in Imbalanced Domains," Ramon Liull University, 2006.

[10] A. Orriols-Puig and E. Bernad´o-Mansilla, "Evolutionary rule-based systems for imbalanced datasets," *Soft Computing Journal,* vol. 13, no. 3, pp. 213-225, 2009.

[11] A. Orriols-Puig and E. Bernad´o-Mansilla, "Bounding XCS parameters for unbalanced datasets," in *Genetic and Evolutionary Computation Conference*, 2006.

[12] A. Orriols-Puig and E. Bernad´o-Mansilla, "The Class Imbalance Problem in Learning Classifier Systems:A Preliminary Study," in *Genetic and Evolutionary Computation Conference*, 2005.

[13] K. Deb, A. Pratap , S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II.," *Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[14] M. Erickson, A. Mayer and J. Horn, "Multi-objective optimal design of groundwater remediation systems: application of the niched Pareto genetic algorithm (NPGA)," *Advances in Water Resources,* vol. 25, no. 1, pp. 51-65, 2002.

[15] D. Corne, N. R. Jerram, J. D. Knowles and M. J. Oates, "Corne, David W., et al. "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Genetic and Evolutionary Computation Conference*, 2001.