Rough Set and XCS in Classification Problems

Thach H. Nguyen*, Sombut Foitong† and Ouen Pinngern‡
Department of Computer Engineering, Faculty of Engineering,
Research Center for Communication and Information Technology (ReCCIT)
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 THAILAND.
E-mail: nguyenhuythach1983@yahoo.com, s8060022@ kmitl.ac.th†, kpouen@kmitl.ac.th‡*

Abstract

XCS is known to degrade in classification performance when faced with many features that are redundant for rules discovery. In this paper, we propose a novel system combining of rough sets and XCS to deal with the mentioned problem. Firstly, rough set theory is used to handle inconsistent input datasets. The purpose of feature reduction by rough set is to identify the most significant attributes and eliminate the irrelevant ones to form a good feature subset for classification. Secondly, the reduced datasets are used to create a set of rules by using XCS. The main contribution of XCS to learning theory is its rules generation without experts. Finally, by applying the set of rules, we can classify unseen datasets into their specific classes. Experimental results on real-life datasets show that the proposed method can reduce storage space as well as can preserve and may also improve solution accuracy. Beside that, the rule retrieval time is also greatly reduced because the use of Rough-XCS classifier contains a smaller amount of instances with fewer features. Furthermore, the proposed method has a high potential to be used as a mean to construct a classifier system that copes with incomplete, noisy and chaotic data.

***Keywords**: Classification, Learning Classifier System, XCS, Rough set, redundant datasets.*

I. INTRODUCTION

Learning classifier systems (LCSs) are adaptive systems, often called Genetics Based Machine Learning Tools, that learn to achieve a task through interacting with environment. Introduced in 1995 by Wilson, accuracy-based learning XCS is one of the most successful learning classifier systems [2]. XCS classifier system has recently shown a high degree of competence on a variety of data mining problems. There are some performance comparisons of strength-based fitness systems and accuracy-based fitness systems showing that accuracy-based fitness systems, including XCS, have a significant impact on data mining field [3], [4].

Datasets containing redundant attributes appear in many applications of data mining and machine learning. These datasets may be of very high dimensionality and consist of much complex structure that even most well planned data mining or analysis techniques might have difficulty extracting meaningful patterns from it.

Feature Reduction (FR) is commonly applied as preprocessing step to overcome the curse of dimensionality, where many types of data analysis become significantly harder as the dimensionality of the data increases. FR removes non-informative features and preserves informative ones. The related work to FR involves reduction of pattern dimensionality through feature selection or feature extraction methods. Many methods have been proposed for feature reduction, such as Principal Component Analysis and Genetic Algorithms. Another often used approach in FR is rough sets (RS) [5]. Rough sets theory provides a formal and robust of manipulating the roughness of the knowledge in information systems. It is a powerful tool for data analysis and knowledge discovery from imprecise and ambiguous data. The effectiveness of RS is which has been demonstrated in many different domains as medicine, economics, finance and business [7].

In this paper, we propose a combination of Rough set and XCS, called Rough set-XCS or RS-XCS, to solve datasets containing redundant attributes. Employing an idea from [6], the approximate reduct is

978-1-4244-1692-9/08/$25.00 ©2008 IEEE

used in our approach. By generating different approximate reducts in FR, the “best” subset of features is obtained. We use this subset to train XCS to increase classification accuracy. Eleven UCI datasets were used in the empirical study. The results suggest that our approach improves performance of the classifier system significantly.

The rest of paper is organized as follows: In section II, XCS classifier system is introduced briefly. Section III analyzes the performances of XCS on original datasets. In section IV, the approximate Rough set-based attribute reduction is described. The framework of our proposed method, experimental results and comparisons will be presented on section V. Finally, we present our main conclusions and discuss future works.

II. XCS Overview

XCS represents the knowledge extracted from the problem in a set of rules. This ruleset is incrementally evaluated by means of interacting with the environment through a reinforcement learning scheme and is improved by a search mechanism based on a genetic algorithm. XCS evolves a population [P] of classifiers, where each classifier consists of a condition, an action and four main parameters: prediction p, prediction error ε, fitness F and numerosity *num*. This section gives a short introduction of XCS. The more details are referred to [2].

A. Performance Component

At each step, an input x is presented to the system. Given x, the system builds a match set [M], which is formed by all the classifiers in [P] whose conditions are satisfied by the input example. If the number of actions represented in [M] is less than a threshold θ_{mna}, then covering is triggered to create a new classifier that matches the current input and has a random action from those not present in [M]. From the resulting match set, an action must be selected and sent to the environment. For this purpose, a payoff prediction $P(a_i)$ is computed for each action a_i in [M]. $P(a_i)$ estimates the payoff that the system will receive if action a_i is chosen. It is computed as fitness-weighted average of the predictions of all classifiers proposing that action. The system chooses the winning action based on these prediction values. The chosen action determines the action set [A] which consists of all the classifiers in [M] advocating this action.

B. Reinforcement Component

Once the action is sent to the environment, the environment returns a reward r, which is used to update the parameters of the classifiers in [A] following order [2]: prediction, prediction error, and finally fitness. Prediction p is updated with learning rate as follows:

$$p \leftarrow p + \beta(r - p) \qquad (5)$$

Where $\beta\ (0 \le \beta \le 1)$ is the learning rate. Then, the prediction error is updated as: $\varepsilon \leftarrow \varepsilon + \beta(|\,r - p\,| - \varepsilon)$. Finally, classifier fitness is updated in two steps: first, the *relative accuracy* κ’ of the classifiers in [A] is computed; then κ’ is used to update the classifier fitness as: $F \leftarrow F + \beta(\kappa' - F)$.

III. XCS with Datasets Containing Redundant Attributes

This section analyses effects of datasets containing redundant attributes to accuracy and speed performance of XCS. For this purpose, we ran XCS on eleven UCI datasets [8] shown in table I. These datasets are individually significant to the evaluation of XCS for redundant datasets. We ran XCS with the following parameter setting (as introduced in [2]): N= 6400, β=0.2, α=0.1, ε_0=1, ν=5, θ_{GA}=25, χ=0.8, μ=0.04, θ_{del}=20, δ=0.1, θ_{sub}=200, $P_{\#}$=0.6.

TABLE I: UCI DATASET DESCRIPTION

Dataset	#Attr.	#Inst.	#Class	XCS accuracy
Balloom1	4	20	2	100%
Corral	6	64	2	83.3%
SPECTF	45	267	2	74.07%
Monk3	6	432	2	95%
Balance-scale	4	625	3	81.6%
m-of-n	13	1000	2	72%
Exactly	13	1000	2	69%
Parity5+2	10	1024	2	50%
Car	6	1728	4	98.2%
Led(10%n)	24	2000	10	61.8%
Mushroom	22	8124	2	99.2%
Average				80.38%

#Attr. = number of attributes, #Inst. = number of instances, #Class = number of classes.

The first test sets are small datasets and no noise as *Balloml, Corral, balance-scale* and *SPECTF*. Experiments show that the training time of XCS on these datasets are short and population is small. The second experiments require complex sets in number of instances. We test XCS with larger datasets (the number of instances is greater than or equal to 1000).

These experiments will test the ability of XCS to maintain a large population size in order to represent the classification accurately. By experiments, the explore time and population size of XCS on these datasets increase significantly. The last experiments test XCS on noisy datasets which are *Monk3* (5% noise) and *Led* (10% noise). Datasets containing noise could be preventing XCS from forming accurate generalizations and thereby hinder the search process of the GA within XCS.

Experiments in table I show performances of XCS on original datasets. We can see that XCS performs best on *Balloom1* where 100% data are classified correctly. *Parity5+2* is classified correctly only 50% that is dataset classified worst. However, the notice datasets are *Mushroom* and *Car*. Although these datasets contain a large number of instances, their accuracies are still high. This is because not only the number of attributes and instances affect to performance of XCS, but also there are other factors affecting to performance of XCS such as: complexity of problem and distribution of dataset. This may be our future works.

IV. Approximate Reduct

In this section, we focus on our approach, approximate reduct. The more details of Rough set foundation are referred to [5]. Our combination of feature reduction uses *β*-reduct or approximate reduct with the definition as follows:

Let IS=(U, A, f) be an information system, where U is a finite nonempty set of N objects {x_1, x_2, …, x_N}; A is a finite nonempty set of n attributes (features) {a_1, a_2, …, a_n}; f_a: $U \rightarrow V_a$ for any $a \in A$, where V_a is called the domain of attribute a.

B is called a β-reduct or approximate reduct of conditional attribute set A if B is the minimal subset of A such that $\delta_B^D \geq \beta * \delta_A^D$. Where D={d} is the decision attribute and δ_A^D is the relative dependency degree of A with regard to D [5]. The rough set-based attribute reduction algorithm in our developed approach is given as Fig.1.

In algorithm of Fig. 1, the parameter β, $\beta \in [0, 1]$, is

called the consistency measurement. β represents how consistent the sub-decision table (with respect to the considered subset of attributes) is relative to the original decision table (with respect to the original attribute set). It also reflects the relationship of the approximate reduct and the exact reduct.

For selecting feature subsets from datasets with a lot of redundant attributes, we select the best attributes one by one by using the evaluation criterion of dependency γ, until a reduct is found. A feature subset, good or not, depends on the dependency of *decision D* on that feature subset. To select a strong feature subset, the following selection strategies are used:

- Selecting the features that cause the number of consistent instances increases faster, to obtain the subset of features as small as possible.
- The size of maximal subset in *$POS_R(D)/IND(R, D\})$* should be considered. In general, the more the number of attribute values in a feature in R, the more the number of subsets, and the smaller the size of the maximal subset. Selecting a feature, by which a bigger subset can be acquired, is a way for our purpose.
- Next, we consider the consistent instances. Let *largest $POS_{RU\{a\}}(D)/IND(RU\{a\}, D\})$* denote the size of the maximal subset of the lower approximation of the set *$POS_{RU\{a\}}(D)$*.

```
//step 1
$R = \{\}$;
$x = \gamma_C(D)$;
While ($\gamma_R(D) < \beta * x$)
{
    $\forall a \in C - R$
        Choose $a$ with largest $\gamma_{R \cup \{a\}}(D)$;
        If ties occur, select $a$ with largest
        $POS_{R \cup \{a\}}(D) / IND(\{R \cup \{a\}, D\})$;
        If further ties occur then select $a$ with smallest
        $\left| POS_{R \cup \{a\}}(D) / IND(\{R \cup \{a\}, D\}) \right|$;
    $R = R \cup \{a\}$;
    $U$ = $U$ - $POS_R$ ($D$); //Remove all consistent instances
    $C = C - \{a\}$;
}
//step 2
$\forall\ a \in R$
    If $\gamma_{R-\{a\}}(D) = \gamma_R(D)$ then $R = R - \{a\}$;
Return $R$;
```

Figure 1: Approximate Reduct Algorithm

- When two features have the same performance described above (ties occur), the one that contains a less number of different values will be selected.

- After performing all steps above, we should have a last step to remove all redundant features by re-computing the dependency of decision D on each feature.

V. Combining Rough Set and XCS

A. The Framework

We address problems discussed in section III by combining rough set-based FR approach with XCS learning classifier system as Fig. 2. Feature importance is taken into account through reduct generation. The features in the reduct are considered to be important while other features are considered to be irrelevant. Reduct computation by RS does not require any domain knowledge and the computational complexity is only linear with respect to the number of attributes and instances [5]. After combining the FR method and XCS, the knowledge representation is still the same as that the original one. This form of knowledge representation is easier to understand and more convenient for classifying unseen inputs. Furthermore, since only the features in the reduct are evolved in the rule generation of XCS, the running time is also reduced.

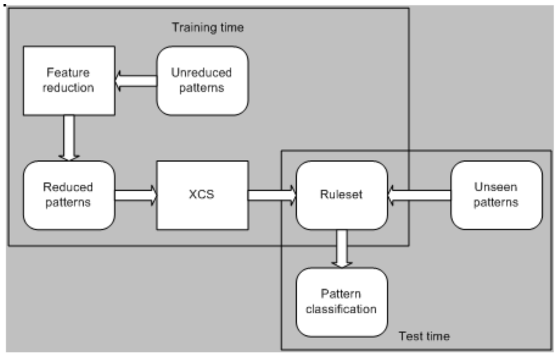

Figure 2: the RS and XCS combination framework

In redundant-datasets classification problems, there are three main benefits from combining RS and XCS. 1) Classification accuracy can be preserved or even improved by removing non-informative features, redundant and noisy instances, 2) Storage requirements are reduced by deleting irrelevant features and redundant instances, and 3) the classification decision response time can be reduced because fewer features and instances will be examined when an unseen instance occurs. As shown in Fig. 2, the system in this paper includes the following modules.

+ Rough Set-based Feature Reduction: Reads a set of feature patterns and outputs another subset with reduced dimensionality, implemented with the Rough set-based Feature Reduction algorithm.

+XCS-based Rule induction: Reads a sequence feature patterns and outputs a set of *if-then* rules connecting features and the implied classes.

B. Experimental Results

1) Attribute reduction using RS

In this section, we test and analyze the feature reduction capability of the rough set-based algorithm proposed in the section IV. The results of this algorithm on 11 datasets described on the previous section are shown on the second and third columns of table II.

```
Step 1: Initialize $Acc = \phi$ and $\beta=1$
Step 2: While ($\beta>0$)
    + Implement Approximate Reduction Algorithm; (Output the reduced dataset RD)
    + Training XCS with RD.
    + Implement test XCS with unseen instances; (Output the current accuracy, $a$)
    $Acc \leftarrow Acc\ U\ \{a\};$
    $\beta = \beta - \lambda;$
Step 3: Find $a^*$, $a^* = max\{a \in Acc\}$; and find the corresponding $\beta^*$ and $R^*$.
Step 4: Output $a^*$, $\beta^*$ and $R^*$.
```

Figure 3: The RS and XCS combination algorithm

The *SPECTF* dataset achieves the greatest attribute reduction, reducing the number to attributes to 3 (i.e. only 1/15 of 45 unreduced attributes), however only one instance is reduced (from 267 instances to 266 instances). With *Mushroom* dataset, when number of attributes is reduced from 22 to 5, the number of instances is reduced significantly from 8124 to 47 instances. This also occurs with *Parity5+2, m-of-n, Exactly* and *Led(10%n)* datasets. On datasets where there is no redundant or noise attribute as *Car* and *Balance-scale*, there is no reduction on any value of β. The remained datasets achieve the attribute reduction and instance reduction slightly.

In Fig. 1, we can see that the larger the value of β, the less number of attributes and instances are reduced. However, our objective is to find the value of β in which accuracy of XCS is highest. We ran the proposed system on 11 datasets with various values of β as algorithm in Fig. 3. We found that the best β value after applying both RS and XCS for these real-life datasets is 0.95. This is why the β value is set at 0.95 in this paper.

2) XCS on reduced dataset

TABLE II: XCS AND RS-XCS PERFORMANCES

DATASET	#Red. Attributes	#Red. Instances	RS-XCS accuracy
Balloom1	2	4	100%
Corral	4	16	85%
SPECTF	3	266	80%
Monk3	3	36	99%
Balance-scale	4	625	81.6%
m-of-n	6	64	76%
Exactly	6	64	74%
Parity5+2	5	32	75%
Car	6	1728	98.2%
Led(10%n)	5	10	75%
Mushroom	5	47	100%
Average			85.80%

#Red. Attributes = Reduct attributes, #Red. Instances = Reduct instances,

Experiments in table II show that after attribute reduction, all the XCS performances were able to preserve or even improve classification accuracy. The most improved classification accuracy is occurred on two datasets *Monk3* and *Led*(10%n). This is because these datasets contain noisy instances and after applying Rough set-based attribute reduction, not only redundant attributes are eliminated but also noisy instances are. Note that since Rough set-XCS specializes in the removal of redundant attributes and noises, it might not improve performance on complex datasets as *Car* and *Balance-scale*. In such settings, we might expect Rough set-XCS to throw out perfectly good data. On average, the classification accuracy after applying the combination of RS and XCS increase by 5.42 percent.

To conclude, when Rough set-XCS is applied, the results of almost datasets are positive. The classification accuracy and speed performance show a notable improvement.

VI. CONCLUSIONS AND FUTURE WORKS

We have introduced a combination of an approximate Rough set-based feature reduction approach and XCS learning classifier system to solve datasets containing redundant attributes. In the FR approach, the concept of a reduct is generalized to an approximate reduct, which makes the reduct computation faster and more flexible. In some situations, the crisp reduct (β=1) is the best subset of features in term of the classification accuracy. In some other situations, the crisp reduct is not the optimal subset of feature; in this paper, the β value is determined to optimize the classification accuracy. The developed approach can remove not only the redundant attributes but also the noise instances. We first analyzed effects of these datasets to performance of XCS. The results show that XCS is quite robust to datasets containing redundant attributes although the speed is slow, especially with large datasets. This is mainly the fact that when implementing classifier tasks, XCS tries to cover all space of problem. The combination of approximate Rough set-based feature reduction and XCS that we proposed makes it possible to address this problem. By this combination, we could further enhance the accuracy; reduce training time and the storage. We tested 11 UCI datasets by XCS first and Rough set-XCS following. The results show that performances of the proposed method are improved significantly in term of accuracy.

There are still some limitations of our developed FR and XCS approaches, which may need to be tackled in our future work: 1) The determination of using dummy variables or nominal categorical variables in rule representation of XCS is empirical and heuristic based during the testing and depends on dataset. 2) The rough set-based FR method works better with symbolic data. The numerical data needs to be discretized before applying FR process. We plan to expand the framework to deal with real-value inputs with multiple classes, and to test the system on other datasets.

ACKNOWLEDGEMENT

This work was sponsored by the AUN/SEED-Net organization. We thank Kreangsak Taemee for making to us aware of Wilson's XCS, to Phaitoon Srinil for pointing out important properties of redundant datasets and to Sornchai Udomthanapong for essential help in guiding during our simulation.

REFERENCES

[1] P.N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining,* Addison Wesley, USA, 2006.

[2] Stewart W. Wilson, "Classifier Fitness Based on Accuracy," *Evolutionary Computation*, Vol.3 (2), pp.149-175, 1995.

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

[4] A. J. Bagnall. and G. C. Cawley, *"Learning classifier systems for data mining: A comparison of XCS with other classifiers for the Forest Cover dataset*", In Proceedings of the IEEE/INNS

International Joint Conference on Artificial Neural Networks (IJCNN-2003), vol. 3, pp. 1802-1807, 2003.

[5] Komorowski, J., Pawlak, Z., Polkowski, L., et. al., "Rough Sets: A Tutorial," in *Rough Fuzzy Hybridization: A New Trend in Decision-Making* (ed. S. K. Pal, A. Skowron), Springer-Verlag, Singapore, pp. 3-98, 1999.

[6] Dominik Slezak, "Approximate Entropy Reducts", *Journal of Fundamenta Informaticae*, vol. 53 (3-4), pp.365 - 390, 2002.

[7] Pawlak, Z., "Rough set theory for intelligent industrial applications", *Proceedings of the Second IEEE International Conference on Manufacturing of Materials*. vol. 1, pp.37-44, 1999.

[8] http://archive.ics.uci.edu/ml/datasets.html.