International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(A)**, January **2012**
ICIC International ©2012 ISSN 1349-4198
pp. 33–47

TO BOUND SEARCH SPACE AND BOOST PERFORMANCE OF LEARNING CLASSIFIER SYSTEMS: A ROUGH SET APPROACH

Farzaneh Shoeleh, Ali Hamzeh and Sattar Hashemi

Department of Computer Science and Engineering and Information Technology
School of Electrical Computer Engineering
Shiraz University
Mollasadra Ave., Shiraz, Iran
{ shoeleh; ali }@cse.shirazu.ac.ir; s_hashemi@shirazu.ac.ir

Received July 2010; revised November 2010

Abstract. *Learning classifier system is a machine learning technique which combines genetic algorithm with the power of the reinforcement learning paradigm. This rule based system has been inspired by the general principle of Darwinian evolution and cognitive learning. XCS, eXtended Classifier System, is currently considered as state-of-the-art learning classifier systems due to its effectiveness in data analysis and its success in applying to varieties of learning problems. It tries to evolve a rule set as a solution, which can cover the whole problem space effectively. Since XCS does not discriminate between class boundary and non-class boundary regions of input space, learning non-class boundaries with more samples might prevent XCS to learn the class boundary regions precisely. This paper explores XCS from this perspective and provides an elegant approach based on rough set theory to deal with this issue. Here, rough set is a mathematical formalism adapted for XCS, which offers our model, i.e., RSXCS, the ability to analyze the problem space and differentiate certain regions from vague ones. The certain regions are first explored by PSU, as an important component of RSXCS, and the rests are directed to other learning components for further rule discovery. To investigate the advantages of RSXCS, quite a lot of experiments across different UCI data sets are conducted. The results show that our approach presents statistically significant improvement in term of classification accuracy compared with its state-of-the-art rival algorithm, i.e., XCS.*

Keywords: Learning classifier systems, XCS, Rough set theory

1. **Introduction.** The basic framework of Learning Classifier System (LCS) was introduced more than 30 years ago by J. Holland in [1]. The LCS framework was reformed to use reinforcement learning techniques such as Q-learning in order to ensure appropriate reward estimation and propagation. LCSs are also known as rule-based evolutionary online learning systems. Each rule based system such as LCS tries to learn a rule set which is able to cover the whole problem space in order to model the environment regularities in classification tasks whose main aim is to predicate the class of incoming samples accurately. In classification tasks, there are three main objectives: high accuracy, comprehensibility and compactness. It is obvious that, in rule based classifiers, achievement of these goals directly depends on defining not only the structures of rules but also the main mechanism which generates and improves rules.

The year 1995 marked as a milestone in LCS researches due to Wilson's flavor of Holland's recipe, the most popular Michigan system called XCS [2]. In Lanzi's view, the effectiveness of XCS as a machine learning paradigm is that *"XCS was the first classifier system to be both general enough to allow applications to several domains and simple enough to allow duplication of the presented results"* [3]. Since 1995, the applicability of XCS has been extended to a wide range of applications, such as computational economics, classification and data mining, autonomous robotics, power distribution network, traffic light control, function approximation/regression tasks and many more [4-7].

Although XCS is a reliable system in realm of data mining (DM) especially in classification tasks, there are still some challenging issues that must be solved. For example, since XCS tries to cover the whole input space for modeling the environment regularities, it needs large population size and learning iterations in dealing with problems that have vast input space with a large number of samples. On the other hand, since XCS does not discriminate between class boundary and non-class boundary regions of input space, learning non-class boundaries with more samples might prevent XCS to learn the class boundary regions more precisely. So, it seems that bounding the search space of XCS to the portion of input space which only contains class boundaries may lead to better prediction of these boundaries. Here, to overcome the mentioned shortcoming and achieve such objective, a rough set based approach is proposed, which manages to bound the search space of XCS in order to have more accurate classifier system.

There are a host of efforts to reduce data sets size by selecting significant features or reducing redundant instances in data mining problems. For example, several researches have been focused on using rough set theory to refine the data set [8,9]. It is worth mentioning that the success of this notion is due to the two main reasons. First, it is able to reveal the underlying information captured inside a data set. Second, rough set data analysis does not need to use information outside the target data set [10]. However, in LCS realm, the only methods which are examined to reduce the size of data sets are feature selection and feature extraction methods [11,12]. Also, to the best of our knowledge, there is no effort to bound the search space of an LCS, like XSC, to the portion of problem space which has efficient information about how to model the decision boundaries.

The goal of this paper is proposing a novel approach to boost the performance of XCS in classification tasks by refining the train data sets of XCS and bounding its search space to the class boundary regions that have more knowledge about the regularities of problems. This new extension of XCS is called RSXCS which has a new component named Prototype Selection Unit (PSU). In PSU, genetic algorithm is used to prepare the basic element of rough set theory, i.e., the indiscernibility relation. When such relation is defined, the fundamentals of classic rough set theory are used to differentiate certain regions from vague ones. Then, the search space of XCS would be bounded to the class boundary regions containing classes' boundaries to model the problem regularities precisely.

The rest of this paper is organized as follows: Section 2 reviews some important work on applying XCS on DM problems and discusses the related researches for using rough set in purpose of instance selection; Section 3 describes the proposed method; the experimental results are given in Section 4; Section 5 provides some concluding remarks.

2. Related Work.

XCS was firstly proposed by Wilson in 1995 [2] and soon after researchers in XCS realm had tried to demonstrate the capabilities of XCS for DM problems. As an early attempt, XCSR and XCSI with interval based representation were applied to real valued problems [13,14]. Bernado et al. [15] described an experimental comparison of XCS with seven well-known learning algorithms. The obtained results showed the effectiveness of XCS in classification tasks. Also, a bit later, Bacardit and Butz [16] investigated and compared the performance of XCS and GAssist, a Pittsburgh-style LCS, on several interesting data sets. The reported results and discussion showed that both systems are suitable for DM applications. From then on, most researchers who intend to apply XCS on DM problems use the same parameter settings suggested in [16]. Consequently, it can be claimed that XCS have a significant impact on DM field [5,17]. However, there are some issues which must be handled. One of these issues is dealing with large data sets. There are two main approaches for purpose of data reduction: (1) Dimensionality reduction while preserving as much of the class discriminatory information as possible. In this approach, the size of data set decreased by reducing the number of columns in a data set and (2) prototype selection and/or instance selection [18,19] which mean reducing the number of rows in a data set.

There are many researches in order to dimensionality reduction with different measures [20]. In [11], to solve this problem, a combination of feature selection technique based on the rough set and XCS was proposed. There, the purpose of using rough set theory is to identify the most significant attributes and eliminate the irrelevant ones. The experiments denoted positive results. The developed approach was tested only on nominal problems because its rough set based feature selection method is designed for symbolic data.

One of the main problems faced in DM realm is how to deal with huge amount of information. To tackle this problem, many current researches focus on scaling down DM algorithms using various instance selection and prototype selection algorithms [21-25]. Cano et al. [23] addressed the analysis of some representative EA models for data reduction. Since rough set theory is a mathematical tool for data analysis, several approaches have been developed based on this theory [24-29]. In [25], a rough set based method is proposed to reduce the scale of SVM training set to the non boundary instances.

In this paper, a new approach is proposed to incorporate the idea of rough set into XCS. By this mechanism, XCS just focus on subspaces of problems which contain class boundaries. As there is a certain amount of risk in classifying the instances of such subspaces, precisely modeling these subspaces are more important to achieve higher performance in classification problems.

3. **Proposed Approach.**

In spite of the fact that XCS is a reliable system to handle data set with large amount of samples, it seems that bounding the XCS search space to class boundary regions that consist of more than one class instances, may lead better performance in classification problems. In other words, XCS does not discriminate between class boundary and non-class boundary regions. So, if there is an algorithm to detect this discrimination and model the non-class boundary regions, XCS can make its main effort to model classes' boundaries more precisely. In the literature, there exist many invaluable efforts to select instances for classification task [24,25,27]. A possible approach to address this problem is using a mathematical formalism such as rough set theory. In this paper, a new model named RSXCS is proposed. RSXCS tries to reduce the training data set of XCS and bound its search space to the boundary region instances by using a rough set based technique.

The first step is to extract the boundary regions for each class. As mentioned before, rough set has an ability to divide the search space into the boundary regions and non-boundary regions which are known as positive region. It is worth mentioning that instances which are located in the positive region have same label. Since the rough set theory requires indiscernibility relation to define an approximation space, as a first step a mechanism is needed to define such a relation. Here, partitioning the search space plays the role of indiscernibility relation and genetic algorithm is used as a search technique to find the best way of partitioning the input space. The partitioning of concern favors the larger positive regions with more certain instances. Once the partitioning has been found, it is used as an indiscernibility relation, what in turn gives the rough set reducer block the opportunity to distinguish between the positive regions and boundary regions. After finding these positive regions, their characteristics are encapsulated in certain rule forms and the instances matched with these rules are going to be eliminated from the training data set. These rules are stored to determine whether an incoming unseen data is located in the positive or boundary regions. After this reduction process, the data set comprise only the boundary data. Thus, it is possible to say that the obtained refined data set shows some portion of the problem space where only classes' boundaries are located and consequently the decision about samples' label is relatively hard. Finally, XCS is applied to the reduced data set to model the classes' boundaries with its evolved rule set. XCS generates a set of if-then rules which are used to predict the label of boundary data. As a result of this approach, two separate rule sets are obtained, one contains the certain rules for positive regions and the other is produced by XCS for boundary regions.

Obviously, applying such method enforces XCS to focus only on the boundary regions, and hence, to provide better performance in classification tasks. The overall architecture of our model, i.e., RSXCS, is illustrated in Figure 1. RSXCS consists of two main units as follows: 1) Prototype selection unit (PSU): The whole process of reducing training data set instances and bounding the search space of XCS is done by this component. Therefore, it consists of a genetic algorithm and a rough set data reducer block. It is worth mentioning that the output of this unit would be a certain rule set and a refined training data set. 2) XCS: XCS is applied to the boundary region data supported by the previous block and the class boundaries are modeled by its rule set.

During the test phase, to predict the class label of an unseen data, first the certain rules in PSU are examined. If one of them can match this data, it means that it pertains to the positive region represented by matching rule. So, the corresponding rule suggests the label of this unseen data. Otherwise, it can be concluded that this unseen data is located in boundary regions so the rule set extracted by XCS has an ability to predict its class label. According to the type of given problem (real valued problem or nominal problem); there are different possible mechanisms to define PSU components. In the following, we will describe the proposed approach for dealing with real valued problems in the next subsection and nominal problems in Subsection 3.2. After that, in Subsection 3.3, the rule representation technique used for XCS classifier is described.

3.1. **Real valued problem.** For the sake of partitioning the input space of real valued problems, the most used method is discretization. Genetic algorithm searches among all possible candidate discretization points and finds the best of all that discriminates large positive regions with more certain instances. So, each chromosome consists of genes which hold a real number showing the distance between two consecutive discretization points.

In evolutionary algorithms, fitness function has a critical role since it directs the search process to find the best solution by assigning quality measure to genotypes. Here, fitness function is defined based on classical rough set theory. The fitness function of concern favors the larger positive regions with more certain instances. The quality of each chromosome is evaluated as follows:

$$\text{Objective model: } \max\left(\sum_{i=1}^{n} \neq p_i * isPR(p_i)\right) \tag{1}$$

$$\text{Subject to: } \min\left(\sum_{i=1}^{n} isPR(p_i)\right) \tag{2}$$

where P is the number of partitions produced by candidate chromosome and consisting of more than T instances. $\neq p_i$ indicates the number of instances located in i'th partition (p_i), and $isPR(p_i)$ returns '1' if all instances in $\neq p_i$ is in the same class. Otherwise, it returns '0'. In other words, $isPR(p_i)$ determines whether $\neq p_i$ is a positive region or not.

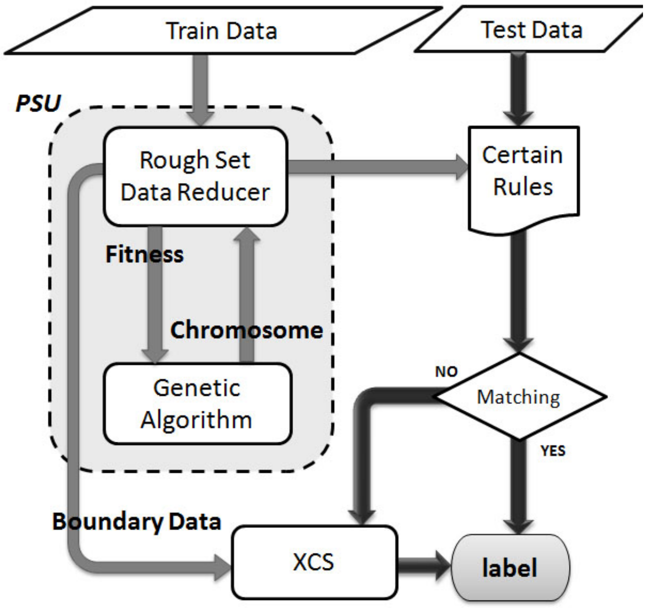

FIGURE 1. The overall architecture of RSXCS. In typical learning iteration, PSU uses genetic algorithm and fundamentals of rough set theory to bound the search space of XCS by reducing the instances of positive regions. Certain rules produced to model these certain regions. The instances of boundary regions are directed to XCS to model the classes' boundaries precisely. In the test phase, firstly the unseen data is examined by certain rules. Then, if there is no rule to match it, the concept of this data would be predicted by XCS's rule set.

The objective model consists of two conflicted parts and the genetic algorithm must optimize these two terms simultaneously. There have been many approaches to multi-objective optimization using evolutionary algorithms. Here, the method suggested by Goldberg in [30] is used. He suggested the use of fitness based on dominance rather than on absolute objective scores, coupled with niching methods to preserve diversity.

For ranking a solution based on dominance, the number of solutions which are able to dominate corresponding solution is counted. One solution is said to dominate the other if it has higher score for all objectives. A solution is called non-dominated if it is not dominated by any other. The set of all non-dominated solution is known as Pareto set or Pareto front. Usually, for multi-objective problem with conflicting objectives, there exists no single solution that dominates all others. In such problems, the goal is to find a non-dominated solution which lies in Pareto front. Here, the genetic algorithm is applied to find such Pareto front, after that, the best chromosome is the one that can reduce the size of train data set more than the others.

The types of crossover and mutation operators play a major role in performance of the GA optimization. Here, whole arithmetic recombination which is the most commonly used operator for real valued representation and uniform mutation are used.

In the following, there is an example to understand how the evolved solution obtained from applying GA can be used as an indiscernibility relation in rough set data reducer block and which certain rules would be generated in before eliminating positive regions instances. This example is about applying PSU on "Tao" [31] problem with 1800 samples. Figure 2 shows the process of PSU in order to find boundary and positive regions.

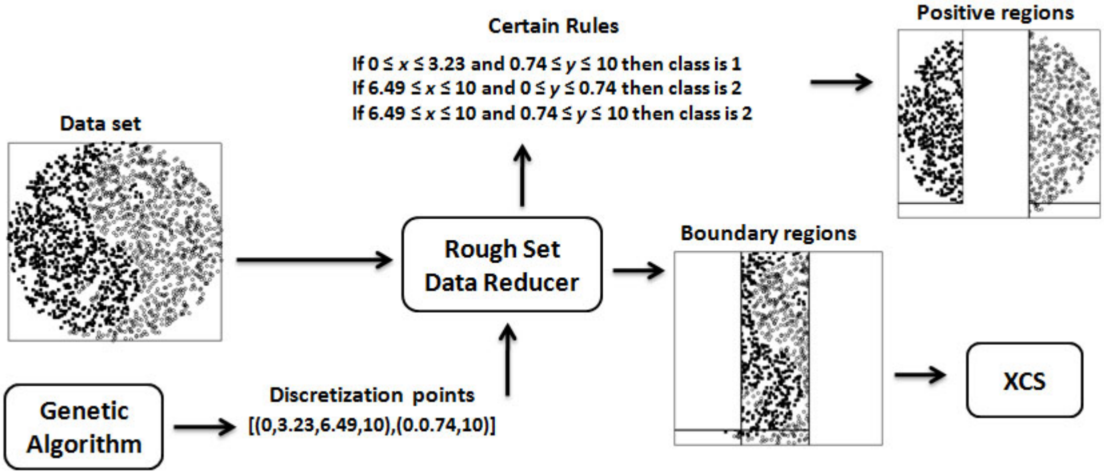
Certain Rules
If 0 ≤ x ≤ 3.23 and 0.74 ≤ y ≤ 10 then class is 1
If 6.49 ≤ x ≤ 10 and 0 ≤ y ≤ 0.74 then class is 2
If 6.49 ≤ x ≤ 10 and 0.74 ≤ y ≤ 10 then class is 2
Positive regions
Data set
Rough Set Data Reducer
Boundary regions
Genetic Algorithm
Discretization points
[(0,3.23,6.49,10),(0.0.74,10)]
XCS

Figure 2. The process of PSU to find reduced training data set for XCS in "Tao" problem. At first, the positive (with 1076 instances) and boundary regions (with 724 instances) are differentiated by Rough Set Data Reducer block, then three certain rules are produced to present the positive regions and XCS is applying to the boundary regions to model them.

3.2. **Nominal problems.** Recall that the first step in applying the proposed approach is determining the indiscernibility relation which is needed to utilize rough set theory. In nominal problems, such relation can be defined by selecting subset of features and considering the possible value of these selected features in order to partition the input space. Like real valued problems, here a genetic algorithm is used to find subset of features while preserving as much of the class discriminatory information as possible.

In nominal problems, the used genetic algorithm has binary representation, two-point crossover, bit flip mutation, tournament selection and elitism survival selection. Chromosomes are represented by binary bit strings with length n, where n is the total number of input attribute of the problem. Every gene indicates an attribute; the value '1' shows the corresponding attribute is selected while '0' not selected. Here, the objective model to measure the quality of chromosomes is defined as follows:

$$\text{Objective model: } \max\left(\frac{|POS_P(Q)|}{|U|}\right) \tag{3}$$

$$\text{Subject to: } \min\left(\frac{|C|-|B|}{|C|}\right) \tag{4}$$

where B is a feature subset suggested by the candidate chromosome so in rough set point of view the indiscernibility relation is $P = (U, IND(B))$. So, $\frac{|POS_P(Q)|}{|U|}$ shows the classification quality of the selected feature subset (B) relative to decision Q (Q indicates the class of instances). It must be mentioned that to calculate $|POS_P(Q)|$, only the equivalence classes having more than T instances are attended. As $|.|$ is the cardinality of a set, $|C|$ is the total number of features and $|B|$ is the number of selected features in corresponding chromosome. In sum, this objective model consists of two conflicted terms: (1) $\frac{|POS_P(Q)|}{|U|}$ shows the classification quality by considering B as available information. This term forces GA to find a subset of features which can maximize the number of instances located in positive regions. (2) $\frac{|C|-|B|}{|C|}$ indicates the length of selected subset, which prevent GA to select all features. Genetic algorithm tries to find a solution that optimizes these two terms simultaneously. So, as it is described in the previous subsection, the common technique for solving such multiobjective problem by GA is finding Parento Front by ranking solutions according to the number of their dominating solutions.

After the best feature subset is founded by GA and positive regions are detected, the certain rule is extracted and the instances that can be classified certainly with them are eliminated. The rests are directed to XCS as training set in order to model the class boundaries located in boundary regions. All these processes are shown in Figure 3 for the "Car" data set [32].

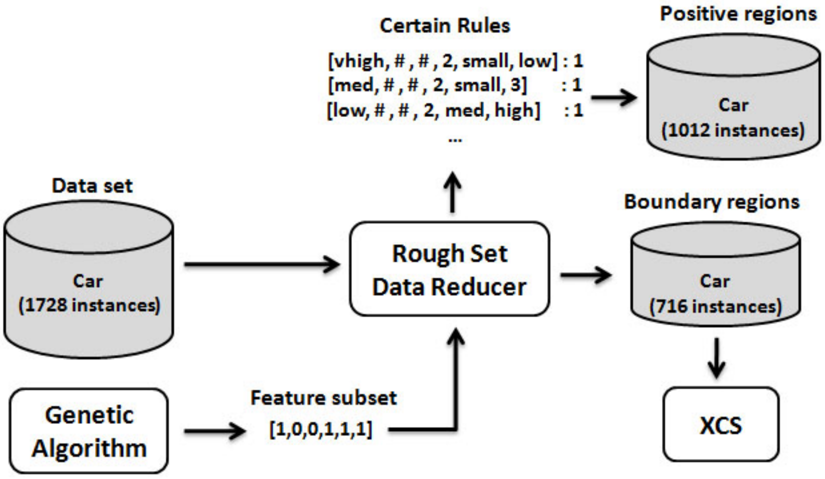

Figure 3. The process of PSU to find reduced training data set for XCS in "Car" problem. At first, the positive (with 1012 instances) and boundary regions (with 716 instances) are differentiated by Rough Set Data Reducer block, then certain rule set is produced to present the positive regions and XCS is applying to boundary regions to model them.

3.3. **XCS representation.** Here, XCS with interval based representation [13,14] is used. For each dimension, an interval is presented in the condition part of classifiers. We use Lower-Upper Bound Representation [14] which is the most common method in literatures for dealing with real valued inputs. In this method, an interval predicate $[p_i, q_i)_p$ is encoded as a tuple $(l_i, u_i)_g$ where l_i, $u_i \in$ R. A rule forms as if $x_1 \in [l_1, u_1]\Lambda \dots \Lambda x_n \in [l_n, u_n]$ then class where n indicates the number of input features of the problem. An input instance $x = (x_1, x_2, \dots, x_n)$ is matched by a rule if and only if $\forall l_i < x_i < u_i$. In nominal problems, the condition part of XCS classifiers consists of n genes. Each gene is related to each problem dimension and can have an allele between 0 to m or $\neq$; where m is the number of possible values for the corresponding dimension and $\neq$ is do not care symbol which indicates the value of related dimension can be ignored.

4. Experimental Results.

4.1. **Experiments design.** To assess the proposed method, several standard data sets including Tao, Car, Sign, Shuttle, Solar flare, Monk's problem 3, Pima, King+ Rook versus King+ Pawn (kr vs. kp), page blocks and Nursery are selected. Nine of these data sets are derived from UCI Repository database [32] and one of them named Tao is a synthetic data set that was introduced in [31]. Table 1 summarizes the most important properties of selected data sets.

As it is clear in Table 1, a wide spectrum of data sets is selected in order to find the advantages and disadvantages of the proposed method. For instance, "Tao" is a low

Table 1. Important properties of selected data sets. The columns describe: the name of the data set, the number of instances, the total number of features, the number of real-valued features, the number of nominal features and the number of classes.

Data Set	≠ Instances	≠ Attribute	≠ Real-valued attr.	≠ Nominal attr.	≠ Classes
Tao	2000	2	2	0	2
Car	1728	6	0	6	4
Sign	12546	8	8	0	3
Shuttle	43500+14500	9	9	0	7
Solar-flare 2	1066	12	0	12	6
Monk's problem 3	122+432	6	0	6	2
Pima	768	8	8	0	2
kr vs. kp	3196	36	0	36	2
Page-blocks	5473	10	10	0	5
Nursery	12960	8	0	8	5

dimensional data set while "kr vs. kp" is high dimensional ones. "Shuttle" consists of several classes whereas "Pima", "Tao", "Monk's problem 3" and "kr vs. kp" include just two classes. "Monk's problem 3" is small data set with low number of samples; in contrast, "Shuttle" is a large data set with 43500 instances as the training samples and 14500 instances as the test samples. "Tao" is a balanced data set whereas "Car" is imbalanced.

4.2. **Experimental setting.** To show the effectiveness of PSU, the proposed model namely RSXCS is compared with XCS. To obtain reliable performance for both classifiers (RSXCS and XCS), 10-folded cross-validation method is used. The average and standard deviation of the classification accuracy over 30 independent runs of 10-folded cross-validation are calculated. To compare the differences in the performances of both classifiers, one-tailed pairwise t-test is used to check whether the difference between two classifiers' results over the selected data sets is statistically significant or not. The null hypothesis of this test is that the average performances of XCS and RSXCS over 30 independent runs are same against the alternative hypothesis that the average performances are not equal.

For nominal problems, XCS was configured as follows (see [2] for notation details): N = 6400, $P_{\neq} = 0.6$, $\beta = 0.2$, $\alpha = 0.1$, $\epsilon_0 = 1$, $\mu = 5$, $\theta_{GA} = 25$, $\chi = 0.8$, two point crossover, $\tau = 0.4$, $\mu = 0.04$, $\delta = 0.1$, $\theta_{del} = 20$, $\theta_{sub} = 200$ (like XCS parameter settings in [11]). And for real valued problems, the parameter settings for XCS are the same and $m_0 = 0.2$ (as used in the literature, [16]). The genetic algorithm in PSU is run with following parameter settings: maximum number of iterations = 1000, population size = 1000, T = 10, intermediate crossover and two-point crossover with $\chi = 0.8$ is used in real valued and nominal problems respectively, uniform mutation with $\mu = 0.2$, tournament selection with $\tau = 0.6$ and survival selection $(\mu + \chi)$ where $\chi = 100$.

As the PSU component of RSXCS analyzes the input space before applying XCS, it is worth mentioning that some learning iterations have been done in PSU to find the positive regions and encapsulate their characteristics in certain rules. How many learning iterations are done in PSU directly depends on the two main parameters of the GA, namely the maximum number of iterations and the population size. Since the population size in PSU is set to 1000 which is so less than N in XCS, it can counteract the effect of the amount of this parameter in comparing RSXCS with XCS. Consequently, the only parameter that affects the comparison of both learning classifier systems is the number of learning iterations. To make a fair comparison, the number of learning iterations for both XCS and RSXCS is set to 101,000. Please note that 1000 out of RSXCS's iterations are devoted to PSU to discriminate the boundary and positive regions. By using such parameter settings for both RSXCS and XCS, we expect that they have nearly same computation power. Of course, RSXCS might have lower elapsed time to achieve the better solution because extracting certain rules by PSU has lower computation power and takes less time in contrast with XCS which is made to model this kind of regions.

4.3. **Results.** The results of applying RSXCS and XCS on the introduced different data sets are presented in Table 2. This table highlights the difference of RSXCS and XCS in terms of their average performance with the standard deviation in both train and test phases.

Table 2. The results show the average performance (with its standard deviation) of train and test phases of RSXCS and XCS using 10-folded cross-validation. Moreover, the results of applying pairwise t-test are also provided.

Data Set	RSXCS		XCS		Pairwise t-test p-value	
	train	test	train	test	train	test
Tao	93.58%(5.12)	92.84%(5.32)	81.72%(13.24)	81.32%(13.62)	$5.28 * 10^{-23}$	$9.68 * 10^{-23}$
Car	99.43%(0.22)	96.70%(1.45)	95.87%(0.89)	93.08%(2.18)	$1.62 * 10^{-57}$	$6.12 * 10^{-41}$
Sign	76.43%(1.11)	75.53%(1.46)	75.63%(1.01)	74.84%(1.49)	$6.02 * 10^{-15}$	$1.03 * 10^{-09}$
Shuttle	99.76%(0.08)	99.77%(0.07)	99.56%(0.18)	99.59% 0.18)	$6.79 * 10^{-07}$	$3.03 * 10^{-06}$
Solar-flare 2	80.50%(1.44)	73.42%(4.51)	78.90%(1.01)	73.68%(4.27)	$4.69 * 10^{-22}$	$9.06 * 10^{-01}$
Monk's problem 3	100.0%(0.00)	93.84%(0.39)	100.0%(0.00)	93.39%(0.62)	–	$6.99 * 10^{-04}$
Pima	98.38%(0.65)	71.99%(5.06)	97.27%(0.78)	71.70%(5.34)	$4.25 * 10^{-12}$	$2.83 * 10^{-01}$
kr vs. kp	99.97%(0.01)	99.49%(0.10)	99.86%(0.02)	99.50%(0.09)	$1.28 * 10^{-25}$	$6.18 * 10^{-01}$
Page-blocks	96.43%(0.39)	95.67%(0.96)	95.50%(0.48)	94.94%(1.08)	$7.83 * 10^{-33}$	$7.54 * 10^{-20}$
Nursery	98.85%(0.15)	97.86%(0.48)	93.80%(0.35)	93.45 %(0.74)	$2.94 * 10^{-87}$	$2.80 * 10^{-74}$

In Table 2, the average performance and the standard deviation over 10-folded cross-validation on 30 independent runs are shown. As the results show, the proposed model may have more or equal performance in comparison with XCS. So, to test whether our results are statistically significant better or not, the one-tailed pairwise t-test with 5% significance level ($\alpha = 0.05$) is performed. According to the *p-values* obtained by applying t-test, RSXCS can achieve better performance over almost all data sets in both train and test phases. On average, RSXCS has 2.5% performance improvement in the train phase and 2.2% performance improvement in the test phase. In addition, according to the obtained variances, it can be concluded that a more stable and higher performance is acquired by RSXCS. In the cases that pairwise t-test is rejected (Pima, kr vs. kp and Solar flare), three pairwise t-test with three different alternative hypotheses are performed; (1) *'right-tail test'* which indicates average performances of RSXCS is greater than average performances of applying XCS, (2) *'two-tailed test'* that average performances of RSXCS and XCS are not equal and (3) *'left-tail test'* that average performances of RSXCS is less than average performances of XCS. These t-tests are performed to verify that RSXCS and XCS perform the same or RSXCS has significantly lower performance and consequently PSU cannot help to boost XCS performance by bounding its search space. The results of applying these pairwise t-tests are shown in Table 3.

Table 3. The Results of applying pairwise t-test with three different alternative hypotheses on average performances of both systems where 'right-tail', 'two-tailed' and 'left-tailed' hypotheses indicate that average performance of RSXCS is greater than, equal to, less than XCS's performance, alternatively.

Data Set	Pairwise t-test p-value					
	right-tail test		two-tailed test		left-tail test	
	train	test	train	test	train	test
Pima	$4.252 * 10^{-12}$	$2.830 * 10^{-01}$	$8.504 * 10^{-12}$	$5.662 * 10^{-01}$	≈ 1	$7.169 * 10^{-01}$
kr vs. kp	$1.277 * 10^{-25}$	$6.180 * 10^{-01}$	$2.555 * 10^{-25}$	$7.638 * 10^{-01}$	≈ 1	$3.819 * 10^{-01}$
Solar-flare 2	$4.694 * 10^{-22}$	$9.065 * 10^{-01}$	$9.839 * 10^{-22}$	$1.869 * 10^{-01}$	≈ 1	$9.437 * 10^{-02}$

As the *p-values* presented in Table 3 are not greater than confidence level of applied pairwise t-tests (0.05) and also according to the results presented in Table 2; it is possible to say that RSXCS can reach the same as or even better performance than XCS over the 95% confidence level in all introduced data sets. So, the results given in Tables 2 and 3 confirm the hypothesis that applying PSU before XCS to distinct positive and boundary regions in RSXCS is useful to solve a given classification problem with better performance.

As mentioned before, the proposed model is going to discriminate between positive and boundary regions and make XCS focus just on boundary regions which contain classes' boundaries. It seems that the proposed approach helps XCS to make its main efforts to cover the boundary regions and predict the classes' boundaries more precisely. To this hypothesis, Table 4 presents the train and test performance of applying RSXCS and XCS on boundary data. Like Table 3, Table 5 shows the results of applying three pairwise t-tests in the cases that their *p-values* of this test in Table 4 are not less than 0.05.

Table 4. The average and standard deviation of train and test performances of RSXCS and XCS on boundary data. Additionally, results of applying pairwise t-test are provided.

Data Set	RSXCS		XCS		Pairwise t-test p-value	
	train	test	train	test	train	test
Tao	84.14%(12.57)	82.47%(12.99)	70.52%(12.58)	69.97%(13.32)	$1.47 * 10^{-18}$	$9.17 * 10^{-16}$
Car	98.62%(0.54)	92.06%(3.43)	89.90%(2.18)	83.37%(5.03)	$1.29 * 10^{-57}$	$3.08 * 10^{-40}$
Sign	72.46%(1.36)	71.69%(1.72)	71.98%(1.22)	71.31%(1.78)	$1.79 * 10^{-06}$	$1.37 * 10^{-03}$
Shuttle	99.25%(0.19)	99.28%(0.20)	98.73%(0.67)	98.83%(0.68)	$7.45 * 10^{-05}$	$4.34 * 10^{-04}$
Solar-flare 2	65.25%(2.54)	52.71%(6.92)	62.39 %(1.75)	53.16%(6.52)	$4.34 * 10^{-22}$	$8.90 * 10^{-01}$
Monk's problem 3	100.0%(0.00)	92.08%(0.49)	100.0%(0.00)	91.68%(0.76)	–	$8.81 * 10^{-03}$
Pima	98.07%(0.77)	67.98%(5.80)	96.74%(0.93)	67.98%(6.02)	$6.38 * 10^{-12}$	$3.98 * 10^{-01}$
kr vs. kp	99.91%(0.02)	98.58%(0.32)	99.55%(0.06)	98.60%(0.29)	$2.03 * 10^{-25}$	$5.81 * 10^{-01}$
Page-blocks	94.24%(0.70)	93.12%(1.61)	92.74%(1.16)	91.93%(2.03)	$3.76 * 10^{-20}$	$1.97 * 10^{-13}$
Nursery	95.54%(0.58)	94.67%(1.59)	81.00 %(1.14)	82.67 %(2.32)	$6.65 * 10^{-85}$	$5.83 * 10^{-69}$

As it was shown in Table 4, on average, PSU can help XCS (in RSXCS) to achieve 4.4% performance improvement in the train phase and 3.5% performance improvement in the test phase. So, the results in Tables 4 and 5 justify that the existence of PSU can improve performance of XCS. Consequently as results show, eliminating the positive regions is useful to model the classes' boundaries precisely since XCS is made to focus just on the portion of problem space where contain the decision boundaries.

TABLE 5. Results of applying pairwise t-test with three different alternative hypotheses on average performances of both systems in boundary regions where 'right-tail', 'two-tailed' and 'left-tailed' hypotheses indicate that average performance of RSXCS is greater than, equal to, less than XCS's performance, alternatively.

Data Set	Pairwise t-test p-value					
	right-tail test		two-tailed test		left-tail test	
	train	test	train	test	train	test
Pima	$6.383 * 10^{-12}$	$3.979 * 10^{-01}$	$1.277 * 10^{-11}$	$7.956 * 10^{-01}$	≈ 1	$6.002 * 10^{-01}$
kr vs. kp	$2.029 * 10^{-25}$	$5.807 * 10^{-01}$	$4.059 * 10^{-25}$	$8.384 * 10^{-01}$	≈ 1	$4.192 * 10^{-01}$
Solar-flare 2	$4.340 * 10^{-22}$	$8.903 * 10^{-01}$	$8.680 * 10^{-22}$	$2.193 * 10^{-01}$	≈ 1	$1.096 * 10^{-02}$

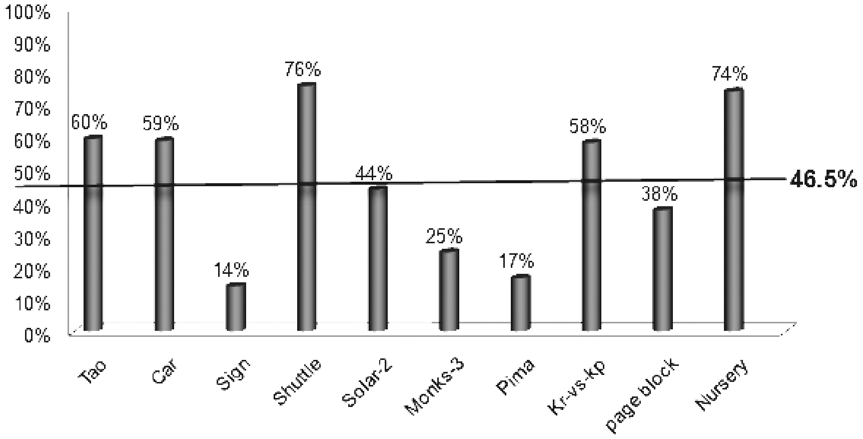

FIGURE 4. Percentages of selected instances in bounding the search space of XCS by PSU component on various data sets

Figure 4 depictes the percentages of selected instances for elimination in PSU on the introduced data sets. As Figure 4 illustrates, PSU can reduce the size of data sets effectively in particular for the problems which have larger data sets in turn bounding the search space to the regions which are more important for classification problems. For example, in Shuttle data set which has larger training data set, the PSU component of RSXCS reduces 76% of training data set of XCS without any loss of performance.

4.4. **Discussions.** In order to find why PSU leads better performance in RSXCS, it is necessary to recall that XCS is designed to evolve a complete, maximally accurate and maximally general rule set as a solution to the problem. XCS can achieve these goals by its evolution pressures [33]: (1) set pressure, which quantifies Wilson's generalization hypothesis [34], (2) mutation pressure, which shows the influence of mutation on specificity, (3) deletion pressure, which quantifies additional deletion influences, (4) fitness pressure, which qualifies the exact influence of accuracy based fitness and (5) subsumption pressure, which qualifies the subsumption deletion mechanism influences. The set pressure pushes classifier population, [P], toward more general classifiers and fitness pressure pushes [P] toward more accurate ones. The basic idea behind the set pressure is that XCS reproduces classifiers in action sets, [A], whereas it deletes classifiers from [P]. So, it can be said that the set pressure is a combination of niche based reproduction which is produced by selection pressure in GA applied in [A] with population wide deletion which is produced by applying deletion in [P]. The probability of a classifier being selected as a parent to reproduce new offspring depends on its fitness also the probability of a classifier being deleted from [P] depends on its fitness. Recall that the fitness value of each classifier in [A] is updated with respect to its relative accuracy which is derived from the reward prediction error. So, the classifiers which has larger prediction error would has high chance to be deleted whereas the classifier with prediction error close to zero would be more selected as being parent in GA. The reward prediction $cl.R$ of a classifier and prediction error $cl.\epsilon$ can be approximated by the following estimate:

$$cl.R \approx \frac{\sum_{s|\text{cl.C matches s}} p(s)p(cl.A|s)p(s,cl.A)}{\sum_{s|\text{cl.C matches s}} p(s)p(cl.A|s)} \tag{5}$$

$$cl.\epsilon \approx \frac{\sum_{s|\text{cl.C matches s}} p(s)p(cl.A|s)|cl.R - p(s,cl.A)|}{\sum_{s|\text{cl.C matches s}} p(s)p(cl.A|s)} \tag{6}$$

In two-class classification problem, the reward of 1000 is usually provided if the chosen class label was correct and 0 otherwise, Consider p_c as the probability that a particular classifier predicts class label correctly. Therefore, the reward prediction $cl.R$ of classifier cl is $1000p_c$ and its prediction error $cl.\epsilon$ is equal to $2000p_c(1-p_c)$. To have better elaboration on the goodness of PSU existence, consider the example presented in Figure 5. In this figure, two indicated classifiers (cl_1, cl_2) have the same generality since they have the same coverage area but their centers are located in different regions; one of them, cl_1, is placed in the positive region which is delineated in gray color and the other, cl_2, is located in boundary region. The class label suggested by cl_1 is the first class (○) and cl_2 suggests the second one (■). As shown, both classifiers can cover 30 instances; cl_1 covers 27 instances of the first class and 3 instances of the second one, so $cl_1.R \approx 1000 * \left(\frac{27}{30}\right) = 900$ and $cl_1.\epsilon \approx 2000 * \left(\frac{27*3}{30*30}\right) = 180$. On the other hand, 14 of 30 instances in cl_2 belongs to the first class and 16 of them belongs to the second class, so $cl_2.R \approx 1000 * \frac{16}{30} = 533.33$ and $cl_2.\epsilon \approx 2000 * \frac{16*14}{30*30} = 497.77$. Thus, cl_1 which its center is located in positive region is more accurate than cl_2 placed in boundary region.

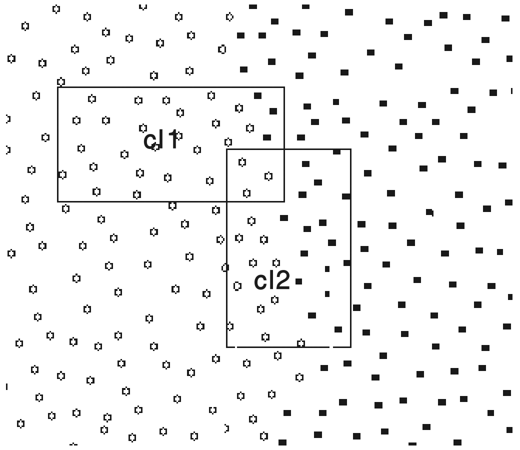

FIGURE 5. An elaboration on why separating problem space into positive and boundary regions can boost the performance of XCS. Suppose cl_1 is a classifier that its center is located in the certain region and cl_2 is a classifier with high overlapping coverage area with boundary regions. As cl_1 has better accuracy and consequently better fitness, it has more chance to participate in GA process and be selected as a parent in the selection mechanism. On the other hand, cl_2 has more chance to be deleted in the deletion mechanism because of its low accuracy.

Since the fitness of each classifier in XCS is calculated according to its relative accuracy which is directly comes from the prediction error of the corresponding classifier, the fitness of cl_1 would be better due to its lower prediction error. Consequently, the classifier with better fitness value cl_1 has higher chance to win in selection competition whereas cl_2 has more chance to be deleted from [P] in deletion mechanism due to its lower fitness value. By and large, it can be concluded that classifiers which their coverage area have higher overlap with positive regions would participate in GA competition in XCS more and consequently their genetic material would be more reproduced.

To recapitulate, by elimination of such positive regions, the discovery component in XCS is stimulated to focus on the boundary regions. Recall that the responsibility of this component is to produce new rules and evolve existing ones. Therefore, in comparison to XCS without applying PSU, in RSXCS, the XCS rules which can cover these regions, would have higher chance to participate in GA competition, and consequently, their genetic material would be more reproduced. So, when the learning phase is terminated, the boundary regions would be covered with more accurate rule set and also the classes boundaries would be modeled precisely. In contrast, XCS without PSU tries to cover whole problem space including positive and boundary regions. So, boundary rules, like cl_2, have lower chance for getting evolve during GA competition. Therefore, using PSU and reducing the training data set to the boundary regions would result in statistically significant improvement over XCS; one may note that the results presented in this section clearly approve this statements.

It must be mentioned that to the best of our knowledge, this is almost the first study to boost the performance of XCS by bounding its search space using a rough set based approach. As mentioned before, in RSXCS two separate but similar methods (one for nominal problems and the other is designed for real valued problems) are proposed to find the indiscernibility relation used in rough set data redactor block of PSU. To make further improvement, our approach can be extended to be applicable on problems with not only real valued or nominal attribute but also mixed attributes.

5. Conclusions and Future Work.

In this work, a classifier named RSXCS consisting of a rough set based preprocessor named PSU and XCS is proposed. PSU is designed to discriminate between class boundary and non-class boundary regions. It uses genetic algorithm and rough set theory to determine the boundary regions and find the instances which can be eliminated from training set of XCS. Genetic algorithm is used to define an indiscernibility relation. Then, rough set data reducer block uses this relation to partition the problem space and differentiate boundary regions from non-boundary ones. The main advantage of proposed model is bounding the search space of XCS and enforces it to focus only on the boundary regions. As there is a certain amount of risk in classifying the instances of such regions, doubtless modeling these regions is more important to achieve higher performance in classification problems. In proposed model, the boundary regions would be covered with more accurate rule set and also the classes boundaries would be modeled precisely. As experimental results show, RSXCS presents statistically significant improvement in term of classification accuracy compared to XCS in both nominal and real valued problems.

The bottleneck of the introduced method is to deal with noisy data. To overcome this shortcoming, as the future work, it might be useful to change the mechanism of determining positive region in PSU in order to consider noise ratio. Another way which might be suitable is to use fuzzy rough set instead of classical one, which has an institutive ability to handle noisy data. Moreover, we intend to evaluate the proposed method for data sets containing both nominal and numeric attributes jointly.

REFERENCES

[1] J. Holland, Adaptation, in *Progress in Theoretical Biology*, R. Rosen and F. Snell (eds.), New York, Academic Press, 1976.

[2] S. W. Wilson, Classifier fitness based on accuracy, *Evolutionary Computation*, vol.3, no.2, pp.149-175, 1995.

[3] P. L. Lanzi, Learning classifier systems: Then and now, *Evolutionary Intelligence*, vol.1, pp.63-82, 2008.

[4] L. Bull and T. Kovacs, *Foundations of Learning Classifier Systems (Studies in Fuzziness and Soft Computing)*, Springer, Heidelberg, 2005.

[5] L. Bull, E. B. Mansilla and J. H. Holmes, *Learning Classifier Systems in Data Mining (Studies in Computational Intelligence)*, Springer, Heidelberg, 2008.

[6] A. Hamzeh and A. Rahmani, Approximation the environmental reinforcement signal with non-linear polynomials using learning classifier systems, *International Journal of Innovative Computing, Information and Control*, vol.4, no.7, pp.1797-1809, 2008.

[7] S. Schulenburg and P. Ross, An adaptive agent based economic model, learning classifier systems, *Foundations to Applications*, pp.263-282, 2000.

[8] Z. Pawlak, Rough sets, *International Journal of Computer and Information Sciences*, vol.11, pp.341-356, 1982.

[9] W. Marek and Z. Pawlak, Rough sets and information systems, *Fundamenta Informaticae*, vol.17, pp.105-115, 1984.

[10] L. Polkowski, *Rough Sets: Mathematical Foundations (Advances in Soft Computing)*, Physica-Verlag, Berlin, Germany, 2002.

[11] T. H. Nguyen, S. Foitong and Q. Pinngern, Rough set and XCS in classification problems, *International Conference on Computer and Communication Engineering*, pp.806-811, 2008.

[12] M. V. Butz, M. Pelikan, X. Llor and D. E. Goldberg, Automated global structure extraction for effective local building block processing in XCS, *Evolutionary Computation*, vol.14, no.3, pp.345-380, 2006.

[13] S. W. Wilson, Get real! XCS with continuous-valued inputs, *Foundations to Applications, LNCS*, vol.1813, pp.209-222, 2000.

[14] S. W. Wilson, Mining oblique data with XCS, *The 3rd IWLCS, LNCS*, Paris, France, vol.1996, pp.158-176, 2001.

[15] E. Bernadeo-Mansilla, X. Llor and J. M. Garrell, XCS and GALE: A comparative study of two learning classifier systems with six other learning algorithms on classification tasks, *The 4th International Workshop on Learning Classifier Systems*, pp.337-341, 2001.

[16] J. Bacardit and M. V. Butz, Data mining in learning classifier systems: Comparing XCS with GAssist, *Learning Classifier Systems: International Workshops, LNCS*, vol.4399, pp.282-290, 2007.

[17] H. H. Dam, K. Shafi and H. A. Abbass, Can evolutionary computation handle large datasets? A study into network intrusion detection, in *Australian Conference on Artificial Intelligence*, S. Zhang and R. Jarvis (eds.), 2005.

[18] H. Brighton and C. Mellish, Identifying competence-critical instances for instance-based learners, in *Instance Selection and Construction for Data Mining*, H. Liu and H. Motoda (eds.) Norwell, MA, Kluwer, 2001.

[19] D. Kibbler and D. W. Aha, Learning representative exemplars of concepts: An initial case of study, *Proc. of the 4th Int. Workshop Machine Learning*, pp.24-30, 1987.

[20] H. Liu and H. Motoda, On issues of instance selection, *Data Mining Knowledge Discovery*, vol.6, no.2, pp.115-130, 2002.

[21] S. Tan, X. Cheng and H. Xu, An efficient global optimization approach for rough set based dimensionality reduction, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.725-736, 2007.

[22] H. Liu and H. Motoda, Data reduction via instance selection, in *Instance Selection and Construction for Data Mining*, H. Liu and H. Motoda (eds.), Boston, Kluwer Academic Publishers, 2001.

[23] J. R. Cano, F. Herrera and M. Lozano, On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining, *Applied Soft Computing*, vol.6, no.3, pp.323-332, 2006.

[24] Y. Caballero, S. Joseph, Y. Lezcano, R. Bello, M. M. Garcia and Y. Pizano, Using rough sets to edit training set in k-NN method, *Proc. of the 5th International Conference on Intelligent Systems Design and Applications*, Washington, DC, pp.456-463, 2005.

[25] H. Liu, S. Xiong and Q. Chen, Training fuzzy support vector machines by using boundary of rough set, *Proc. of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, Shanghai, China, pp.883-886. 2009.

[26] Y. Matsumoto and J. Watada, Knowledge acquisition from time series data through rough sets analysis, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4885-4898, 2009.

[27] S. W. Han and J. Y. Kim, A new decision tree algorithm based on rough set theory, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2749-2758, 2008.

[28] W. Kasemsiri and Y. Shi, Thai characters recognition based on tolerant rough sets with fuzzy C-mean, *ICIC Express Letters*, vol.3, no.4(B), pp.1399-1404, 2009.

[29] R. Yan, J. Zheng and J. Liu, Rough set over dual-universes and its applications in expert systems, *ICIC Express Letters*, vol.4, no.3(A), pp.833-838, 2010.

[30] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., 1989.

[31] X. Llor and J. M. Guiu, Inducing partially-defined instances with evolutionary algorithms, *Proc. of the 18th International Conference on Machine Learning*, pp.337-344, 2001.

[32] C. Blake, E. Keogh and C. Merz, *UCI Repository of Machine Learning Databases*, www.ics.uci.edu/mlearn/MLRepository.html, 1998.

[33] M. V. Butz, *Rule-Based Evolutionary Online Learning Systems: Learning Bounds, Classification, and Prediction*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 2004.

[34] S. W. Wilson, Generalization in the XCS classifier system, *Proc. of the 3rd Annual Conference*, pp.665-674, 1998.