

A new classification method based on rough sets theory

Rasim Cekik¹ · Sedat Telceken¹

Published online: 19 December 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Discovering the common attributes of an object is an important problem in classification. The rough sets theory (RST) successfully reveals the relationship between an object, its attributes and classes and helps bring a solution to the classification problem. In this study, a new classification method has been developed that uses RST and a similarity-based method to create the weight matrix scoring system. The proposed method is named feature weighted rough set classification (FWRSC) and is compared with the classification methods in WEKA for five different datasets. The experimental results show that FWRSC gives higher performance than most of the methods in WEKA. Additionally, FWRSC produces the highest performance in terms of accuracy with an overall average of 67.47% for five different datasets.

Keywords Rough sets theory (RST) · Data mining · Classification

1 Introduction

The rapid development of computer technology has given rise to an exponential growth of data. According to a study presented in (Witten and Frank 2005), the amount of data doubles every 20 months in the world. It is obvious that this growing big data would be redundant with

many repetitions, incompleteness, inconsistencies, vagueness, uncertainties and therefore is difficult to interpret and analyze. This low-quality nature of data creates big problems for such areas as classification, clustering, feature selection, feature extraction, knowledge discovery, image processing, prediction and text mining. Among these areas, data mining is the most affected problem from missing and hidden data, and therefore, the focus of this paper is to decrease classification problems for missing and hidden data.

This study focuses on classification and proposes a new classification method. The simplest definition of classification is the process of finding the class that an object belongs to. Numbers and labels of classes are clear in the classification process. Moreover, the fact that each object has features and these features are related to the classes is important for classification. Nevertheless, there are some problems that make correct classification a difficult job. For example, classifying within a dataset that is inconsistent, incomplete and contains redundant data is difficult. To solve these problems in data mining, preprocessing procedures such as data cleaning, relevance analysis, data transformation, and reduction are applied. Recently, however, rough sets theory (RST) has been widely and effectively used for this purpose in the literature. Lately, some researchers have also proposed classification methods for inconsistent and incomplete data. For example, Khoo et al. (1999) proposed a novel approach for the classification and rule induction of incomplete information systems. Xiang-Wei and Yian-Fang (2012) proposed a novel effective preprocessing algorithm using RST. Chakhar and Saad (2012) (<https://archive.ics.uci.edu/ml/datasets.html>) proposed a methodology to support groups in multi-criteria classification problems.

RST makes a dataset more consistent by working on the incomplete and inconsistent data. Additionally, RST can find which features of an object are more effective than the others

Communicated by V. Loia.

✉ Sedat Telceken
stelceken@gmail.com

Rasim Cekik
rasimcekik@anadolu.edu.tr

¹ Department of Computer Engineering, Anadolu University, 26555 Eskisehir, Turkey

in the classification of the object. The experimental results as well as related works in the literature show that more effective classification can be achieved by using these aspects of RST.

2 Rough sets theory

Rough sets theory (RST) (Pawlak 1982, 1991, 1998; Pawlak and Skowron 2007) was proposed by Pawlak in 1982, and it is a mathematical method used for generating rules and evaluation for expert systems such as pattern recognition, machine learning and knowledge discovery. RST has recently been attracting people's attention due to its ability to obtain useful information from missing and inconsistent data, which has opened up very wide application areas (Kryszkiewicz 1998; Dubois and Prade 1990; Pawlak and Sowinski 1994; Jensen and Shen 2004; Duntsch and Gediga 1998; Pawlak 2002; Swiniarski and Skowron 2003; Hu and Cercone 1995; Slowinski and Zopounidis 1995; Pawlak 1984), and to resolve problems related to vague, uncertain and incomplete information. RST also has wide usage for feature extraction and reduction in data mining.

Constructing a wide and comprehensive Information system (IS) related with the problem is the first step of RST (Cekik and Telceken 2014).

$$IS = (U, A, D) \quad (1)$$

U is a non-empty finite set of objects ($U = x_1, x_2, \dots, x_m$), A is non-empty finite set of features ($A = a_1, a_2, \dots, a_m$), D is non-empty finite set of classes decisions ($D = d_1, d_2, \dots, d_k$). $\forall a \in A$ defines an information function $f_a : U \rightarrow V_a$, where V_a is the set of values of a , called the domain of feature a .

$$Z_{i,j} = \{\forall a \in A \mid a(x_i) \neq a(x_j)\} \quad i, j = 1, 2, \dots, n \quad (2)$$

Core and reducts of features are computed by using a *Discernibility matrix* of size $n \times n$, where n denotes the number of objects. $Z_{i,j}$ is defined as the set of all features which discern objects x_i and x_j . A *Discernibility function* is a Boolean function of m Boolean variables $a_1^*, a_2^*, \dots, a_m^*$ which correspond to the features a_1, a_2, \dots, a_m defined as below,

$$f_A(a_1^*, a_2^*, \dots, a_m^*) = \bigwedge \{\bigvee Z_{i,j}^* \mid 1 \leq j \leq i \leq n, Z_{i,j} \neq \emptyset\} \quad (3)$$

where

$$Z_{i,j}^* = \{a^* \mid a \in Z_{i,j}\}.$$

The set of all prime implicates of f_A determines the set of all reducts of A (Komorowski 1999). In this study, core and

reduced attributes for every object have been obtained using the discernibility matrix and functions in RST. The goal is to demonstrate different characteristics that determine the class of an object depending on the attributes of the object in different classes. For example, if the first and the third attributes are obtained as discernibility attributes for an object, these attributes are derived to be the discernibility of this object from the other objects. Let d_k denote a class and,

$$i, j = 1, 2, \dots, n \\ Z_{i,j} = \{\forall a \in A \mid a(x_i) \neq a(x_j) \text{ and } d(x_i) \neq d(x_j)\} \quad (4)$$

Another concept in RST is upper/lower approximation sets. Let $R \subseteq A$ and $X \subseteq U$. Set X can be approximated by using the features contained in R by defining the lower and upper approximations of X , denoted $\underline{R}(X)$ and $\overline{R}(X)$, respectively, where $\underline{R}(X) = \{x \in U \mid [x]_R \subseteq X\}$ and $\overline{R}(X) = \{x \in U \mid [x]_R \cap X \neq \emptyset\}$. The objects in $\underline{R}(X)$ can be classified as members of X on the basis of the knowledge in R . On the other hand, the objects in $\overline{R}(X)$ can only be classified as possible members of X . The boundary set of X is defined as $BN_R(X) = \overline{R}(X) - \underline{R}(X)$ and thus consists of those objects that cannot be classified exactly into X on the basis of the knowledge in R . Four basic classes of rough sets can be defined as follows:

- X is *roughly R-definable*, $\Leftrightarrow \underline{R}(X) \neq \emptyset$ and $\overline{R}(X) \neq U$.
- X is *internally R-undefinable*, $\Leftrightarrow \underline{R}(X) = \emptyset$ and $\overline{R}(X) \neq U$.
- X is *externally R-undefinable*, $\Leftrightarrow \underline{R}(X) \neq \emptyset$ and $\overline{R}(X) = U$.
- X is *totally R-undefinable*, $\Leftrightarrow \underline{R}(X) = \emptyset$ and $\overline{R}(X) = U$.

The last concept we talk about RST is the positive, negative and boundary regions of features. Let P and Q be sets of attributes including equivalence relations over U . Then the positive, negative and boundary regions are defined as $POS_P(Q) = U_{X \in U/Q} \underline{P}X$, $NEG_P(Q) = U - U_{X \in U/Q} \overline{P}X$ and $BND_P(Q) = U_{X \in U/Q} \overline{P}X - U_{X \in U/Q} \underline{P}X$, respectively.

3 A new rough sets classification method

Data mining discipline has been formed to analyze the big data and extract meaningful information to cope with the problem of data explosion. Data mining uses statistics, machine learning and artificial intelligence and has been the focus of attention in the industry, economy and academy with great interest from the business community. Data mining methods that contain numerous algorithms are used in

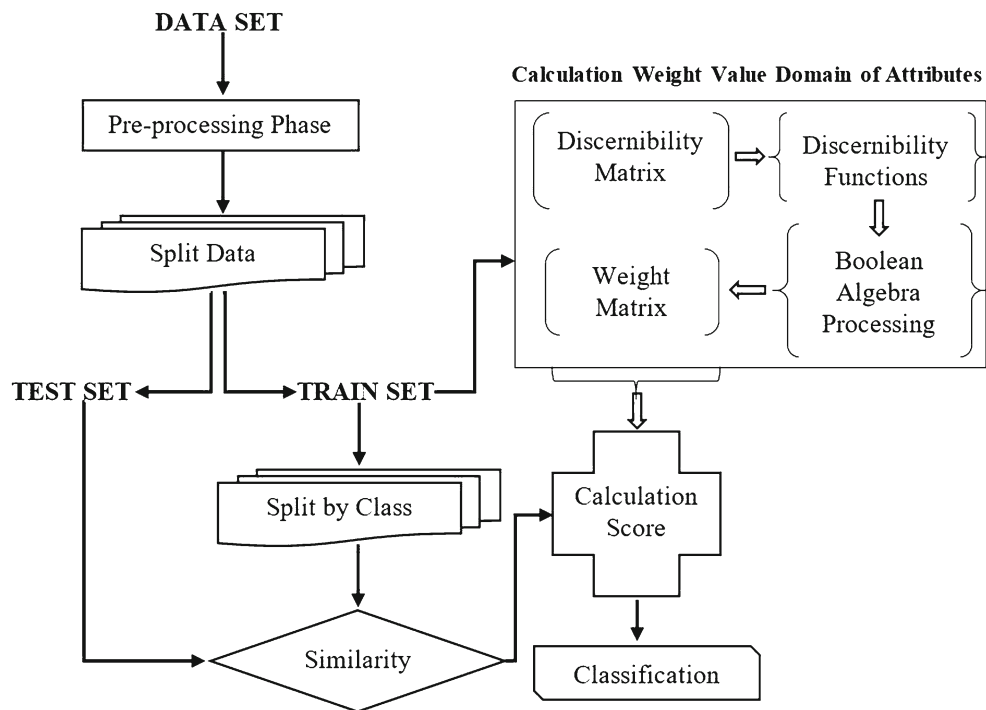


Fig. 1 FWRSC classification model

health industry, basic sciences, banking, finance and market research. Methods used in data mining can be classified as unsupervised clustering, supervised learning and market basket analysis in the most general sense. Supervised learning can be classified as classification, estimation and prediction in itself.

Classification is the process of putting an object into a pre-defined class using its features. The classification technique is a systematic approach, which builds classification models from input datasets. Each technique relies on a learning algorithm, which tries to construct a relation between features of the input data and the class labels (Han et al. 2011; Tan et al. 2006; Cios et al. 2012).

The classification is accomplished by computing relations between features and classes according to common and distinct features among the classes. Classification becomes easier whenever this relation is simple and the number of common features is small. Unfortunately, the reverse condition is applicable for most of the applications in real life. A feature can be common for more than one class in nature, which would make classification hard and complex. Additionally, some attributes or attribute values can be redundant and unnecessary for the classification of an object. In practice, it means that the object can be classified despite the fact that the values of some attributes may be missing, since the remaining attributes or attribute values are sufficient for classification or to make a decision.

In this study, discernible features were determined for each class. Hence, the relationship between a class and a feature value is simplified avoiding the problem of hidden classification. Consequently, the performance of the classification process is increased. In order to handle the problems explained above, a new rough-set-based classification method, named feature weighted rough set classification (FWRSC), is proposed in this paper. Steps of FWRSC are shown in Fig. 1. The preprocessing phase shown in the figure is used to fill missing values in the dataset using discretization, interpolation, etc. This step was not used in this paper since there are no missing values in the used datasets. In the second step of the processing pipeline in Fig. 1, i.e., the split data phase, the dataset is first divided into two parts as training and testing. The training set is then divided into several classes based on the class label. In this step, FWRSC processes the objects in the training dataset to determine the best attributes that discriminate the objects in their classes. To achieve this, FWRSC creates what is called the *weight matrix scoring system*. The rows of this matrix are the objects, and the columns are the attributes. Each cell of the matrix signifies the weight of the attribute in determining the particular object's class label. The higher the weight of an attribute, the higher its contribution toward determining the class of that object. Similarly, the attributes with smaller values have smaller effect on determining the class label of the object. The details of how the weight matrix scoring system is cal-

culated are given in Sect. 3.1. After the feature vectors are determined using the training set, they are used to classify the objects in the test set. The classification of a test object works as follows: Each attribute value of the test object is compared with the attributes of a training object. If the attribute values are the same, a score of 1 is assigned to that attribute, otherwise 0 is assigned. The score values of all attributes are then added up to compute the cumulative score for the object. The test object is then assigned to the same class as the training object for which this score is maximized. The details of this procedure is discussed in Sect. 3.2.

3.1 Weight matrix calculation

The discernibility relations are closely related to indiscernibility and belong to the most important relations considered in RST. The ability to discern between perceived objects is important for constructing many structures such as reducts, classification, decision rule or algorithm (Pawlak and Skowron 2007). Particularly, efficiency of this relation affects the whole classifier performance in data mining. It reveals core and reduced attributes, and the important attributes for an object is clearly made visible. Based on this information, a scoring system for each attribute of an object is made applicable. The scoring system process works similar to term frequency (TF) (Manning et al. 2008) in the text document classification problem. Here the discernibility function can be thought as the text document and the reducts which are combined with the logical OR operator can be thought as the sentences in the text document. Then, computation of the weight of a feature is similar to the computation of the term frequencies. The only difference is that the number of the OR operators plus is used instead of the number of terms. For example, in an information system with five attributes, if the core attributes of an object are the first and the third attributes, then the effect of these two attributes must be improved. In this way, the effect of core attributes that better define the class of an object must be increased while reducing the influence of non-core attributes. Thus, a more accurate classification process can be carried out.

In this study, a matrix for scoring the system attributes, called the weight matrix ($W_{i,j}$) is created. f_i is discernibility function to i th object, and $\text{Count}(f_i, a_j)$ is the frequency of the attribute a_j in f_i . \bigvee denotes the Boolean OR operator in f_i , then $\text{Count}(f_i, \bigvee)$ is the frequency of the \bigvee which is the Boolean OR operator in f_i . It is formalized:

$$N = \text{Count} \left(f_i, \bigvee \right) + 1 \tag{5}$$

$$d_{i,j} = \frac{\text{Count}(f_i, a_j)}{N} \tag{6}$$

Weight matrix ($W_{i,j}$):

$$\begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ d_{m,1} & d_{m,2} & \dots & d_{m,n} \end{pmatrix}$$

Let $IS = (U, A)$ be an information system that has 10 objects and four attributes. The discernibility function for the fifth object:

$$f_5 = a_1 \wedge a_2 \bigvee a_2 \wedge a_3 \wedge a_4$$

The weight of each attribute for 5th object:

$$\begin{aligned} \text{Count}(f_5, a_1) &= 1, & \text{Count}(f_5, a_2) &= 2, \\ \text{Count}(f_5, a_3) &= 1, & \text{Count}(f_5, a_4) &= 1, \\ N &= \text{Count} \left(f_i, \bigvee \right) + 1 = 2 \end{aligned}$$

According to this;

$$\begin{aligned} d_{5,1} &= \frac{\text{Count}(f_5, a_1)}{N} = \frac{1}{2} = 0.5, \\ d_{5,2} &= \frac{\text{Count}(f_5, a_2)}{N} = \frac{2}{2} = 1.0, \\ d_{5,3} &= \frac{\text{Count}(f_5, a_3)}{N} = \frac{1}{2} = 0.5, \\ d_{5,4} &= \frac{\text{Count}(f_5, a_4)}{N} = \frac{1}{2} = 0.5, \end{aligned}$$

The weight of each attribute for the fifth object:

$$W_{5,j} = [0.5 \quad 1 \quad 0.5 \quad 0.5]$$

Here, a_1 attribute presents only in $a_1 \wedge a_2$, and hence, $\text{Count}(f_5, a_1)$ is equal to 1. a_2 attribute presents both $a_1 \wedge a_2$ and $a_2 \wedge a_3 \wedge a_4$; hence, $\text{Count}(f_5, a_2)$ is equal to 2. Vector $W_{5,j}$ shows the weight of each attribute for the fifth object. As seen from the example, attribute a_1 is more important than the other attributes for the fifth object.

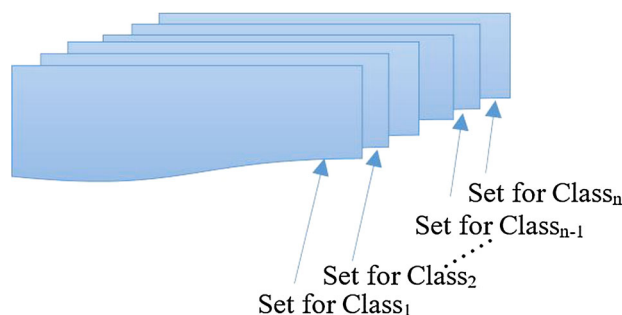


Fig. 2 Sets for classes

3.2 Determining the class of an object

The first step in the classification process is to split the training data into a set of ‘n’ classes. Let C_t denote the class of the t th object, and D define the set of classes (Fig. 2).

Formal definitions of each class are as follows:

$$\begin{aligned}
 r &= \{d_i, d_j \in D \mid \forall d_i \neq \forall d_j, 0 < i \leq j \leq S(D)\} \\
 t &= 1, 2, \dots, S(r) \\
 C_t &= \{x_k \in U; d_i, d_j \in D \mid d_i = d_j, x_k\} \tag{7}
 \end{aligned}$$

Here, X is a finite set, and function $S(X)$ is the number of elements in X . Similarly, $S(D)$ denotes the total number of objects in IS, and r denotes the number of classes in IS.

Let $IS = (U, A, D)$ be an information system, and let $\beta, \gamma \subset IS$ be the test and the training sets, respectively. Then the class score for a test object for class ‘t’ is calculated as follows:

$$\begin{aligned}
 R^* &= S(U) \quad \text{and} \quad r^* = S(A) - 1 \\
 \text{Class}_{\text{score}}^t &= \frac{\sum_{\gamma_i \in C_t}^{R^*} \sum_{\beta(a_j)=\gamma(a_j)}^{r^*} (1 + W_{i,j})}{S(C_t)} + \frac{S(C_t)}{S(U)} \tag{8}
 \end{aligned}$$

where $S(C_t)$ is number of objects (or elements) of t th class set and $S(U)$ is the total number of objects in IS, and the ratio of $S(C_t)$ to $S(U)$ is the class rate.

As seen from Eq. 8, the object to be classified in the test dataset is compared with all objects in the training dataset and a score is computed for each class. The computation of this score proceeds as follows: When the value of an attribute is the same for the object in question and for the object in the training set, its score value for similarity is 1, and this score is added to the weight of that attribute in $W_{i,j}$. The reason for adding the feature weight to the similarity score is to prevent hidden classification and thus enable more sensitive classification by increasing the accuracy. Furthermore, it exposes the important and unimportant attributes that determine the class of an object. The total weight thus computed becomes the feature weight for the object for the particular class. Finally, the class rate is added to the score obtained thus far to come up with the overall score for the class.

To summarize, the computation of the class score for a test object consists of three parts: the similarity value, the feature value and the class rate. The similarity value incorporates how similar an object in test dataset is to each object in the training set, but it does not prevent hidden classification and does not expose hidden pattern in data. At this point, the feature value comes into play and assigns a weight to each attribute. By giving more weight to the important attributes, the hidden classification and pattern problems are avoided. Finally, the class rate ensures internal balance.

Having computed the class scores, the class of an object in the test dataset is determined by:

$$\text{Calculated class} = \max (\text{Class}_{\text{score}}^t) \tag{9}$$

Clearly, an object in the test dataset is included in the class having the highest score value. If two or more classes have the same score value, then one of the classes is randomly chosen for the object.

The algorithm for the proposed FWRSC method is given below:

Algorithm 1 Algorithm for FWRSC

```

Require: //input: training set (a decision table T = (U, A, D))
1: for  $i = 1$  to  $ClassNumber$  do
2:   for  $i = 1$  to  $size(T(:, 1))$  do
3:     if  $Classes(j) == ClassVector$  then
4:        $SeparateClassIndex()$ ;
5:     end if
6:   end for
7: end for
8:  $CalculateDiscernibilityFunctions()$ 
9: for  $i = 1$  to  $length(TestData)$  do
10:  for  $i = 1$  to  $ClassNumber$  do
11:     $WhichClass = SeparateClassIndex(:, j)$ ;
12:     $CalculateWeightMatrix()$ 
13:     $TestVector = TestData(i, :)$ ;
14:     $CalculateClass()$ ;
15:  end for
16: end for
    
```

3.3 Experimental results

In this section, we quantitatively evaluate the performance of the proposed classification method. Data sets used in experiments have been taken from the website of UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets.html>). The classification performance of FWRSC is compared with as many of the classification methods as possible in a program called WEKA (Garner 1995; Markov and Russell 2006; Bouckaert 2013) on five different dataset. The selected datasets have been randomly chosen so as to cover a wide variety of application areas. For example, if a dataset is from the health industry, other datasets belong to areas outside of the health industry. Table 1 gives the details of the selected datasets.

There are several criteria for the evaluation of a classification method. In this paper, accuracy, F -score (F -measure) and mean absolute error (MAE) criteria have been used. The more common definition is that accuracy is a level of measurement with no inherent limitation (i.e., free of systematic error, another form of observational error). In statistical analysis of binary classification, the F -score (also $F1$ score or F -measure) is a measure of a test’s accuracy. It considers

Table 1 Datasets

Datasets	Number attributes	Number classes	Total instances
Ballon	4 + Class = 5	2	56
Balance scale	4 + Class = 5	3	625
Haberman's survival	3 + Class = 4	2	306
Abalone	8 + Class = 9	29	4177
Tic-Tac-Toe	9 + Class = 10	2	958

Table 2 Accuracy results for different classification methods

Classification	Classification methods	Balloon (%)	Balance scale (%)	Haberman survival (%)	Abalone (%)	Tic-Tac-Toe (%)
Bayes	BayesNET	40.00	65.71	66.27	53.36	25.86
	NaiveBayes	70.00	87.61	72.29	53.36	16.20
	NaiveBayesMultinomial	45.00	89.52	68.67	43.01	49.22
	NaiveBayesSimple	60.00	88.09	71.08	53.22	16.20
	ComplementNaiveBayes	65.00	90.95	71.08	50.53	14.33
Functions	Logistic	70.00	89.05	69.88	58.46	07.16
	MultilayerPerceptron (MLP)	70.00	93.33	71.08	58.40	30.53
	SMO	70.00	89.52	66.27	56.38	13.39
Lazy	IB1	60.00	81.42	66.27	49.19	50.15
	IBK	60.00	80.47	65.06	49.19	47.97
	KStar	70.00	82.38	66.27	55.24	37.38
	LWL	60.00	68.57	66.27	55.91	51.71
Meta	AdaBoostM1	70.00	61.90	66.27	55.91	19.93
	Bagging	60.00	80.47	67.45	57.86	23.68
	Decorate	60.00	73.81	63.85	55.37	23.68
	LogitBoost	70.00	82.86	65.06	56.85	15.89
Misc	HyperPipes	40.00	33.33	66.27	36.29	62.30
	VFI	70.00	51.90	59.04	33.40	17.44
Rules	JRip	60.00	76.66	69.88	55.51	26.48
	OneR	80.00	58.09	68.88	50.73	63.55
	PART	60.00	78.57	69.88	53.02	21.18
	Ridor	60.00	81.42	66.26	55.10	42.36
Tree	J48	60.00	69.04	65.06	52.88	23.36
	SimpleCart	60.00	66.66	68.67	57.39	26.16
	NBTree	80.00	71.42	66.27	53.36	42.05
FWRSC	FWRSC	90.00	65.24	66.27	54.91	62.30
	Average (%)	63.84	75.30	67.29	52.49	31.94

both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F -score can be interpreted as a weighted average of the precision and recall, where an F -score reaches its best value at 1 and worst at 0. In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to

the eventual outcomes, and it is one of a number of ways of comparing forecasts with their eventual outcomes.

The success of the classification methods is illustrated in Tables 2, 3 and 4 for accuracy, MAE and F -score, respectively. The best results for each dataset are made darker in all tables. The performance value for the proposed FWRSC method is given at the last row of each table. Let's analyze the accuracy results in Table 2. While FWRSC has the highest accuracy for the Balloons dataset, its performance on other

Table 3 Classification methods having an average accuracy percentage above 60%

Datasets	MLP	IB1	IBK	KStar	LWL	OneR	Ridor	NBTree	FWRSC
Balloon	70.00	60.00	60.00	70.00	60.00	80.00	60.00	80.00	90.00
Balance scale	93.33	81.42	80.47	82.38	68.57	58.09	81.42	71.42	65.24
Haberman survival	71.08	66.27	65.06	66.27	66.27	68.88	66.26	66.27	66.27
Abalone	58.40	49.19	49.19	55.24	55.91	50.73	55.10	53.36	54.91
Tic-Tac-Toe	30.53	50.15	47.97	37.38	51.71	63.55	42.36	42.05	62.30
Average	64.67	61.41	60.54	62.25	60.49	64.25	61.03	62.62	67.74

Table 4 MAE for different classification methods

Classification	Classification methods	Balloon	Balance scale	Haberman survival	Abalone	Tic-Tac-Toe
Bayes	BayesNET	0.5054	0.2716	0.3797	0.3141	0.6618
	NaiveBayes	0.3146	0.1866	0.3480	0.3109	0.7185
	NaiveBayesMultinomial	0.4905	0.2649	0.2990	0.4273	0.5393
	NaiveBayesSimple	0.3042	0.1818	0.3490	0.3114	0.7381
	ComplementNaiveBayes	0.3500	0.0603	0.2892	0.3297	0.8567
Functions	Logistic	0.2787	0.0816	0.3727	0.3463	0.8401
	MultilayerPerceptron	0.2810	0.0719	0.3316	0.3316	0.6974
	SMO	0.3000	0.2614	0.3373	0.3435	0.8660
Lazy	IB1	0.4000	0.1238	0.3373	0.3387	0.4984
	IBK	0.2722	0.1314	0.3406	0.3388	0.5811
	KStar	0.2962	0.1865	0.3567	0.3388	0.5976
	LWL	0.3992	0.2940	0.3701	0.3721	0.5574
Meta	AdaBoostM1	0.2692	0.3488	0.3693	0.4370	0.7270
	Bagging	0.3492	0.2082	0.3693	0.3355	0.7399
	Decorate	0.3885	0.2778	0.3887	0.3493	0.7468
	LogitBoost	0.2823	0.1572	0.3782	0.3493	0.7604
Misc	HyperPipes	0.5000	0.4348	0.5000	0.4388	0.5059
	VFI	0.4748	0.4326	0.4998	0.4335	0.5171
Rules	JRip	0.4167	0.1974	0.3802	0.3519	0.7295
	OneR	0.2000	0.2794	0.3133	0.3284	0.3645
	PART	0.3792	0.1504	0.3525	0.3464	0.7829
	Ridor	0.4000	0.1238	0.3373	0.3189	0.5763
	Tree	J48	0.3792	0.2512	0.3629	0.3375
FWRSC	SimpleCart	0.3792	0.2571	0.3547	0.3402	0.7343
	NBTree	0.3717	0.2433	0.3804	0.3141	0.5905
	FWRSC	0.4000	0.5381	0.2289	0.3010	0.4587
	Average	0.3608	0.2314	0.3587	0.3494	0.6595

datasets is close to the best classification method. For example, on the Tic-Tac-Toe dataset, the best accuracy results are obtained with the Rules-OneR classification method for an accuracy value of 63.55%. On the same dataset, FWRSC produces the second highest classification score of 62.30%. Also observe that the performance of FWRSC on other datasets is also close to the best classification method. To better summarize the accuracy results listed in Table 2, a compact

representation of only the best classification methods with an average accuracy result greater than 60% is listed in Table 6. It can clearly be observed from Table 3 that FWRSC produces the best average classification accuracy of 67.74% over all datasets, while giving the best performance for the Balloon dataset.

Analyzing the MAE and *F*-score results given in Tables 4 and 5, respectively, we see that FWRSC gives the best MAE

Table 5 *F*-measure results for different classification methods

Classification	Classification methods	Balloon	Balance scale	Haberman survival	Abalone	Tic-Tac-Toe
Bayes	BayesNET	0.229	0.631	0.528	0.498	0.268
	NaiveBayes	0.691	0.843	0.650	0.498	0.175
	NaiveBayesMultinomial	0.771	0.865	0.644	0.343	0.411
	NaiveBayesSimple	0.567	0.848	0.641	0.498	0.175
	ComplementNaiveBayes	0.631	0.876	0.679	0.411	0.143
Functions	Logistic	0.691	0.858	0.605	0.578	0.069
	MultilayerPerceptron	0.691	0.931	0.641	0.539	0.315
	SMO	0.691	0.865	0.528	0.471	0.144
Lazy	IB1	0.567	0.807	0.646	0.498	0.469
	IBK	0.567	0.804	0.624	0.498	0.482
	KStar	0.691	0.805	0.566	0.551	0.380
	LWL	0.567	0.661	0.566	0.465	0.425
Meta	AdaBoostM1	0.691	0.603	0.528	0.465	0.211
	Bagging	0.567	0.781	0.574	0.581	0.247
	Decorate	0.567	0.708	0.535	0.553	0.248
	LogitBoost	0.691	0.806	0.542	0.569	0.170
Misc	HyperPipes	0.229	0.345	0.528	0.198	0.478
	VFI	0.691	0.565	0.601	0.205	0.112
Rules	JRip	0.567	0.784	0.653	0.546	0.205
	OneR	0.800	0.568	0.635	0.511	0.438
	PART	0.567	0.773	0.620	0.514	0.216
	Ridor	0.567	0.787	0.566	0.548	0.428
Tree	J48	0.567	0.663	0.542	0.524	0.243
	SimpleCart	0.567	0.645	0.624	0.576	0.273
	NBTree	0.800	0.680	0.528	0.498	0.427
FWRSC	FWRSC	0.899	0.331	0.435	0.338	0.500
	Average	0.620	0.724	0.587	0.479	0.294

performance on Haberman's survival and Abalone datasets and the best *F*-score performance for Balloon and Tic-Tac-Toe datasets.

We also can say that the performance of FWRSC on all date sets for MAE and *F*-score remains just below the overall average of all classifiers with a very small difference. However, FWRSC is one of classifiers that show the best performance after OneR classifier for MAE criteria on Tic-Tac-Teo dataset in Table 4. This shows that FWRSC performs a classification process with low error rate (See Table 4). Additionally, although OneR classifiers indicate the best performance on Tic-Tac-Teo dataset in Table 2, FWRSC classifiers indicate the best performance on this dataset shown in Table 5. This means that FWRSC produces more a balanced classification process than OneR and other classifiers that in tables. All and all, it can be concluded that FWRSC finds itself to be among the best classification methods found in the literature.

4 Conclusions and future work

In this article, a new classification method named feature weighted rough set classification (FWRSC) was proposed to decrease classification problems on account for hidden in data. The most important advantages of FWRSC are reduced complexity, handling inconsistency in the dataset and taking care of hidden classification. RST can find core and reduct attributes. Core attributes that influence the class of an object more than the others, and the attributes that have little impact on the classification performance are determined. Therefore, the ill impacts of classification problems are minimized, and more efficient classification is made possible. FWRSC is compared with the classification methods in WEKA for five different datasets. The experimental results show that FWRSC gives higher performance than most of the methods in WEKA for many of the datasets. Furthermore, FWRSC gives the best average accuracy performance with an over-

all average score of 67.47% for five different datasets. We believe that FWRSC will find its rightful place among the best classification methods found in the literature. Our future goal is to improve the performance of FWRSC by incorporating some hybrid RST techniques.

Acknowledgements This work has been partially supported by Anadolu University Scientific Research Project Commission under the Grant Number 1402F047.

Compliance with ethical standards

Conflict of interest None.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Bouckaert RR et al (2013) WEKA manual for version 3-7-8
- Cekik R, Telceken S (2014) Classification of ECG signals using rough sets theory, Anadolu University. *J Sci Technol A Appl Sci Eng* 15(2):125–135
- Chakhar S, Saad I (2012) Dominance-based rough set approach for groups in multicriteria classification problems. *Decis Support Syst* 54(1):372–380
- Cios KJ, Pedrycz W, Swiniarski RW (2012) Data mining methods for knowledge discovery, vol 458. Springer Science and Business Media, Berlin
- Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. *Int J Gen Syst* 17(2–3):191–209
- Duntsch T, Gediga G (1998) Uncertainty measures of rough set prediction. *Artif Intell* 106(1):109–137
- Garner SR (1995) Weka: the waikato environment for knowledge analysis. In: Proceedings of the New Zealand computer science research students conference, pp 57–64
- Han J, Kamber M, Pei J (2011) Data mining: concepts and techniques: concepts and techniques. Elsevier, Amsterdam
- Hu X, Cercone N (1995) Learning in relational databases: a rough set approach. *Comput Intell* 11(2):323–338
- Jensen R, Shen Q (2004) Fuzzy-rough attribute reduction with application to web categorization. *Fuzzy Sets Syst* 141(3):469–485
- Khoo LP, Tor SB, Zhai LY (1999) A rough-set-based approach for classification and rule induction. *Int J Adv Manuf Technol* 15(6):438–444
- Komorowski J, Pawlak Z, Polkowski L, Skowron A (1999) Rough sets: a tutorial. In: Pal SK, Skowron A (eds) *Rough-fuzzy hybridization: a new trend in decision-making*. Springer, Singapore, pp 1–98
- Kryszkiewicz M (1998) Rough set approach to incomplete information systems. *Inf Sci* 112(1):39–49
- Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval, vol 1. Cambridge University Press, Cambridge
- Markov Z, Russell I (2006) An introduction to the WEKA data mining system. *ACM SIGCSE Bull* 38(3):367–368
- Pawlak Z (1982) Rough sets. *Int J Inf Comput Sci* 11(5):341–356
- Pawlak Z (1984) Rough classification. *Int J Man Mach Stud* 20(5):469–483
- Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Series D: system theory, knowledge engineering and problem solving, vol 9. Kluwer, Dordrecht
- Pawlak Z (1998) Rough set theory and its applications to data analysis. *Cybern Syst* 29(7):661–688
- Pawlak Z (2002) Rough sets and intelligent data analysis. *Inf Sci* 147(1):1–12
- Pawlak Z, Skowron A (2007) Rudiments of rough sets. *Inf Sci* 177(1):3–27
- Pawlak Z, Sowiński R (1994) Rough set approach to multi-attribute decision analysis. *Eur J Oper Res* 72(3):443–459
- Slowinski R, Zopounidis C (1995) Application of the rough set approach to evaluation of bankruptcy risk. *Intell Syst Account Finance Manag* 4(1):27–41
- Swiniarski RW, Skowron A (2003) Rough set methods in feature selection and recognition. *Pattern Recognit Lett* 24(6):833–849
- Tan PN, Steinbach M, Kumar V (2006) Classification: basic concepts, decision trees, and model evaluation. *Introduct Data Min* 1:145–205
- Witten IH, Frank E (2005) Data mining, practical machine learning tools and techniques. Morgan Kaufmann, Los Altos
- Xiang-Wei L, Yian-Fang Q (2012) A data preprocessing algorithm for classification model based on Rough sets. *Phys Proc* 25:2025–2029