Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2020) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

24rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Combining Rough Set-based Relevance and Redundancy for the Ranking and Selection of Nominal Features

Wojciech Froelich [a,*], Petr Hajek[b]

[a] *Institute of Computer Science, University of Silesia, ul. Bedzinska 39, Sosnowiec, Poland*

[b] *Institute of System Engineering and Informatics, Faculty of Economics and Administration, University of Pardubice, Studentska 84, 532 10 Pardubice, Czech Republic*

Abstract

In this paper, we propose a new method for features ranking and selection. Our approach is based on ranking nominal features in terms of their relevance to the assigned class and mutual redundancy with the other features. To calculate the relevance and redundancy, we propose to use a rough-set based approach. After performing the ranking, features filtering is carried out in a supervised way enabling the user to decide on the number of the retained features. The experiments revealed that thanks to our method, it is possible to filter out numerous features describing data while still maintaining satisfactory classification accuracy achieved by the classifier trained using the reduced dataset. The comparative experiments performed with the use of publicly available datasets proved the high efficiency and competitiveness of our approach.

Keywords: feature ranking; feature selection; rough sets

1. Introduction

Describing data by numerous features usually leads to the necessity of making a plethora of measurements or examinations that might be laborious, computationally expensive and even in some cases hardly possible. Therefore, the problem of features selection became crucial for many applications [18]. Especially, when considering the classification task, feature selection is pivotal. In that case, the set of features should be reduced to such an extent that the classifier trained using the retained features ensures the required classification accuracy of new data instances.

Over the years, numerous automated methods have been developed to deal with the feature selection problem. One of the obstacles all features selection methods face is the high computational complexity. The issue is that finding an optimal set of features is a non-deterministic polynomial (NP) problem [11].

* Corresponding author. Tel.: +48-32-368-9764
E-mail address: wojciech.froelich@us.edu.pl

The second drawback of many methods is that the user loses control over the feature selection process. For example, this happens while generating feature subsets using population-based algorithms or by searching for them in a greedy way. In both cases, even if worthwhile from the point of view of the classification accuracy, the content of the obtained subset of features cannot be justified from the semantic, application point of view. The number of the considered features depends in that case merely on the classification accuracy achieved by the automated method, not on the conscious decision of the user.

For both of the above reasons, in this study, we aim to develop a new feature selection method assuring not only a low computational complexity but also the possibility of influencing the selection process by the user. By developing the method presented in this paper, we assist in fact the user in finding a sweet spot between the number of the selected features and the required classification accuracy.

Let us first note that it is possible to distinguish two main factors influencing the computational complexity of a feature selection method. The first one is the algorithm used for the generation of the candidate subsets of features. The second pivotal factor is the method used for the evaluation of that subset.

In fact, the goal of each feature selection algorithm is to produce subsequent subsets of features that are subject to evaluation. Obviously, the greater the number of subsets is generated, the more expensive from the computational point of view becomes the considered method. The lowest computational complexity is demonstrated by the ranker approach which in fact scores each feature individually and does not search through the subsets of features at all. The ranker evaluates each feature separately and then ranks all features according to that evaluation. Though, by using the ranker, we sacrifice to a certain extent the finally obtained classification accuracy while gaining the lowest possible computational complexity. Note that just the ranker approach enables the user to control the feature selection process. By having a list of the ranked attributes, it is possible to select those that ensure the classification performance required by the user confronted at the same time with the requirements imposed by a particular application.

Taking into account its benefits in terms of the computational complexity and the possibility to control the feature selection process, we decided to use for the purpose of this study the ranker approach, i.e., the method relying on the individual assessment of each feature.

As previously mentioned, having a subset of features is not everything. It is necessary to evaluate it. Also, in this case, a pivotal role is played by the computational complexity of the applied evaluation method.

The most computationally expensive is the wrapper (also called closed-loop) method [19]. That's because it evaluates subsets of attributes on the basis of feedback from the classification algorithm. This means that for each of the assessed subset of attributes, a classifier has to be learned. The obtained classification accuracy evaluates, in fact, the considered subset of attributes. The wrapper is deemed as the best feature evaluation method, assuming that its computational complexity is not an issue [6].

Slightly less computationally expensive are embedded methods that in fact combine the search for the optimal subset of features as a part of the classifier training [3].

The lowest computational complexity is demonstrated by filter methods [3]. In order to evaluate each candidate subset of features, the filter exploits merely the information contained within the data. In this case, none of the classifiers is learned, which results in lower computational cost. The cost paid for the lower complexity is, at least theoretically, the decreased classification accuracy exhibited by the classifier trained with the use of the reduced data.

For the purpose of our study, to maintain the lowest possible computational complexity of the ranker approach, we decided to combine it with feature filtering. The ranker method together with the filtering directed our further study towards the development of a feature selection method that is computationally efficient and transparent to the user. To achieve that goal, we performed a thorough literature review in which we focused on feature evaluation methods that could be used for the assessment of individual attributes in conjunction with filtering.

Firstly, we noted that many existing methods devoted to the selection of numerical features rely heavily on the discretization. It means that the discretization influences the performance of the considered feature selection method. Just that dependency requires further, in-depth investigation. Therefore, for the purpose of this study, we decided to consider merely data described by nominal features.

The first group of methods that fits our assumptions relies on entropy [6]. Among them is symmetrical uncertainty, which as its name suggests, evaluates the worth of every feature by measuring symmetrical uncertainty with respect to the class [13]. The information gain method evaluates an attribute by calculating its specifically designed score

with respect to the class [16]. The gain ratio, in turn, is the enhancement of the information gain calculated with the normalized score [15].

The Relief algorithm [9] and its extended version ReliefF [10] calculate the relevance of features to the class by randomly sampling instances from the dataset. Then, the Relief picks at random instances within the vicinity of the previously chosen instance. Depending on the class those instances belong, the near-hit and near-miss scores are calculated. On that basis, the feature weight vector is updated. In the case when dealing with nominal features, the Relief algorithm estimates the similarity between instances by the number of features on which the considered instances differ.

The correlation-based approach (CFS) [4, 6] calculates in its preprocessing step the Pearson's correlation of each feature with the decision. In addition, it creates a quadratic matrix, where each of its elements is the correlation between two features. On that basis, in its second step, the method evaluates a subset of features by calculating the merit. The merit averages the previously calculated relevancy and redundancies over the actually considered (by the assumed search method) subset of features. Nominal attributes are considered by treating each value as an indicator [4].

A plethora of feature selection methods is based on the theory of rough sets [12]. However, to the best of our knowledge, only a few of them were used for features ranking.

The traditional rough-set based approach used for the assessment of dependencies among features relies on the calculation of a positive region of a decision table [1]. The main limitation of that approach is that the positive region covers merely the consistent part of the data whereas in real-world cases the data are mostly inconsistent. Even if relaxing that limitation, the approach requires to provide a parameter, establishing a kind of threshold determining what part of data belongs to the positive region. Properly setting that parameter is obviously not straightforward. From the practical point of view, as shown in [17], the complexity of that approach is $O(|A|^2|U|^2)$, where $|A|$ denotes the number of conditional features and $|U|$ is the number of data instances.

A method relying on the construction of the discernibility matrix was proposed in [7]. The method assumes that the lower is the frequency of the features occurring in that matrix, the more important becomes the feature. The shorter is the length of the entry of the discernibility matrix; the more important become the features listed in that entry. Also in this case, the main limitation of the approach is that the discrimination matrix can be constructed only for the consistent decision tables, that is, as already mentioned, a rare case for real-world datasets. In consequence, the proposed method is suitable for the decision tables for which the so-called exact reduct exists. Just for that reason was this method not experimentally evaluated in our paper. The other distinctive feature of the method relying on the construction of discernibility matrix is the linear computational complexity with respect to the number of attributes and the quadratic one with respect to the number of instances in the dataset $O(|A||U|^2)$.

Recently, an approach based on rough sets theory and a Laplacian score has been used for the evaluation of features importance and then feature selection [20]. The method works in fact in two stages. In the first one, a number of features are chosen that cannot be reduced using the rough set-based reduct. In the second stage, for each selected attribute, a Laplacian score is calculated using the nearest neighbor graph with m data samples involved. The m is a parameter of the method. The main downside of that method can be recognized in the first step of the proposed algorithm, which works well only in the case the original decision table is consistent. For that reason, also, this method is not experimentally investigated in this paper. The method has linear complexity with respect to the number of features and quadratic complexity $O(|A||U|^2)$ with respect to the number of data instances considered in the nearest neighbor graph.

Also recently, a heuristic technique based on rough set theory has been proposed in [17]. The approach uses an intermediate grid representation for the calculation of inconsistent data instances. The method can be combined with, e.g.: particle swarm optimization, genetic algorithm or harmony search. Note that due to the application in our study of the ranker approach, the method is not suitable for our approach. As the authors claim, the complexity of the proposed method is $O(|A|^2|U|^2)$.

Note that the computational complexity of all of the above mentioned rough-set methods is quadratic with respect to the number of instances. The main advantage of the approach proposed in this paper is its low computational complexity which is linear with respect to the number of data instances and quadratic to the number of attributes, i.e., $O(|A|^2|U|)$.

A review and comparison of feature selection methods is available in [14, 2, 17].

The paper is organized in the following way. In Section 2 we provide the basics of rough set theory. Section 3 presents our contribution which is a new method for ranking and selecting nominal attributes. The results of comparative experiments aimed at the assessment of our method are presented in Section 4. Section 5 concludes.

2. Basics of rough set theory

The main role in rough set theory(RST) plays the notion of information system which is defined as an ordered pair $IS = (U, A)$, where U is a nonempty, final set of data instances (called in rough set terminology as objects). The nonempty, final set A consists of labels which are the names of the features (attributes). More formally, an attribute $a_i \in A$ is a mapping $a_i : U \rightarrow \mathrm{V}_{a_i}$, where V_{a_i} is the domain of a_i which is assumed as the set of symbolic values. The attributes of that type are called nominal or categorical.

The B-indiscernibility relation is defined as $IND_B = \{(u, u') \in U^2 : \forall a \in B, a(u) = a(u')\}$. This means that each instance $u \in U$ on the subset of attributes B determines certain indiscernibility class Q_j within U. When considering all indiscernibility classes, the set U is getting partitioned into the family $Q = \{Q_1, Q_2, ..., Q_m\}$, where m denotes the number of the indiscernibility classes.

A decision table $DT = (U, A, D)$ is an extension of information system which is complemented by $D \notin A$, a decision(class) attribute with the domain V_D. By assigning to each data instance described by A one of the class labels from V_D the decision table plays the role of a classifier. In the context of the decision table, the attributes from A are called conditions or conditional attributes. Every decision $d_l \in V_D$ determines the set of data instances $D_l = \{u \in U : D(u) = d_l\}$, which is called the decision class. We denote by $|V_D|$ the number of decision classes within U, i.e., the cardinality of the set V_D.

Note that each decision class $D_l \subseteq U$ is described by the indiscernibility classes Q_j, those that entirely belong to D_l, and the other that merely overlap with D_l. The sets $\underline{IND_B}(D_l) = \{u \in U : IND_B \subseteq D_l\}$ and $\overline{IND_B}(D_l) = \{u \in U : IND_B \bigcap D_l \neq \emptyset\}$ are called the lower and upper approximations of D_l, respectively. The set $BND(D_l) = \overline{IND_B}(D_l) - \underline{IND_B}(D_l)$ is called a boundary of D_l. Note that for the data instances within the lower approximation of the decision class, the assignment of decision is unique.

3. The proposed method

The feature selection method we propose in this paper consists of three major stages. The first one aims at assessing the relevance of each individual feature to the decision attribute. The second stage, in turn, evaluates the redundancy of each feature with respect to the other. In the third stage, we combine the previously calculated relevance and redundancy. On that basis, we produce the ranking of features. Afterward, assuming diverse numbers of features retained, we use 10-fold cross-validation technique to train a classifier and calculate its mean classification accuracy. Both the ranking and the related list of classification accuracies are shown to the user who selects the ultimate subset of features.

According to the above scheme, we partition the set of data instances U into the indiscernibility classes. We make that partitioning separately with respect to each i^{th} feature, i.e., by assuming $B = \{a_i\}$. This way, for each a_i, we obtain the set of indiscernibility classes $Q_i = Q_{i1}, Q_{i2}, ..., Q_{im}$, where m is its cardinality that obviously depends on the actually considered a_i.

After that, we assess whether it is possible to assign to each of the generated indiscernibility class Q_{ij} any of the decision classes D_l. That depends on the consistency (dispersion) of the decisions within Q_{ij}. To assess that, we calculate:

$$\mu_{D_l}(Q_{ij}) = \frac{|Q_{ij}| \bigcap |D_l|}{|Q_{ij}|}. \tag{1}$$

Formula (1) evaluates in fact how much the decision d_l is supported by the data instances from Q_{ij}. Furthermore, using formula (1), we are able to construct for each indiscernibility class Q_{ij}, a vector:

$$\mu_D(Q_{ij}) = \left\langle \mu_{D_1}(Q_{ij}), \mu_{D_2}(Q_{ij}), ..., \mu_{D_m}(Q_{ij}) \right\rangle. \tag{2}$$

In order to asses the consistency of the entire indiscernibility class Q_{ij}, irrespective of a particular decision. we propose to calculate the minimum and maximum values of the elements of the vector $\mu_D(Q_{ij})$:

$$\mu_{min}(Q_{ij}) = argmin_{D_l}\mu_D(Q_{ij}), \quad \mu_{max}(Q_{ij}) = argmax_{D_l}\mu_D(Q_{ij}). \tag{3}$$

Note that the higher is $\mu_{max}(Q_{ij})$ and the lower is $\mu_{min}(Q_{ij})$, the higher is getting the consistency of the indiscernibility class with respect to the decisions. Consequently, the dispersion of decisions within Q_{ij} is lower. By calculating: $\mu_{max}(Q_{ij}) - \mu_{min}(Q_{ij})$, we assess the consistency rate of the indiscernibility class Q_{ij}. On that basis, we are able to calculate the weight of the i^{th} attribute as:

$$w_i^{rel} = \frac{1}{m}\sum_{j=1}^{m}(\mu_{max}(Q_{ij}) - \mu_{min}(Q_{ij})), \tag{4}$$

where m denotes the number of indiscernibility classes generated by the attribute a_i.

Now we are able to rank all features according to the ascending order of w_i^{rel}. We obtain an ordered list of features:

$$R^{rel} = \left\langle r_1^{rel}, r_2^{rel}, ..., r_n^{rel} \right\rangle. \tag{5}$$

Note that r_1^{rel} is evaluated as the worst and $r_n^{rel}, n = |A|$ is assessed as the best one.

At the second stage of our method, we aim at evaluating the redundancy of each feature with respect to the rest of them. The higher the redundancy of a feature, the lower its impact on the classification accuracy is expected. To calculate that redundancy, we perform in fact a procedure that is similar to the previously described. The difference is that this time, we repeatedly replace the previously considered decision attribute by each of the considered features from A, except a_i that undergoes the evaluation. Consequently, for each pair of features a_i and a_k, where $i \neq k$, we calculate:

$$\mu_{Q_{kl}}(Q_{ij}) = \frac{|Q_{ij}| \bigcap |Q_{kl}|}{|Q_{ij}|}, \tag{6}$$

obtaining the vector:

$$\mu_{Q_k}(Q_{ij}) = \left\langle \mu_{Q_{k1}}(Q_{ij}), \mu_{Q_{k2}}(Q_{ij}), ..., \mu_{Q_{km}}(Q_{ij}) \right\rangle, \tag{7}$$

with its minimum and maximum:

$$\mu_{Q_k,min}(Q_{ij}) = argmin_{Q_{kl}}\mu_D(Q_{ij}), \quad \mu_{Q_k,max}(Q_{ij}) = argmax_{Q_{kl}}\mu_D(Q_{ij}). \tag{8}$$

After that, we are able to calculate the redundancy between features a_i and a_k as:

$$w_{ik}^{red} = \frac{1}{|m|}\sum_{j=1}^{m}(\mu_{max}(Q_{ij}) - \mu_{min}(Q_{ij})), \tag{9}$$

where m is the number of indiscernibility classes generated by attribute a_i.

To evaluate the mean redundancy of i^{th} attribute with respect to all of the other, i.e., independent of k, we calculate:

$$w_i^{red} = \frac{1}{|A| - 1}\sum_{k=1,k\neq i}^{|A|} w_{ik}^{red}. \tag{10}$$

At the final, third stage of our method, we combine the relevance with the redundancy of an attribute by calculating its ultimate weight:

$$w_i = \frac{w_i^{rel}}{w_i^{red}}. \tag{11}$$

By ranking features with respect to their weights (11), we produce an ordered list of features:

$$R = \langle r_1, r_2, ..., r_n \rangle . \tag{12}$$

Let us assume now that $1 \leq \alpha < n - 1$, $n = |A|$ is a number of features the user wants to retain. It means that $n - \alpha$ features from the left-hand side of the list R should be discarded and merely the remaining features $r_{n-\alpha+1}, r_{n-\alpha}, ..., r_n$ will be used to describe data from U.

After assigning a certain value to α, the user wants to assess what classification accuracy can be obtained by a classifier trained using the dataset described by merely α features. The greater α, the lower classification accuracy is expected. Therefore, choosing the ultimate value of α is, in fact, the main problem the user faces.

To assist the user in solving that problem, we partition the set of data instances into two disjoint subsets, namely $L \subset U$, and $T \subset U$, where $L \bigcup T = U$. The sets L and T are used for learning and testing the classifier, respectively. We assume that the cardinality of the learning set, i.e., $|L| = \frac{p-1}{p}|U|$, where $1 < p \leq 10$ is the parameter. To construct L we drew at random data instances from $|U|$. After learning the classifier using the data instances from L, we test it using the remaining set $T \subset U$, where $|T| = \frac{1}{p}|U|$. The partitioning process of U and the following learn and test trial is repeated p times. This way we perform in fact the $p - fold$ cross-validation of the classifier.

For each of the p trials, the classification accuracy is calculated as $\frac{|T_{correct}|}{|T|} \cdot 100$, where $|T_{correct}|$ is the number of correctly classified examples and $|T|$ is the number of instances in the testing set. As a result of the cross-validation, we are able to confront the ranking R (12) with the following list:

$$P = \langle acc_{\alpha=n-1}, acc_{\alpha=n-2}, ..., acc_{\alpha=n-k}, ..., acc_{\alpha=1} \rangle , \tag{13}$$

where $acc_{\alpha=x}$ denotes the mean classification accuracy calculated over all p trials whilst using x features for the description of data. Having the list P available, the user is able to confront different values of α with the related classification accuracies. On that basis, the user makes the ultimate decision regarding the number of features used for the description of data.

4. Case studies and comparative experiments

We stated two main goals for our experiments. The first one was the evaluation of the proposed method in terms of the classification accuracy. The second goal was to compare the effectiveness of our method with the competitive ones, those already known from the literature. In both cases, we repeated the experiments treating the number of the considered features as a parameter. Before starting out the experiments, we had to make several assumptions.

4.1. Experimental setup

Firstly, to make the obtained results easily repetitive by other researchers, we decided to use publicly available data from the UCI machine-learning repository. According to the previously made assumption, we selected merely the datasets with nominal attributes, i.e.: lenses, tic-tac-toe, flare, and zoo. In addition, we used a high-dimensional dataset on fake hotel reviews available at https://myleott.com/op-spam.html. We selected the datasets with the increasing number of features intending to make our experiments representative for different real-world problems.

To evaluate the classification accuracy, we decided to use the Naive Bayes classifier which was over the years most commonly used for the classification of data described by nominal features [4, 6, 8, 14, 21]. For all experiments, we used 10-fold cross-validation ($p = 10$), which we found as the most commonly used in the literature.

For the comparative experiments, we decided to use eight feature ranking methods. Six of them are the methods publicly available as a part of WEKA toolkit [5]. We described those methods in brief in the introduction to this paper.

We denoted all the considered methods by the following abbreviations: SU - symmetrical uncertainty, IG - information gain, GR - gain ratio, RL - ReliefF, PC - Pearson Correlation, T-RS - traditional rough-set [1], R-RS - the proposed method based on rough set based - relevance leading to the ranking (5), R2-RS - the proposed method combining relevance with redundancy leading to the ranking (12).

4.2. Results of experiments

We started experimenting with the lenses dataset which is described by merely $n = 4$ features. It can be noted in Table 1 that for $\alpha = 3$ both of the proposed methods R-RS and R2-RS turned out to be very competitive because they yielded at least the same results than the other methods. For $\alpha = 2$, the proposed R2-RS substantially outperformed all other methods. Note also that the R2-RS method outperformed in those cases both T-RS and R-RS. However, for $\alpha = 1$, i.e., for the single feature left, our method got substantially outperformed. This means that the rankings produced by R-RS and R2-RS were not suitable for the situation if the user would like to retain a single feature describing the data.

Table 1. Results for the lenses dataset

α	SU	IG	GR	RL	PC	T-RS	R-RS	R2-RS
3	83.33	83.33	83.33	75.0	75.0	83.33	83.33	83.33
2	87.5	87.5	87.5	87.5	75.0	87.5	87.5	91.67
1	70.83	70.83	70.83	70.83	70.83	70.83	50.0	54.17
	3 2 1 0	3 2 1 0	3 2 1 0	3 2 1 0	3 2 0 1	3 2 1 0	2 3 1 0	3 2 1 0

In the last row of Table 1, we provide the rankings of attributes produced by the considered methods. Due to the limited space, we were not able to present all 10 rankings produced by all of the trials performed within the $10 - fold$ cross-validation. The rankings given in the last row of Table 1 were produced using the entire data set, i.e., without partitioning it to the learning and testing parts. Note that this justifies the fact that the same rankings are associated with different mean classification accuracies.

Note also that when considering R-RS, which prefers feature a_2 over a_3, the method gives the user a substantially different alternative. In general, dependent on the value of α the user assumed, substantially different subsets of features can be selected.

Table 2. Results for the tic-tac-toe dataset

α	SU	IG	GR	RL	PC	T-RS	R-RS	R2-RS
8	69.83	69.83	69.83	70.46	70.46	70.56	69.31	70.25
7	69.31	69.31	69.83	69.62	70.25	69.42	69.1	69.42
6	69.52	69.52	69.73	68.68	69.94	71.5	70.25	70.31
5	69.62	69.73	69.62	69.1	69.62	71.29	69.31	69.94
4	69.21	69.31	68.68	68.79	68.79	65.87	69.52	69.42
3	68.58	69.0	68.58	68.68	68.06	65.34	69.62	69.73
2	68.79	69.21	68.58	68.79	68.16	65.34	69.94	69.94
1	69.94	69.94	69.94	69.94	69.94	65.34	69.94	69.94
SU:	4 8 6 2 0 5 1 3 7							
IG:	4 2 0 6 8 3 1 5 7							
GR:	4 8 6 2 0 5 1 3 7							
RL:	4 8 2 0 6 1 7 3 5							
PC:	4 8 5 3 1 2 6 0 7							
T-RS:	8 3 1 2 4 7 5 6 0							
R-RS:	4 2 0 6 7 1 5 3 8							
R-2RS:	4 6 2 0 3 8 1 7 5							

In the next experiment, we considered the tic-tac-toe data set described by $n = 9$ features. In Table 2, we provide classification accuracies obtained for $1 \leq \alpha < n - 1$. You can easily note, that in the case of this dataset our R-RS and R2-RS methods became more competitive for the lower number of the retained features, i.e., for $\alpha \leq 4$. The proposed R2-RS outperformed the other two rough set-based methods, namely T-RS and R-RS.

Also in this case, dependent on the number of features the user needs to retain, a different method might be preferred. Note that also for the tic-tac-toe dataset, independent of the value of α, the user is able to choose substantially different feature rankings and in consequence different subsets of features.

Table 3. Results for the flare dataset

α	SU	IG	GR	RL	PC	T-RS	R-RS	R2-RS
9	97.56	97.56	97.56	97.56	97.56	98.22	98.03	97.56
8	97.84	97.84	97.84	97.47	97.75	98.22	98.22	97.75
7	97.94	97.75	97.94	97.37	98.03	98.31	98.87	98.22
6	98.31	97.65	98.31	97.56	98.41	98.69	99.44	98.41
5	98.41	97.75	98.41	97.65	98.78	98.59	99.53	98.78
4	98.78	98.31	98.78	97.47	98.69	98.97	99.53	99.16
3	99.53	99.16	99.53	98.31	99.53	99.34	99.53	99.25
2	99.25	99.25	99.25	99.25	99.16	99.44	99.53	99.44
1	99.53	99.53	99.53	99.53	99.53	99.53	99.53	99.53

SU:	8 11 5 3 2 1 6 10 0 7 4 9
IG:	11 2 8 1 0 3 5 6 10 4 7 9
GR:	8 11 5 3 2 1 6 10 0 7 4 9
RL:	2 0 1 8 3 11 6 10 4 5 7 9
PC:	8 11 5 3 6 10 2 1 0 7 4 9
T-RS:	11 10 3 2 1 4 5 6 9 8 7 0
R-RS:	4 10 9 6 0 3 1 5 2 11 8
R2-RS:	5 3 4 6 10 9 8 2 1 7 0 11

Table 4. Results for the zoo dataset

α	SU	IG	GR	RL	PC	TRS	RRS	R2RS
16	95.05	95.05	95.05	95.05	95.05	95.05	95.05	95.05
15	96.04	96.04	96.04	95.05	95.05	95.05	94.06	94.06
14	96.04	96.04	95.05	95.05	96.04	93.07	94.06	94.06
13	95.05	95.05	96.04	96.04	96.04	92.08	96.04	94.06
12	95.05	95.05	95.05	92.08	97.03	92.08	95.05	94.06
11	94.06	94.06	95.05	92.08	92.08	87.13	94.06	94.06
10	94.06	93.07	95.05	94.06	94.06	83.17	93.07	94.06
9	92.08	92.08	90.06	90.1	91.09	81.19	92.08	93.07
8	90.1	90.1	87.13	90.07	90.1	82.18	87.13	92.08
7	86.14	86.14	87.13	87.13	87.13	83.17	85.15	89.11
6	86.14	86.14	89.11	86.14	87.13	79.21	84.16	88.12
5	87.13	87.13	83.17	86.14	86.14	70.3	86.14	89.11
4	86.14	87.13	83.17	76.24	76.24	64.36	84.16	86.14
3	84.16	84.16	82.18	70.3	64.36	51.49	79.21	85.2
2	84.16	73.27	70.3	61.39	60.4	47.52	63.37	71.29
1	73.27	40.59	52.48	60.4	60.4	40.59	40.59	60.4

SU:	13 4 0 8 3 1 2 9 10 14 12 5 6 16 11 7 15
IG:	0 13 4 8 3 1 2 9 10 14 5 12 6 16 11 7 15
GR:	4 2 9 8 3 10 1 12 13 14 5 6 0 11 16 7 15
RL:	4 3 8 1 2 9 10 14 5 12 16 6 13 7 11 0 15
PC:	4 3 1 8 2 10 5 9 16 14 6 13 12 11 15 7 0
TRS:	0 11 5 4 3 7 2 1 6 16 15 13 14 12 9 10 8
TRS:	0 2 13 4 3 1 12 8 5 10 9 15 16 6 14 7 11
R2RS:	2 8 4 12 3 5 10 13 1 15 9 0 6 7 14 16 11

The results obtained for the flare dataset are given in Table 3. This time, the R-RS method turned out to be the best for almost all values of α. What is, however, worth noting, the classification accuracies produced by alternative

methods variate very slightly and are quite high. This means that the flare dataset has an extremely high potential for feature reduction. Due to the diverse ranking approaches applied, the considered methods produced substantially different orders of features. This, in turn, resulted in a huge benefit to the user, the availability of different rankings while still maintaining good classification accuracy, independent of the number of the retained features.

The next file we investigated contained 17 nominal features. The results obtained for the zoo dataset are given in Table 4. As can be noted, the obtained accuracies and the winning method substantially depend on the number of the retained features. For $\alpha \geq 10$, the produced results are quite similar to each other with the excerp of those produced by T-RS. An evident superiority of our R2-RS method can be observed for the lower number of the retained features, i.e. within the range $3 \leq \alpha \leq 9$. For $1 \leq \alpha \leq 2$ the R2-RS method outperformed the rough set-based T-RS and R-RS but became worse than SU.

The last dataset we experimented with was named hotel. It contained 128 features representing the binary weights of words with the highest frequency of appearance. Due to the huge amount of the produced results, we present in Table 5 only an exception from them. For the high number of features retained, i.e., for $\alpha > 70$ the results yielded by all methods turned out to be similar to each other. The R-RS and R2-RS methods became really good and almost always outperformed other methods for $10 \leq \alpha \leq 70$. Unfortunately, for the lower number of features retained all rough set-based methods worked poorly.

Table 5. Results for the hotel dataset

α	SU	IG	GR	RL	PC	T-RS	R-RS	R2-RS
127	79.25	79.25	79.25	79.5	79.25	79.63	79.25	79.25
126	79.25	79.25	79.25	79.38	79.25	79.88	79.25	79.25
125	79.25	79.25	79.25	79.13	79.25	79.75	79.25	79.25
...	...	...	...	...	...	...	...	...
70	79.88	79.88	80.13	77.75	79.88	77.38	80.25	80.13
69	80.0	79.5	80.38	77.88	79.5	77.38	80.5	80.5
68	79.88	80.13	80.38	78.38	80.13	77.38	80.5	80.5
67	80.13	79.5	80.13	78.38	79.63	76.88	80.5	80.25
66	79.88	79.75	80.5	78.38	79.75	76.63	80.38	80.13
...	...	...	...	...	...	...	...	...
30	77.13	77.25	77.13	75.75	77.25	69.25	78.13	78.13
29	76.75	77.38	77.88	75.5	77.38	67.75	78.13	78.13
28	77.0	77.5	78.13	75.38	78.13	68.13	77.75	78.0
27	77.63	79.25	77.13	74.88	79.0	68.25	77.5	77.5
26	76.13	77.88	77.88	75.5	77.88	67.63	78.5	78.75
...	...	...	...	...	...	...	...	...
5	68.63	70.0	67.63	69.5	70.38	61.5	67.5	67.75
4	67.5	69.75	67.38	68.13	70.63	61.88	67.38	67.38
3	65.38	66.88	63.38	67.38	66.38	54.75	63.13	62.75
2	65.25	63.5	59.63	67.0	63.25	55.25	57.88	59.13
1	56.0	62.5	54.5	60.63	62.5	51.25	52.88	54.0

SU:	33 3 4 2 71 22 55 8 1 0 53 60 116 25 111 106 15 110 103 115 126 41 123 67 11 42 77 91 100 113
IG:	3 4 33 55 8 22 2 71 1 60 53 15 0 25 116 106 67 111 110 41 103 115 126 123 42 11 13 66 44 77
GR:	33 0 71 2 22 3 8 1 4 53 116 55 111 60 25 106 103 115 110 126 123 100 41 15 11 77 91 67 92 113
RL:	3 55 22 42 4 8 19 15 71 1 33 115 116 60 69 110 20 7 77 109 44 28 2 68 118 57 89 25 11 37
PC:	3 4 33 55 8 22 2 71 1 60 53 15 25 0 116 67 106 110 111 41 115 103 126 123 42 11 13 66 44 77
T-RS:	127 126 41 42 43 44 45 40 39 38 34 33 35 37 36 46 47 48 59 58 60 56 61 57 55 49 51 50 52 54
R-RS:	0 33 71 2 22 1 53 8 116 3 4 111 55 106 60 25 103 115 126 110 123 100 77 11 41 91 15 92 114 113
R2-RS:	0 33 71 2 22 1 53 8 116 4 111 3 55 60 25 103 106 115 126 110 123 100 41 77 11 91 15 92 114 113

4.3. Conclusions coming from the experiments

As the results of the experiments revealed, our R-RS and R2-RS methods belong to the most competitive ranking approaches available. It became clear that depending on the properties of the data, the user faces the problem of

selecting the best method. The classification accuracy obtained while using our methods depends heavily on the considered dataset and on the number of attributes the user wants to retain. The only possibility to deal with that issue is to perform computational experiments similar to those presented in our paper. This way, the user is able to select the best method for a given dataset and for the preferred number of the retained attributes. For that purpose, we definitely recommend considering our R-RS and R2-RS methods that, as the experiments revealed, are very competitive.

5. Final remarks

In this paper, we proposed a new method that is used for the ranking and selection of nominal features. The proposed method is one of the most efficient ones exhibiting very low computational complexity which is linear with respect to the number of data instances and quadratic with respect to the number of features. Moreover, due to the application of the ranker approach, our method enables the influence of the feature selection process by the user. Along with the above advantages, using the approach proposed in this paper ensures satisfactory classification accuracy, especially in comparison with the other competitive methods. A limitation of the proposed approach is the expected lower classification accuracy in comparison to the methods relying on greedy or population-based feature selection methods. The estimation of that is however a challenge for future research.

Acknowledgements

This article was supported by the scientific research project of the Czech Sciences Foundation Grant No: 19-15498S.

References

[1] Chouchoulas, A., Shen, Q., 2001. Rough set-aided keyword reduction for text categorization. Applied Artificial Intelligence 15, 843–873.
[2] Cilia, N., De Stefano, C., Fontanella, F., Scotto di Freca, A., 2019. A ranking-based feature selection approach for handwritten character recognition. Pattern Recognition Letters 125, 77–86.
[3] Derrac, J., Cornelis, C., Garca, S., Herrera, F., 2012. Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. Information Sciences 186, 73 – 92.
[4] Hall, M., 2000. Correlation-based feature selection for machine learning. Department of Computer Science 19.
[5] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: An update. SIGKDD Explor. Newsl. 11, 10–18.
[6] Hall, M.A., Holmes, G., 2003. Benchmarking attribute selection techniques for discrete class data mining. IEEE Trans. on Knowl. and Data Eng. 15, 1437–1447.
[7] Hu, K., Lu, Y., Shi, C., 2003. Feature ranking in rough sets. AI Commun. 16, 41–50.
[8] Karimi, Z., Riahi Kashani, M., Harounabadi, A., 2013. Feature ranking in intrusion detection dataset using combination of filtering methods. International Journal of Computer Applications 78, 21–27.
[9] Kira, K., Rendell, L.A., 1992. A practical approach to feature selection, in: Sleeman, D.H., Edwards, P. (Eds.), Ninth International Workshop on Machine Learning, Morgan Kaufmann. pp. 249–256.
[10] Kononenko, I., 1994. Estimating attributes: Analysis and extensions of relief, in: Bergadano, F., De Raedt, L. (Eds.), Machine Learning: ECML-94, Springer Berlin Heidelberg. pp. 171–182.
[11] Masoudi-Sobhanzadeh, Y., Motieghader, H., Masoudi-Nejad, A., 2019. Featureselect: a software for feature selection based on machine learning approaches. BMC Bioinformatics 20, 170.
[12] Pawlak, Z., 1981. Information systems – theoretical foundations. Information Systems 6, 205–218.
[13] Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: Criteria of max-dependency,max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–38.
[14] Phyu, Thu Zar, Oo, Nyein Nyein, 2016. Performance comparison of feature selection methods. MATEC Web of Conferences 42, 06002.
[15] Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1, 81–106.
[16] Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
[17] Raza, M., Qamar, U., 2018. A heuristic based dependency calculation technique for rough set theory. Pattern Recognition 81, 309–325.
[18] Shen, Q., Jensen, R., 2007. Rough sets, their extensions and applications. International Journal of Automation and Computing 4, 217–228.
[19] Swiniarski, R.W., 2001. Rough sets methods in feature reduction and classification. International Journal of Applied Mathematics and Computer Science 11, 565–582.
[20] Yu, S., Zhao, H., 2018. Rough sets and laplacian score based cost-sensitive feature selection. PLOS ONE 13, 1–23.
[21] Yun, C., Yang, J., 2007. Experimental comparison of feature subset selection methods, in: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, pp. 367–372.