Yugoslav Journal of Operations Research
21 (2011), Number 1, 119-135
DOI: 10.2298/YJOR1101119N

TOWARD OPTIMAL FEATURE SELECTION USING RANKING METHODS AND CLASSIFICATION ALGORITHMS

Jasmina NOVAKOVIĆ, Perica STRBAC, Dusan BULATOVIĆ

Faculty of Computer Science,
Megatrend University, Serbia
jnovakovic@megatrend.edu.rs

Received: April 2009 / Accepted: March 2011

Abstract: We presented a comparison between several feature ranking methods used on two real datasets. We considered six ranking methods that can be divided into two broad categories: statistical and entropy-based. Four supervised learning algorithms are adopted to build models, namely, IB1, Naive Bayes, C4.5 decision tree and the RBF network. We showed that the selection of ranking methods could be important for classification accuracy. In our experiments, ranking methods with different supervised learning algorithms give quite different results for balanced accuracy. Our cases confirm that, in order to be sure that a subset of features giving the highest accuracy has been selected, the use of many different indices is recommended.

Keywords: Feature selection, feature ranking methods, classification algorithms, classification accuracy.

MSC: 90B50, 62C99

1. INTRODUCTION

Feature selection can be defined as a process that chooses a minimum subset of M features from the original set of *N* features, so that the feature space is optimally reduced according to a certain evaluation criterion. As the dimensionality of a domain expands, the number of feature *N* increases. Finding the best feature subset is usually intractable [1] and many problems related to feature selection have been shown to be NP-hard [2].

Feature selection is an active field in computer science. It has been a fertile field of research and development since 1970s in statistical pattern recognition [3, 4, 5], machine learning and data mining [6, 7, 8, 9, 10, 11].

Feature selection is a fundamental problem in many different areas, especially in forecasting, document classification, bioinformatics, and object recognition or in modelling of complex technological processes [12, 13, 14, 15]. Datasets with thousands of features are not uncommon in such applications. All features may be important for some problems, but for some target concepts, only a small subset of features is usually relevant.

Feature selection reduces the dimensionality of feature space, removes redundant, irrelevant, or noisy data. It brings the immediate effects for application: speeding up a data mining algorithm, improving the data quality and thereof the performance of data mining, and increasing the comprehensibility of the mining results.

Feature selection algorithms may be divided into filters [16, 17], wrappers [1] and embedded approaches [6]. Filters methods evaluate quality of selected features, independent from the classification algorithm, while wrapper methods require application of a classifier (which should be trained on a given feature subset) to evaluate this quality. Embedded methods perform feature selection during learning of optimal parameters (for example, neural network weights between the input and the hidden layer).

Some classification algorithms have inherited the ability to focus on relevant features and ignore irrelevant ones. Decision trees are a primary example of a class of such algorithms [18, 12]; but also multi-layer perceptron (MLP) neural networks, with strong regularization of the input layer, may exclude the irrelevant features in an automatic way [19]. Such methods may also benefit from independent feature selection. On the other hand, some algorithms have no provisions for feature selection. The k-nearest neighbour algorithm is a family of such methods that classify novel examples by retrieving the nearest training example, strongly relying on feature selection methods to remove noisy features.

Researchers have studied the various aspects of feature selection. Search is a key topic in the study of feature selection [13], such as search starting points, search directions, and search strategies. Another important aspect is how to measure the goodness of a feature subset [13]. There are filter methods [5, 20, 21], wrapper methods [22, 23, 8] and recently, hybrid methods [10, 24]. According to class information availability in data, there are supervised feature selection approaches [25, 7] as well as unsupervised feature selection approaches [26, 27, 14, 8].

The main aim of this paper was to experimentally verify the impact of different, entropy-based and statistical classifiers on classification accuracy. We have shown that there is no best ranking index for different datasets and different classifiers accuracy curves, as the function of the number of features used may significantly differ. The only way to be sure that the highest accuracy is obtained in practical problems is testing a given classifier on a number of feature subsets, obtained from different ranking indices.

The paper is organized as follows. In the next section we briefly described general architecture for the most of the feature selection algorithms. Section 3 contains general issues concerning diverse feature ranking and feature selection techniques. Section 4 gives a brief overview of adopted algorithms, namely, IB1, Naive Bayes, C4.5 decision tree and the radial basis function (RBF) network. Section 4 presents experimental evaluation. Final section contains discussion of the obtained results, some closing remarks, and issues that remain to be addressed and that we intend to investigate in future work.

2. GENERAL FEATURE SELECTION STRUCTURE

It is possible to derive a general architecture from most of the feature selection algorithms. It consists of four basic steps (refer to Figure 1): subset generation, subset evaluation, stopping criterion, and result validation [7]. The feature selection algorithms create a subset, evaluate it, and loop until an ending criterion is satisfied [15]. Finally, the subset found is validated by the classifier algorithm on real data.

Subset Generation Subset generation is a search procedure; it generates subsets of features for evaluation. The total number of candidate subsets is 2^N, where N is the number of features in the original data set, which makes exhaustive search through the feature space infeasible with even moderate N. Non-deterministic search like evolutionary search is often used to build the subsets [28]. It is also possible to use heuristic search methods. There are two main families of these methods: *forward addition* [29] (starting with an empty subset, we add features after features by local search) or *backward elimination* (the opposite).

Subset Evaluation Each subset generated by the generation procedure needs to be evaluated by a certain evaluation criterion and compared with the previous best subset with respect to this criterion. If it is found to be better, then it replaces the previous best subset. A simple method for evaluating a subset is to consider the performance of the classifier algorithm when it runs with that subset. The method is classified as a *wrapper,* because in this case, the classifier algorithm is wrapped in the loop. In contrast, *filter* methods do not rely on the classifier algorithm, but use other criteria based on correlation notions.

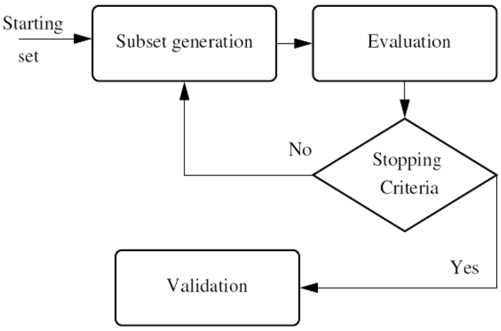

Figure 1: General feature selection structure

Stopping criteria Without a suitable stopping criterion, the feature selection process may run exhaustively before it stops. A feature selection process may stop under one of the following reasonable criteria: (1) a predefined number of features are selected, (2) a predefined number of iterations are reached, (3) in case addition (or deletion) of a feature fails to produce a better subset, (4) an optimal subset according to the evaluation criterion is obtained.

Validation The selected best feature subset needs to be validated by carrying out different tests on both the selected subset and the original set and comparing the results using artificial data sets and/or real-world data sets.

3. FEATURE RANKING AND SELECTION

Diverse feature ranking and feature selection techniques have been proposed in the machine learning literature. The purpose of these techniques is to discard irrelevant or redundant features from a given feature vector. For the purpose of this experiment, we used feature ranking and selection methods with two basic steps of general architecture: subset generation and subset evaluation for the ranking of each feature in every dataset. Filter method was used to evaluate each subset.

In this paper, we consider evaluation of the practical usefulness of the following ranking, commonly used methods, statistical and entropy-based, with good performance in various domains:

- Information Gain (IG) attribute evaluation,
- Gain Ratio (GR) attribute evaluation,
- Symmetrical Uncertainty (SU) attribute evaluation,
- Relief-F (RF) attribute evaluation,
- One-R (OR) attribute evaluation,
- Chi-Squared (CS) attribute evaluation.

Entropy is commonly used in the information theory measure [30], which characterizes the purity of an arbitrary collection of examples. It is in the foundation of the IG, GR, and SU attribute ranking methods. The entropy measure is considered a measure of the system's unpredictability. The entropy of Y is

$$H(Y) = -\sum_{y \in Y} p(y) \log_2(p(y)) \tag{1}$$

where *p(y)* is the marginal probability density function for the random variable *Y*. If the observed values of *Y* in the training data set *S* are partitioned according to the values of a second feature *X*, and the entropy of *Y* with respect to the partitions induced by *X* is less than the entropy of *Y* prior to partitioning, then there is a relationship between features *Y* and *X*. The entropy of *Y* after observing *X* is then:

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2(p(y|x)) \tag{2}$$

where *p(y |x)* is the conditional probability of *y* given *x*.

3.1. Information Gain

Given the entropy is a criterion of impurity in a training set *S*, we can define a measure reflecting additional information about *Y* provided by *X* that represents the amount by which the entropy of *Y* decreases [31]. This measure is known as IG. It is given by

$$IG = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{3}$$

IG is a symmetrical measure (refer to equation (3)). The information gained about *Y* after observing *X* is equal to the information gained about *X* after observing *Y.* A weakness of the IG criterion is that it is biased in favor of features with more values even when they are not more informative.

3.2. Gain Ratio

The Gain Ratio is the non-symmetrical measure that is introduced to compensate for the bias of the IG [31]. GR is given by

$$GR = \frac{IG}{H(X)} \tag{4}$$

As equation (4) presents, when the variable *Y* has to be predicted, we normalize the IG by dividing by the entropy of *X,* and vice versa. Due to this normalization, the GR values always fall in the range [0, 1]. A value of GR = 1 indicates that the knowledge of *X* completely predicts *Y,* and GR = 0 means that there is no relation between *Y* and *X.* In opposition to IG, the GR favors variables with fewer values.

3.3. Symmetrical Uncertainty

The Symmetrical Uncertainty criterion compensates for the inherent bias of IG by dividing it by the sum of the entropies of X and Y [31]. It is given by

$$SU = 2\frac{IG}{H(Y) + H(X)} \tag{5}$$

SU takes values, which are normalized to the range [0, 1] because of the correction factor 2. A value of SU = 1 means that the knowledge of one feature completely predicts, and the other SU = 0 indicates, that *X* and *Y* are uncorrelated. Similarly to GR, the SU is biased toward features with fewer values.

3.4. Chi-Squared

Feature Selection via chi square (X^2) test is another, very commonly used method [32]. Chi-squared attribute evaluation evaluates the worth of a feature by computing the value of the chi-squared statistic with respect to the class. The initial hypothesis H_0 is the assumption that the two features are unrelated, and it is tested by chi-squared formula:

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(Oij - Eij)^2}{Eij} \tag{6}$$

where O_{ij} is the observed frequency and E_{ij} is the expected (theoretical) frequency, asserted by the null hypothesis. The greater the value of χ^2, the greater the evidence against the hypothesis H_0 is.

3.5. One-R

OneR is a simple algorithm proposed by Holte [33]. It builds one rule for each attribute in the training data and then selects the rule with the smallest error. It treats all numerically valued features as continuous and uses a straightforward method to divide the range of values into several disjoint intervals. It handles missing values by treating "missing" as a legitimate value.

This is one of the most primitive schemes. It produces simple rules based on one feature only. Although it is a minimal form of classifier, it can be useful for determining a baseline performance as a benchmark for other learning schemes.

3.6. Relief-F

Relief-F attribute evaluation [34], evaluates the worth of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different class. This attribute evaluation assigns a weight to each feature based on the ability of the feature to distinguish among the classes, and then selects those features whose weights exceed a user-defined threshold as relevant features. The weight computation is based on the probability of the nearest neighbors from two different classes having different values for a feature and the probability of two nearest neighbors of the same class having the same value of the feature. The higher the difference between these two probabilities, the more significant is the feature. Inherently, the measure is defined for a two-class problem, which can be extended to handle multiple classes, by splitting the problem into a series of two-class problems.

4. CLASSIFICATION ALGORITHMS

Methods of ranking rank each feature in the dataset. The results were validated using different algorithms for classification. A wide range of classification algorithms is available, each with its strengths and weaknesses. There is no single learning algorithm that works best on all supervised learning problems. Four widely used supervised learning algorithms are adopted here to build models, namely, IB1, Naive Bayes, C4.5 decision tree and the radial basis function (RBF) network. The advantage of IB1 is that they are able to learn quickly from a very small dataset. An advantage of Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. C4.5 decision tree has various advantages: simple to understand and interpret, requires little data preparation, robust, performs well with large data in a short time. RBF network offers a number of advantages, including requiring less formal statistical training, ability to implicitly detect complex nonlinear relationships between dependent and independent variables, ability to detect all possible interactions between predictor variables, and the availability of multiple training algorithms. This section gives a brief overview of these algorithms.

4.1. IB1

IB1 is nearest neighbour classifier. It uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used. Nearest neighbour is one of the simplest learning/classification algorithms, and has been successfully applied to a broad range of problems [35].

To classify an unclassified vector X, this algorithm ranks the neighbours of X amongst a given set of N data (Xi, ci), $i = 1, 2, ..., N$, and uses the class labels cj ($j = 1, 2, ..., K$) of the K most similar neighbours to predict the class of the new vector X. In particular, the classes of these neighbours are weighted using the similarity between X and each of its neighbours, where similarity is measured by the Euclidean distance metric. Then, X is assigned the class label with the greatest number of votes among the K nearest class labels.

The nearest neighbour classifier works based on the intuition that the classification of an instance is likely to be most similar to the classification of other instances that are nearby within the vector space. Compared to other classification methods such as Naive Bayes, nearest neighbour classifier does not rely on prior probabilities, and it is computationally efficient if the data set concerned is not very large. However, if the data sets are large, each distance calculation may become quite expensive. This reinforces the need for employing PCA and information gain-based feature ranking to reduce data dimensionality, in order to reduce the computation cost.

4.2. Naive Bayes

This classifier is based on the elementary Bayes' Theorem. It can achieve relatively good performance on classification tasks [36]. Naive Bayes classifier greatly simplifies learning by assuming that features are independent given the class variable. More formally, this classifier is defined by discriminant functions:

$$fi(X) = \prod_{j=1}^{N} P(xj|ci)P(ci) \tag{7}$$

where $X = (x_1, x_2, ..., x_N)$ denotes a feature vector and c_j , $j = 1, 2, ..., N$, denote possible class labels.

The training phase for learning a classifier consists of estimating conditional probabilities $P(xj|ci)$ and prior probabilities $P(c_i)$. Here, $P(c_i)$ are estimated by counting the training examples that fall into class ci and then dividing the resulting count by the size of the training set. Similarly, conditional probabilities are estimated by simply observing the frequency distribution of feature x_j within the training subset that is labeled as class c_i. To classify a class-unknown test vector, the posterior probability of each class is calculated, given the feature values present in the test vector; and the test vector is assigned to the class that is of the highest probability.

4.3. C4.5 Decision Tree

Different methods exist to build decision trees, but all of them summarize given training data in a tree structure, with each branch representing an association between feature values and a class label. One of the most famous and most representative amongst these is the C4.5 tree [37]. The C4.5 tree works by recursively partitioning the training data set according to tests on the potential of feature values in separating the classes. The decision tree is learned from a set of training examples through an iterative process of choosing a feature and splitting the given example set according to the values of that feature. The most important question is which of the features is the most influential in determining the classification and hence should be chosen first. Entropy measures or equivalently, information gains are used to select the most influential, which is intuitively deemed to be the feature of the lowest entropy (or of the highest information gain). This learning algorithm works by: a) computing the entropy measure for each feature, b) partitioning the set of examples according to the possible values of the feature that has the lowest entropy, and c) estimating probabilities, in a way exactly the same as with the Naive Bayes approach. Note that although feature tests are chosen one at a time in a greedy manner, they are dependent on results of previous tests.

4.4. RBF Network

A popular type of feed forward network is RBF network. RBF network has two layers, not counting the input layer. Each hidden unit essentially represents a particular point in input space, and its output, or activation, for a given instance depends on the distance between its point and the instance—which is just another point. Intuitively, the closer these two points are, the stronger is the activation. This is achieved by using a nonlinear transformation function to convert the distance into a similarity measure. A bell-shaped Gaussian activation function, whose width may be different for each hidden unit, is commonly used for this purpose. The hidden units are called RBFs because the points in instance space, for which a given hidden unit produces the same activation, form a hypersphere or hyperellipsoid.

The output layer of an RBF network takes a linear combination of the outputs of the hidden units and—in classification problems—pipes it through the sigmoid function. The parameters that such a network learns are: (a) the centers and widths of the RBFs and (b) the weights used to form the linear combination of the outputs obtained from the hidden layer.

One way to determine the first set of parameters is to use clustering, without looking at the class labels of the training instances at all. The simple k-means clustering algorithm can be applied, clustering each class independently to obtain k basis functions for each class. Intuitively, the resulting RBFs represent prototype instances. Afterwards, the second set of parameters can be learned, keeping the first parameters fixed. This involves learning a linear model using one of the techniques such as linear or logistic regression. If there are far fewer hidden units than training instances, this can be done very quickly.

A disadvantage of RBF networks is that they give the same weight for every feature because all are treated equally in the distance computation. Hence, they cannot deal effectively with irrelevant features.

5. EXPERIMENTS AND RESULTS

Real datasets called "Statlog (Australian Credit Approval)" and "Statlog (German Credit Data)" were used for tests, taken from the UCI repository of machine learning databases [38]. These datasets were used to compare different feature ranking and feature selection methods on data.

German Credit Data

This dataset classifies people described by a set of features as good or bad credit risks. Data set characteristics is multivariate, feature characteristics are categorical and integer. Number of instances is 1000, number of features is 20, and there are no missing values.

Table 1: Results of ranking methods on German credit dataset

German Credit Data - Ref	IG	GR	SU	CS	OR	RF
1- checking_status	1	1	1	1	3	1
2 - duration	3	20	3	3	2	3
3 - credit_history	2	3	2	2	9	4
4 - purpose	6	2	5	6	11	6
5 - credit_amount	4	5	6	4	10	7
6 - savings_status	5	6	13	5	6	9
7 - employment	12	13	4	12	4	12
8 - installment_commitment	7	15	15	7	8	8
9 - personal_staus	15	14	12	15	7	19
10 - other_parties	13	4	20	13	18	2
11 - residence_since	14	10	14	14	17	14
12 - property_magnitude	9	12	7	9	20	10
13 - age	20	7	10	20	19	13
14 - other_payment_plans	10	9	9	10	14	18
15 - housing	17	19	17	17	12	17
16 - existing_credits	19	17	19	19	15	11
17 - job	18	18	18	18	16	5
18 - num_dependents	8	8	8	8	13	16
19 - own_telephone	16	16	16	16	1	15
20 - foreign_worker	11	11	11	11	5	20

Australian Credit Approval

This file concerns credit card applications. Data set characteristics is multivariate; feature characteristics are categorical, integer and real. Number of instances is 690, number of features is 14, and there are missing values. This dataset is interesting because there is a good mix of features – continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values.

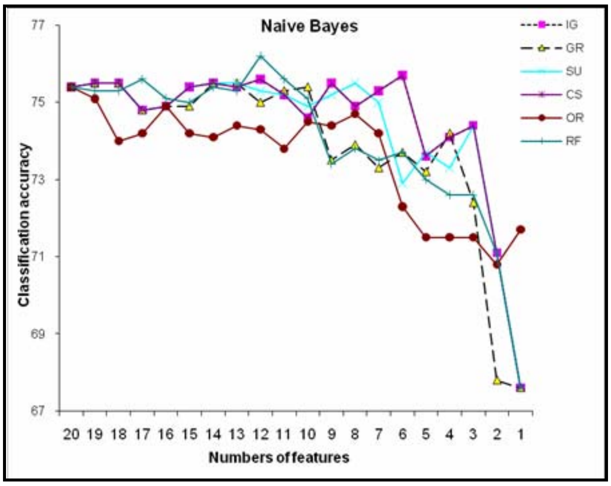

Figure 2: Ranking methods and balanced classification accuracy for German credit dataset, Naive Bayes classifier

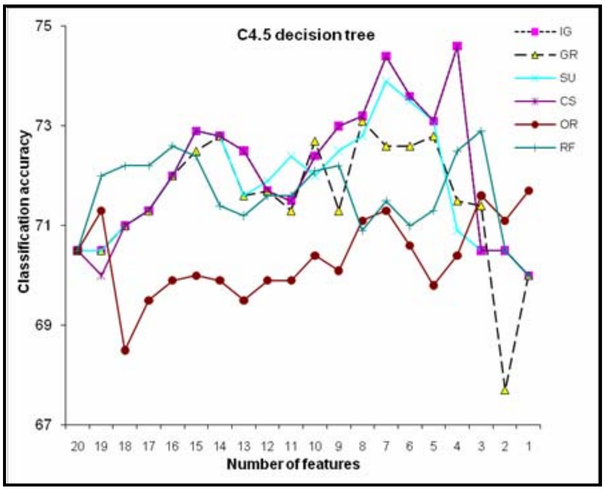

Figure 3: Ranking methods and balanced classification accuracy for German credit dataset, C4.5 decision tree classifier

The datasets described above have been used in tests. Six ranking methods have been used in each case: CS, OR, RF, IG, GR and SU.

The ranking of features obtained for the training data is presented in Table 1 and Table 2.

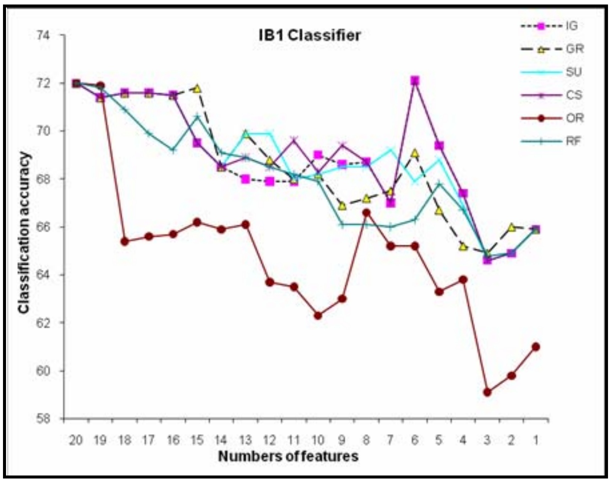

Figure 4: Ranking methods and balanced classification accuracy for German credit dataset, IB1 classifier

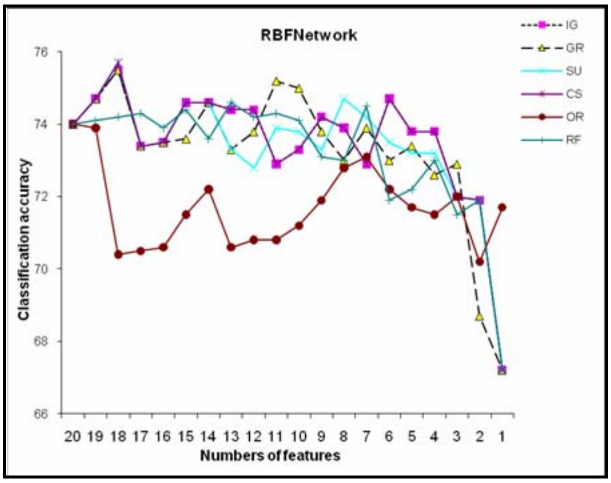

Figure 5: Ranking methods and balanced classification accuracy for German credit dataset, RBF network

Significant differences are observed in the order of the features in different ranking methods on German credit dataset. The first feature ranked as top (1) is same in different ranking methods, except OR. The last 4 features ranked as bottom are the same in different ranking methods based on entropy indices and CS: 11, 16, 8, and 18.

Table 2: Results of ranking methods on Australian credit approval dataset

Australian Credit Ap. - Ref	IG	GR	SU	CS	OR	RF
A1	9	9	9	9	9	9
A2	11	10	11	11	10	6
A3	10	11	10	10	11	7
A4	15	15	15	8	15	10
A5	8	8	8	15	8	12
A6	6	3	6	6	6	1
A7	14	5	14	14	7	5
A8	7	4	3	7	14	4
A9	3	14	7	3	4	3
A10	4	6	4	4	5	2
A11	5	7	5	5	13	8
A12	2	2	2	2	1	13
A13	13	13	13	13	12	11
A14	12	12	12	12	2	14
A15	1	1	1	1	3	15

On Australian credit approval dataset, the first feature ranked as top is the same in different ranking methods: 9. Less difference is observed in the order of the features in different ranking methods on this dataset, compared to German credit dataset. But, we have the same result with irrelevant features on both datasets. On Australian credit approval dataset, the last 4 features ranked as bottom are the same in different ranking methods based on entropy indices and CS: 1, 12, 13, 2.

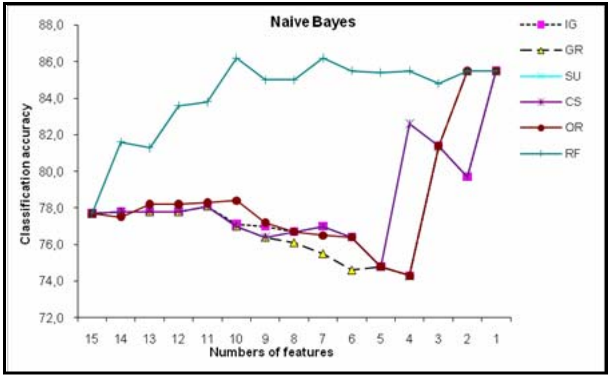

Figure 6: Ranking methods and balanced classification accuracy for Australian credit approval dataset, Naive Bayes classifier

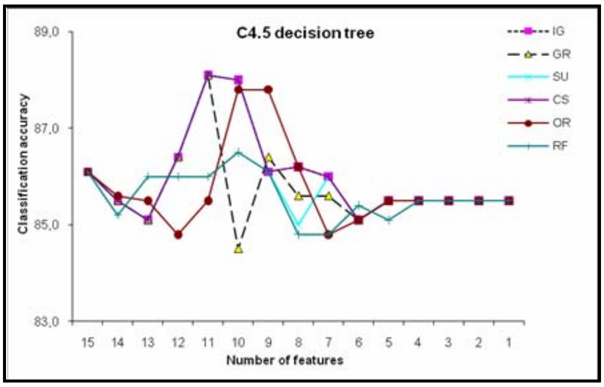

Figure 7: Ranking methods and balanced classification accuracy for Australian credit approval dataset, C4.5 decision tree classifier

Investigation of classification accuracy of the test data as a function of the n best features has been done for each ranking method. Four classifiers were used: Naive Bayes, C4.5 decision tree, IB1 classifier and RBF Network. Naive Bayes, C4.5 decision tree, IB1 classifier give deterministic results, simplifying the comparison (in contrast to these methods, neural classifiers give slightly different results after each restart).

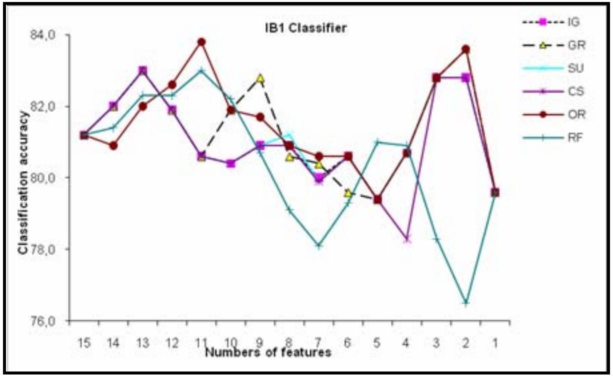

Figure 8: Ranking methods and balanced classification accuracy for Australian credit approval dataset, IB1 classifier

The same calculations were performed for both datasets. First, the ranking algorithms were applied to the whole dataset, and classification accuracy was estimated using ten-fold crossvalidation. For the purpose of comparing different ranking methods this approach is sufficient, producing one ranking for each index. As for real application, this could lead to some overfitting, therefore ranking and classification should be done separately for each training partition. Good generalization may be obtained by selecting only those features that were highly ranked in all data partitions.

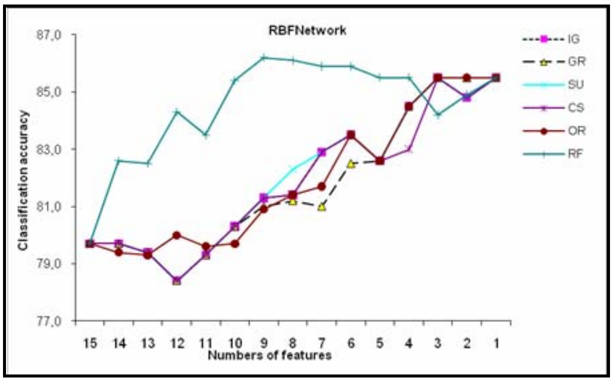

Figure 9: Ranking methods and balanced classification accuracy for Australian credit approval dataset, RBF network

Classification results for German credit dataset are presented in Figure 2, to Figure 5. Classification accuracy for German credit dataset is influenced by the choice of ranking indices. Unfortunately, OR ranking method gives very similar poor results for balanced accuracy with all classifier, expecially IB1, C4.5 decision tree and RBF network. Others ranking methods give very similar good results for balanced accuracy.

Classification results for Australian credit approval dataset are presented in Figure 6 to Figure 9. The quality of classification for this dataset is obviously influenced by the choice of ranking indices. Classification accuracy with RF ranking method is quite high for RBF network and Naive Bayes, but quite low with IB1 classifier. All ranking methods give better results for balanced accuracy for Australian credit approval dataset, then for German credit dataset. In this case, simple ranking is sufficient to obtain good results.

6. CONCLUSIONS

The problem of ranking has recently gained much attention in machine learning. Ranking methods may filter features to reduce dimensionality of the feature space. This is especially effective for classification methods that do not have any inherent feature selections built in, such as the nearest neighbour methods or some types of neural networks. Different entropy-based and statistical indices have been used for feature ranking, evaluated and compared using four different types of classifiers on two real benchmark data. Accuracy of the classifiers is influenced by the choice of ranking indices.

There is no best ranking index, for different datasets and different classifiers accuracy curves as a function of the number of features used may significantly differ. Evaluation of ranking indices is fast. The only way to be sure that the highest accuracy is obtained in practical problems requires testing a given classifier on a number of feature subsets, obtained from different ranking indices. The number of tests needed to find the best feature subset is very small comparing to the cost of wrapper approach for larger number of features.

There are many questions and issues that remain to be addressed and that we intend to investigate in the future work. Several improvements of the ranking methods presented here are possible:

- The algorithms and datasets will be selected according to precise criteria: entropy-based algorithms and several datasets, either real or artificial, with nominal, binary and continuous features.
- Features with the lowest ranking values of various indices in all crossvalidations may be safely rejected.
- The remaining features should be analyzed using selection methods that allow elimination of redundant and correlated features.

These conclusions and recommendations will be tested on larger datasets using various classification algorithms in the near future. Future work will also focus on extending this work to more datasets, developing a more thorough analysis and building interpretable metamodels to extract those aspects of datasets responsible for the observed results.

REFERENCES

[1] Kohavi, R., and John, G.H., “Wrappers for feature subset selection”, *Artificial Intelligence*, 97 (1997) 273-324.

[2] Blum, A.L., and Rivest, R.L., “Training a 3-node neural networks is NP-complete”, *Neural Networks*, 5 (1992) 117–127.

[3] Wyse, N., Dubes, R., and Jain, A.K., “A critical evaluation of intrinsic dimensionality algorithms”, in: E.S. Gelsema and L.N. Kanal, (eds), *Pattern Recognition in Practice*, Morgan Kaufmann Publishers, Inc., 1980, 415–425.

[4] Ben-Bassat, M., “Pattern recognition and reduction of dimensionality”, in: P. R. Krishnaiah and L. N. Kanal, (eds), *Handbook of Statistics-II*, North Holland, 1982, 773–791.

[5] Siedlecki, W., and Sklansky, J. “On automatic feature selection”, *International Journal of Pattern Recognition and Artificial Intelligence*, 2 (1988) 197–220.

[6] Blum, A.I., and Langley, P., “Selection of relevant features and examples in machine learning”, *Artificial Intelligence*, 97 (1997) 245-271.

[7] Dash, M., and Liu, H., “Feature selection methods for classifications”, *Intelligent Data Analysis: An International Journal*, 1 (3) 1997. http://www-east.elsevier.com/ida/free.htm.

[8] Dy, J.G., and Brodley, C.E., “Feature subset selection and order identification for unsupervised learning”, in: *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, 247–254.

[9] Kim, Y., Street, W., and Menczer, F., “Feature selection for unsupervised learning via evolutionary search”, in: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, 365–369.

[10] Das, S., “Filters, wrappers and a boosting-based hybrid for feature selection”, in: *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[11] Mitra, P., Murthy, C. A., and Pal, S. K., “Unsupervised feature selection using feature similarity”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (3) (2002) 301–312.

[12] Quinlan, J. R., *C4.5: Programs for Machine Learning*, San Mateo, Morgan Kaufman, 1993.

[13] Doak, J., “An evaluation of feature selection methods and their application to computer security”, Technical report, Davis CA: University of California, Department of Computer Science, 1992.

[14] Talavera, L.,“Feature selection as a preprocessing step for hierarchical clustering”, in: *Proceedings of Internationl Conference on Machine Learning (ICML’99)*, 1999.

[15] Liu, H., and Motoda, H., *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998.

[16] Almuallim, H., and Dietterich, T. G., “Learning with many irrelevant features”, in: *Proc. AAAI-91*, Anaheim, CA, 1991, 547-552.

[17] Kira, K., and Rendell, L. A., “The feature selection problem: tradional methods and a new algorithm”, in: *Proc. AAAI-92*, San Jose, CA, 1992, 122-126.

[18] Breiman, L., Friedman, J.H., Olshen, R.H., and Stone, C.J., *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA, 1984.

[19] Duch, W., Adamczak, R., and Grabczewski, K., “A new methodology of extraction, optimization and application of crisp and fuzzy logical rules”, *IEEE Transactions on Neural Networks*, 12 (2001) 277-306.

[20] Fayyad, U.M., and Irani, K.B. “The attribute selection problem in decision tree generation”, in: *AAAI-92, Proceedings of the Ninth National Conference on Artificial Intelligence*, AAAI Press/The MIT Press, 1992, 104–110.

[21] Liu, H., and Setiono, R. “A probabilistic approach to feature selection - a filter solution”, in: L. Saitta, (ed.), *Proceedings of International Conference on Machine Learning (ICML-96), July 3-6, 1996*, Bari, Italy, 1996, San Francisco: Morgan Kaufmann Publishers, CA, 319–327.

[22] John, G.H., Kohavi, R., and Pfleger, K., “Irrelevant feature and the subset selection problem”, in: W.,W. and Hirsh H., Cohen, (eds.), *Machine Learning: Proceedings of the Eleventh International Conference*, New Brunswick, N.J., 1994, Rutgers University, 121–129.

[23] Caruana, R., and Freitag, D., “Greedy attribute selection”, in: *Proceedings of International Conference on Machine Learning (ICML-94)*, Menlo Park, California, 1994, AAAI Press/The MIT Press, 28–36.

[24] Xing, E., Jordan, M., and Karp, R., “Feature selection for high-dimensional genomic microarray data”, in: *Proceedings of the Eighteenth International Conference On Machine Learning*, 2001.

[25] Weiss, S.M., and Kulikowski, C.A., *Computer Systems That Learn*, Morgan Kaufmann Publishers, San Mateo, California, 1991.

[26] Dash, M., Liu, H., and Yao, J., “Dimensionality reduction of unsupervised data”, in: *Proceedings of the Ninth IEEE International Conference on Tools with AI (ICTAI’97),* November, 1997, Newport Beach, California, 1997, IEEE Computer Society, 532–539.

[27] Dash, M., and Liu, H., “Handling large unsupervised data via dimensionality reduction”, in: *Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop*, 1999.

[28] Yang, J., and Honavar, V., “Feature subset selection using a genetic algorithm”, *IEEE Intelligent Systems,* 13 (1998) 44-49.

[29] Koller, D., and Sahami, M., “Toward optimal feature selection”, in: *International Conference on Machine Learning*, 1996, 284-292.

[30] Abe, N., and Kudo, M., “Entropy criterion for classifier-independent feature selection”, *Lecture Notes in Computer Science*, 3684 (2005) 689-695.

[31] Hall, M.A., and Smith, L.A., “Practical feature subset selection for machine learning”, *Proceedings of the 21st Australian Computer Science Conference*, 1998, 181–191.

[32] Liu, H., and Setiono, R., “Chi2: Feature selection and discretization of numeric attributes”, *Proc. IEEE 7th International Conference on Tools with Artificial Intelligence*, 1995, 338-391.

[33] Holte, R.C., “Very simple classification rules perform well on most commonly used datasets”, *Machine Learning*, 11 (1993) 63-91.

[34] Marko, R.S., and Igor, K., “Theoretical and empirical analysis of relief and rreliefF”, *Machine Learning Journal*, 53 (2003) 23–69. doi: 10.1023/A:1025667309714

[35] Kuramochi, M., and Karypis, G., “Gene classification using expression profiles: a feasibility study”, *International Journal on Artificial Intelligence Tools*, 14 (4) (2005) 641-660.

[36] Domingos, P., and Pazzani, M., “Feature selection and transduction for prediction of molecular bioactivity for drug design”, *Machine Learning*, 29 (1997) 103-130.

[37] Xing, E. P., Jordan, M. L., and Karp, R. M., “Feature selection for high-dimensional genomic microarray data”, *Proceedings of the 18th International Conference on Machine Learning*, 2001, 601-608.

[38] Frank, A., and Asuncion, A., *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.