Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 6, Issue 07, July -2019

A survey on optimal features selection for large datasets using machine learning algorithms.

Mrs.E.Aarthi, Dr.P.Muthulakshmi

Department of Computer Science, SRMIST, Chennai, India

Abstract: *Feature selection is the method of reducing data dimension while doing predictive analysis. One major reason is that machine learning follows the rule of "garbage in-garbage out" and that is why one needs to be very concerned about the data that is being fed to the model. In this survey, we review work in machine learning on methods for handling data sets containing large amounts of irrelevant information. We focus on two key issues: the problem of selecting relevant features, and the problem of selecting relevant examples. We describe the advances that have been made on these topics in both empirical and theoretical work in machine learning, and we present a general framework that we use to compare different methods. We close with some challenges for future work in this area.*

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 NPhard [2]. Features for use in representing the data, and the problem of selecting the most relevant examples to drive the learning process. We review recent work on these topics, presenting general frameworks that we use to compare and contrast different approaches.We begin with the problem of focusing on relevant features. High number of features in the data increases the risk of **Overfitting** in the Model. Feature Selection method helps to reduce the dimension of features by without much loss of information.

How to select features and what are Benefits of performing feature selection before modeling your data?

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.

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 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 2N, 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 [8]. It is also possible to use heuristic search methods. There are two main families of these methods: *forward addition* [9] (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.

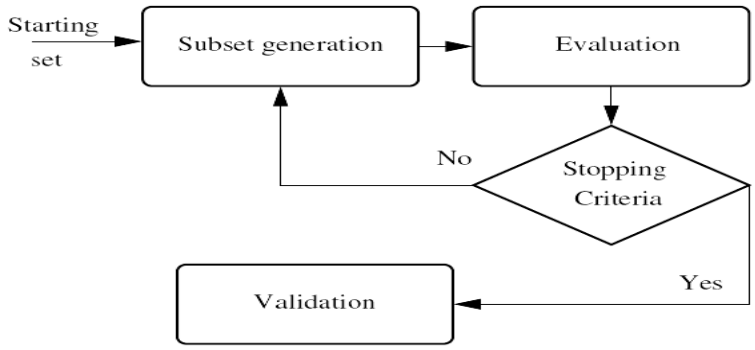


Figure - 1

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.

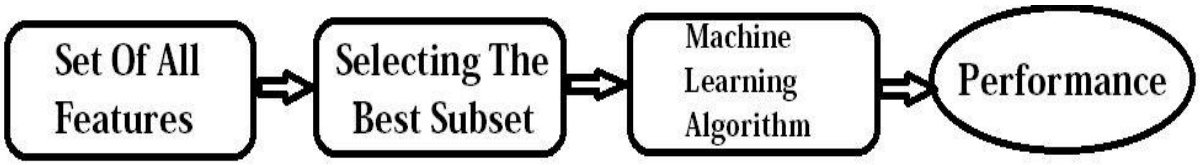
Validation

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 SELECTION TECHNIQUES

Filter Method

This method uses the variable ranking technique in order to select the variables for ordering and here,[7] the selection of features is independent of the classifiers used. By ranking, it means how much useful and important each feature is expected to be for classification. It basically selects the subsets of variables as a pre-processing step independently of the chosen predictor. In filtering, the ranking method can be applied before classification for filtering the less relevant features. It carries out the feature selection task as a pre-processing step which contains no induction algorithm.



Some examples of filter methods are mentioned below:

Chi-Square Test: In general term, this method is used to test the independence of two events. If a dataset is given for two events, we can get the observed count and the expected count and this test measures how much both the counts are derivate from each other.

Variance Threshold: This approach of feature selection removes all features whose variance does not meet some threshold. Generally, it removes all the zero-variance features which means all the features that have the same value in all samples.

Information Gain: Information gain or IG measures how much information a feature gives about the class. Thus, we can determine which attribute in a given set of training feature is the most meaningful for discriminating between the classes to be learned.

Wrapper Method

The Wrapper Methodology was made famous by researchers Ron Kohavi and George H. John in the year 1997. [10] This method utilises the learning machine of interest as a black box to score subsets of variables according to their predictive power. In the above figure, in a supervised machine learning, the induction algorithm is depicted with a set of training instances, where each instance is described by a vector of feature values and a class label. The induction algorithm which is also considered as the black box is used to induce a classifier which is useful in classifying. In the wrapper approach, the feature subset selection algorithm exists as a wrapper around the induction algorithm. One of the main drawbacks of this technique is the mass of computations required to obtain the feature subset.

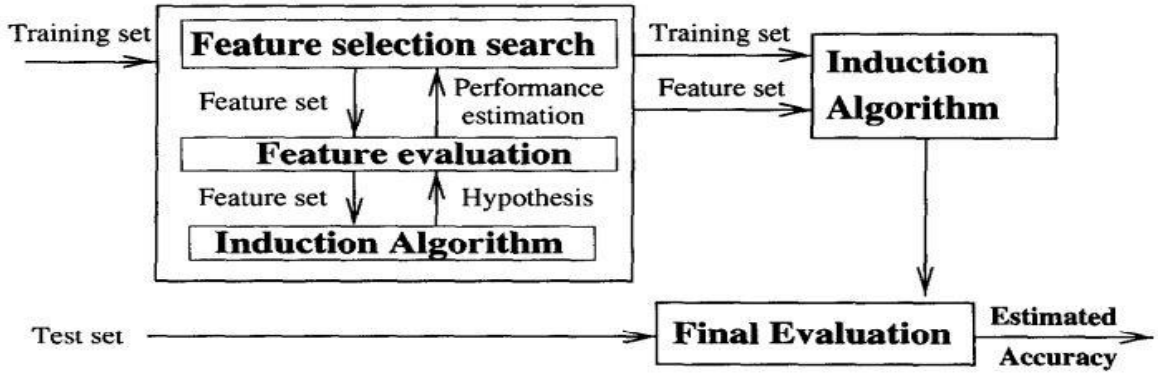


Fig: Wrapper Approach to feature subset selection

Some examples of Wrapper Methods are mentioned below:

Genetic Algorithms: This algorithm can be used to find a subset of features. CHCGA is the modified version of this algorithm which converges faster and renders a more effective search by maintaining the diversity and evade the stagnation of the population.

Recursive Feature Elimination: RFE is a feature selection method which fits a model and removes the weakest feature until the specified number of features is satisfied. Here, the features are ranked by the model's coefficient or feature importances attributes.

Sequential Feature Selection: This naive algorithm starts with a null set and then add one feature to the first step which depicts the highest value for the objective function and from the second step onwards the remaining features are added individually to the current subset and thus the new subset is evaluated. This process is repeated until the required number of features are added.

Embedded Method

[8]This method tries to combine the efficiency of both the previous methods and performs the selection of variables in the process of training and is usually specific to given learning machines. This method basically learns which feature provides the utmost to the accuracy of the model. Some examples of Embedded Methods are mentioned below:

L1 Regularisation Technique such as LASSO: Least Absolute Shrinkage and Selection Operator (LASSO) is a linear model which estimates sparse coefficients and is useful in some contexts due to its tendency to prefer solutions with fewer parameter values.

Ridge Regression (L2 Regularisation): The L2 Regularisation is also known as Ridge Regression or Tikhonov Regularisation which solves a regression model where the loss function is the linear least squares function and regularisation.

Elastic Net: This linear regression model is trained with L1 and L2 as regulariser which allows for learning a sparse model where few of the weights are non-zero like Lasso and on the other hand maintaining the regularisation properties of Ridge.

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.[16] 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. [12]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 [20].

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 [6]. 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:

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

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

where *X* = (*x1*, *x2*, ..., *xN*) denotes a feature vector and *cj* , 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*(*ci*). Here, *P*(*ci*) 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 *xj* within the training subset that is labelled as class *ci*. 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 [14]. 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. [17] 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 hyper ellipsoid. 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 thehidden 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 [19]. 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

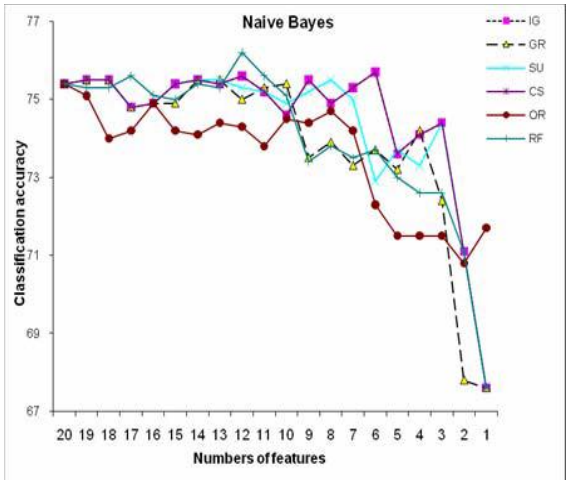


Figure -2 Ranking methods and balanced classification accuracy for German credit dataset, Naive Bayes classifier

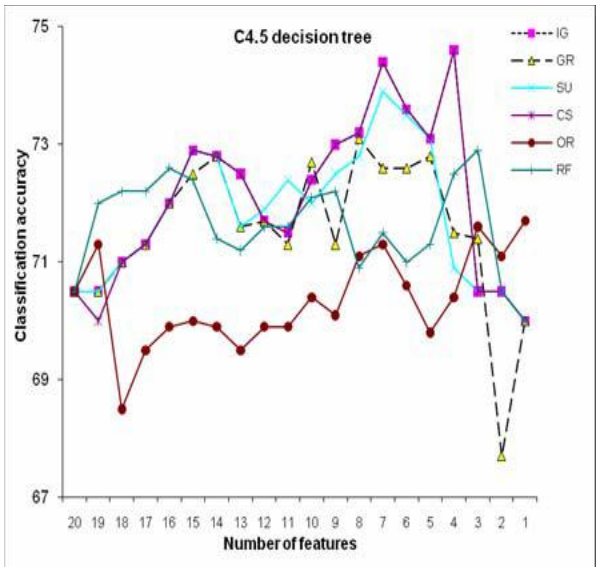


Figure 3: Ranking methods and balanced classification accuracy for German credit dataset, C4.5 decision tree classifier The datasets described above have been used in tests. Six ranking methods have been used in each case: CS, OR, RF, IG, GR and SU

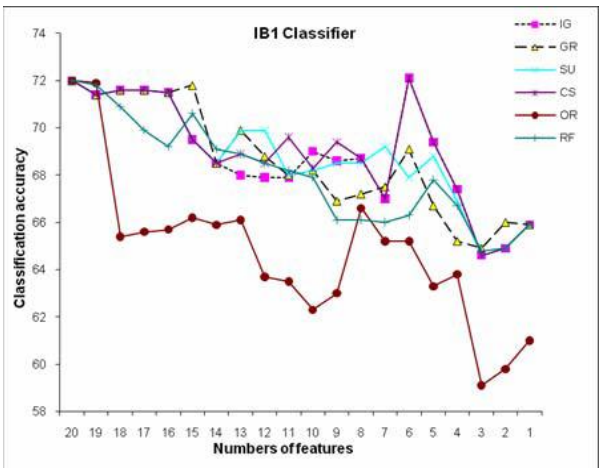


Figure 4: Ranking methods and balanced classification accuracy for German credit dataset, IB1 classifier

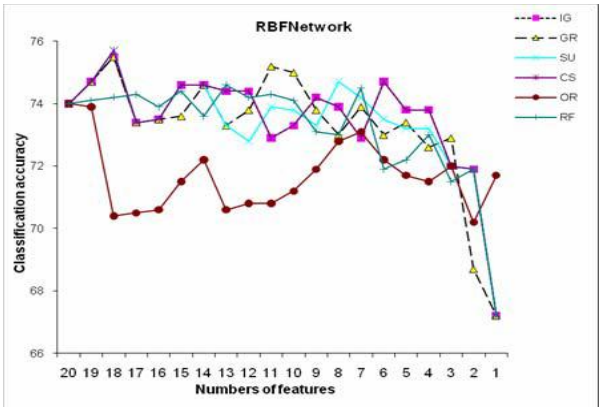


Figure 5: Ranking methods and balanced classification accuracy for German credit dataset, RBF network

Significant differences [21] 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. 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.

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.

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. J. Novakovic, P. 134 Strbac, D. Bulatovic / Toward Optimal Feature Selection

[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.

