Acta Universitatis Apulensis No. 26/2011
ISSN: 1582-5329 pp. 203-216

FEATURE SELECTION FOR CLASSIFIER ACCURACY IMPROVEMENT

Maria Muntean, Honoriu Vălean,
Remus Joldeş, Emilian Ceuca

Abstract. Most of the time a lot of data means better results. This case is not valid all the time because sometimes we have a lot of redundant data and a lot of attributes that are weakly related to what we are trying to find out by evaluating the data.

The main idea behind feature selection is to keep the data that bring the most amount of information for learning how to evaluate future data that are going to be fed to the system and to discard the features that do not bring any new information.

In this paper we investigated whether feature selection methods can improve the accuracy and the time spent for classification. We have used the Support Vector Machine, a powerful classification technique based on kernels, which has proven to be efficient for nonlinearly separable input data.

1.Introduction

Basically what data mining tell us is that the more features we have, the better it is to make more accurate predictions about future instances. Practically the amount of training data is limited, and excessive features are going to slow down the learning process. This is also one of the main causes why classifiers over-fit the training data, and perform poorly when faced with real life problems. Due to the large dimensionality, much time and memory are needed for training a classifier on a large collection of data.

The pre-processing stage of data mining cleans the data. Feature selection, an important pre-processing method, tries to find the minim subset of the training data set, such that, this subset to be equal or as close as possible to the data set that will be used by the classifier in the training step. As a machine learning, we used the Support Vector Machine (SVM), a promising new method for the classification of both linear and nonlinear data.

2. Support Vector Machine

SVM, proposed by Vapnik and his colleagues in 1990's [1], is a new machine learning method based on Statistical Learning Theory and it is widely used in the area of regressive, pattern recognition and probability density estimation due to its simple structure and excellent learning performance. Joachims validated its outstanding performance in the area of text categorization in 1998 [2]. SVM can also overcome the over fitting and under fitting problems [3], [4], and it has been used for imbalanced data classification [5], [6].

The SVM technique is based on two class classification. There are some methods used for classification in more than two classes. Looking at the two dimensional problem we actually want to find a line that " best" separates points in the positive class from the points in the negative class. The hyperplane is characterized by the decision function

$$f(x) = sgn(< w, \Phi(x) > +b),$$

where w is the weight vector, orthogonal to the hyperplane, b is a scalar that represents the margin of the hyperplane, x is the current sample tested, $\Phi(x)$ is a function that transforms the input data into a higher dimensional feature space and " $<$, $>$" representing the dot product. Sgn is the signum function. If w has unit length, then $< w, \Phi(x) >$ is the length of $\Phi(x)$ along the direction of w.

To construct the SVM classifier one has to minimize the norm of the weight vector w (where $||w||$ represents the Euclidian norm) under the constraint that the training patterns of each class reside on opposite sides of the separating surface. The training part of the algorithm needs to find the normal vector w that leads to the largest b of the hyperplane.

The algorithm can be generalized to non-linear classification by mapping the input data into a higher-dimensional feature space via an a priori chosen non-linear mapping function Φ and construct a separating hyperplane with the maximum margin. In solving the quadratic optimization problem of the linear SVM (i.e. when searching for a linear SVM in the new higher dimensional space), the training tuples appear only in the form of dot products, $< \Phi(x_i), \Phi(x_j) >$, where $\Phi(x)$ is simply the nonlinear mapping function applied to transform the training tuples. Expensive calculation of dot products $< \Phi(x_i), \Phi(x_j) >$ in a high-dimensional space can be avoided by introducing a kernel function K:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{1}$$

The kernel trick can be applied since all feature vectors only occur in dot products. The weight vectors than become an expression in the feature space, and

therefore Φ will be the function through which we represent the input vector in the new space. Thus it is obtained the decision function having the following form:

$$f(x) = sgn(\sum_{i \in \Re} y_i \alpha_i k(x, x_i) + b) \tag{2}$$

where α_i represent the Lagrange multipliers and the samples x_i for which $\alpha_i > 0$ are called Support Vectors [7].

The idea of the kernel is to compute the norm of the difference between two vectors in a higher dimensional space without representing those vectors in the new space. Regarding the final SVM formulation, the free parameters of SVMs to be determined within model selection are given by the regularization parameter c and the kernel, together with additional parameters of the respective kernel function.

The new SVM learning algorithm is called Sequential Minimal Optimization (SMO). Unlike the previous methods, SMO chooses to solve the smallest possible optimization problem at every step. For the standard SVM quadratic programming problem, the smallest possible optimization problem involves two Lagrange multipliers, because the Lagrange multipliers must obey a linear equality constraint. At every step, SMO chooses two Lagrange multipliers to jointly optimize, finds the optimal values for these multipliers, and updates the SVM to reflect the new optimal values.

We chose to perform the experiments with Polynomial Kernel function that is given by:

$$K(x_i, x_j) = (m \cdot x_i \cdot x_j + n)^{\exp} \tag{3}$$

The two parameters we focused on were:

- c – the complexity parameter of the SMO classifier;
- exp – the exponent of the Polynomial Kernel function.

In the evaluations performed the value of complexity and then the value of the exponent were changed.

3.Feature Selection Approach

The problem of feature selection involves finding a "good" set of attributes under some objective function that assigns some numeric measure of quality to the patterns discovered by the data mining algorithm [8].

The objective of a data mining algorithm A is to take a training set T and discover a set of patterns P such that P optimizes some objective function $F(P)$ that assigns some real-value measure of goodness to P. The output of A is determined by which attributes are present in the training set. We can parameterize the attributes used

as a Boolean vector b, where $b_i = 0$ means attribute i is not used and $b_i = 1$ indicates that it is used.

In general, finding the optimal subset is impossible for two reasons. First, most objective functions cannot be calculated precisely, and can only be approximated. Even if it would be exact, there is the practical problem that if there are m attributes, there are $2m$ possible values for b, a number of choices typically too large to search exhaustively [10].

Since we cannot always hope to find the optimal subset, we will try to find an approximating subset that will improve prediction accuracy.

An optimal feature subset need not be unique because it may be possible to achieve the same accuracy with different subsets of features (if two features are perfectly correlated, one can be replaced by the other).

There are some important approaches for feature selection: filter approach, embedded approach and wrapper approach [9].

Filter Approach

The filter approach selects features using a preprocessing step.

Figure 1: The Filter approach

The filter approach attempts to asses the merits of features from data alone, without taking into consideration any particular inducer. Since filter methods are oblivious to the choice of the predictor, they must be derived from the properties of the data distribution. Hence, filter methods cannot estimate the optimal subset, as it depends on a particular inducer.

Wrapper Approach

In the wrapper method, the attribute subset selection algorithm exists as a wrapper around the data mining algorithm and result evaluation. The induction algorithm is used as a black box. The feature selection algorithm conducts a search for a

good subset using the induction algorithm itself as a part of the evaluation function.

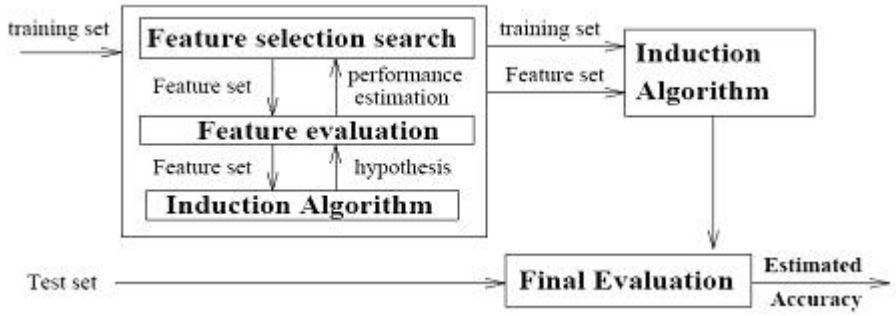

Figure 2: The Wrapper approach

The search approaches most commonly used include: best first search, simulated annealing, genetic algorithms, greedy stepwise forward selection, and greedy stepwise backward elimination.

Embedded Approach

This approach is a form of regularization. Feature selection regularizes predictor estimation by constraining the dimension of the input space.

Some predictors are parameterized in a way that reveals how each feature influences the prediction. The most obvious cases are linear classifiers and linear regression classifiers.

This is referred to as embedded feature selection, since feature selection happens "inside" the inducer and cannot be separated from it.

Embedded methods incorporating feature selection include: decision trees, random forest (which consists of many decision trees), ridge regression, random multinomial logit (RNML), and other machine learning methods that include a pruning step. That pruning is performed on the basis of an internal and implicit feature selection [10].

In this paper, we proposed to implement the wrapper method for selecting features based on a linear SVM and to evaluate the accuracy and the time spent for the

evaluation. Mladenic et al. [11] compared more traditional feature selection methods, such as Odds Ratio and Information Gain, in achieving the desired tradeoff between the vector sparseness and the classification performance. The results indicate that at the same level of sparseness, feature selection based on normal SVM yields better classification performances. In [12] the advantages of using the same methods in the features selection step and in the learning step are explained.

4. The Wrapper method proposed

Since the Wrapper Subset Evaluation produces the best results out of the feature selection methods [14], [15], we performed the experiments with the Weka implementation of this method. As any feature selection, the Weka subset evaluation is going to try to evaluate the attributes, arrange them in the order of their importance, and tell the user, which are the non important attributes and the weakly relevant ones but with redundant features. After this operation the result would be a dataset containing only strongly relevant features, and weakly relevant features, but with non redundant information.

In the experimental part we took two datasets and evaluated the attributes there. In the end, the result was a dataset with no redundancy in the information and with better results in the evaluation of the dataset (time wise, more efficiency and possibly comparable or better accuracy). We used the following procedure for this evaluation:

1. Evaluate the attributes of the dataset using the Weka wrapper subset evaluation;
2. Rank the most important features according to the score received after the Wrapper Subset Evaluation;
3. Divide the attributes into Strong attributes and weak attributes:

   - The attributes with a score above the average were considered strong attributes;

   - The attributes with a score below this average were considered weak;
4. Evaluate each attribute as class in order to see how it is evaluated by the other attributes;
5. Delete the weak attributes;
6. Evaluate the dataset using 10 folds cross validation, and having only the strong attributes remaining.

In cross-validation, data was split into more partitions. Each partition of the data was used in turn as training and testing data. When one partition was used for testing, the remaining partitions were used for training. This is called stratified cross-validation, or n-fold cross-validation, where n is the number of partitions, or folds.

Previous extensive tests on numerous datasets performed by numerous researchers yielded 10-fold cross-validation as the de-facto standard [13]. Neither the stratification, nor the division into folds has to be exact, but there should be made an attempt for the sets to be approximately equal and, most important, the different class values should be represented in the right proportion in all folds.

5. Experimental results

In order to evaluate the Feature Selection using SMO, we applied it on two datasets with many attributes: tic-tac-toe.arff and vote.arff.

Tic-Tac-Toe dataset:

This dataset describes the moves of the same game and contains the optimal responses to these moves. It contains 9 attributes, one for each square of the game and almost 1000 instances.

For simplicity, and for an easier representation, the attributes were numbered from 1 to 9, and each number corresponded to the square indicated in Table 1.

1	2	3
4	5	6
7	8	9

Table 1. Attributes in Tic-Tac-Toe

The table needs to be interpreted like this: 1 is top-left; 2 is top-middle; 5 is middle-middle, 9 is bottom-right and so on.

c	exp	1	2	3	4	5	6	7	8	9
1	1	90%*	50%	50%	50%	100%*	50%	50%	50%	50%
1.1	1	80%*	30%	30%	30%	100%*	30%	30%	30%	30%
1.3	1	80%*	60%*	50%	50%	100%*	60%*	50%	50%	50%
1.5	1	80%*	50%*	40%	40%	100%*	40%	40%	50%*	40%
1.7	1	80%*	50%	50%	50%	100%*	50%	50%	50%	50%
1.9	1	50%*	50%*	40%	50%*	100%*	50%*	40%	60%*	40%
1	1.3	20%*	0%	20%*	20%*	100%*	10%	30%*	20%*	40%*
1.1	1.3	20%*	0%	20%*	20%*	100%*	10%	30%*	20%*	40%*
1.3	1.3	50%*	0%	20%	20%	100%*	10%	50%*	10%	20%
1.5	1.3	30%*	0%	40%*	20%	100%*	10%	30%*	10%	30%*
1.7	1.3	40%*	0%	30%*	20%	100%*	10%	40%*	10%	30%*
1.9	1.3	50%*	0%	50%*	10%	100%*	10%	30%	0%	30%

*= Accepted.

Table 2. The Feature selection attribute evaluation

The process of selecting the best attributes from this dataset proved to be very time consuming, each evaluation taking at least 3.5 hours. After applying the Wrapper subset Evaluation on the dataset, we observed the importance in percentages of each field. Surprisingly, almost at every time all the attributes were of at least little importance. In this case at each evaluation the attributes that had an evaluation percentage above the average were marked as green, and kept for the next evaluation. All of the other attributes were deleted and therefore not important any more for the next evaluation.

The next step in this process was to set all attributes in turn as class and evaluate the dataset in order to see how the algorithm distinguishes between the different features of the dataset. The result of this stage is presented in Table 3 (in %).

c	exp	1	2	3	4	5	6	7	8	9	Class
1	1	99.06	99.06	99.06	99.06	99	99.06	99.06	99.06	99.06	98.32
1.1	1	99.06	99.06	99.06	99.06	99.16	99.06	99.06	99.06	99.06	98.39
1.3	1	99.06	99.06	99.06	99.06	99.16	99.06	99.06	99.06	99.06	98.32
1.5	1	99.06	99.06	99.06	99.06	99.16	99.06	99.06	99.06	99.06	98.33
1.7	1	99.06	99.06	99.06	99.06	99.16	99.06	99.06	99.06	99.06	98.32
1.9	1	99.06	99.06	99.06	99.06	99.16	99.06	99.06	99.06	99.06	98.32
1	1.3	96.45	96.34	96.24	95.72	96.86	96.03	95.82	96.65	97.07	94.88
1.1	1.3	95.61	95.09	93.94	94.57	96.65	97.07	95.51	95.82	95.82	93.73
1.3	1.3	95.51	94.78	94.46	95.40	95.72	95.19	95.19	95.40	95.51	93.31
1.5	1.3	94.25	95.61	94.46	95.19	94.98	94.36	95.09	94.78	95.09	93.31
1.7	1.3	94.25	96.61	94.46	95.19	94.98	94.30	95.09	94.78	94.88	93.00
1.9	1.3	93.73	94.36	94.88	95.19	95.51	95.19	94.57	94.25	95.09	92.17

Table 3. Evaluation of the each attribute as class (in %)

We concluded that the Polynomial Kernel evaluated all the attributes with a high accuracy. From this result we deduced that the attributes were strongly related to one another and this was confirmed also by the fact that the feature selection algorithm gave some importance to almost all of the attributes, meaning that did not have much redundant information stored in them.

After this, the intuition existed that because of the low decoupling and of the attributes, and because in the dataset was not that much redundant data, the result of the feature selection was going to be a dataset that was going to take less to evaluate, but which was going to have a poorer accuracy level, after taking out the non-redundant information.

c	exp	1	2	3	4	5	6	7	8	9	class
1	1	46.03	DEL	DEL	DEL	58.24	DEL	DEL	DEL	DEL	69.93
1.1	1	46.03	DEL	DEL	DEL	58.24	DEL	DEL	DEL	DEL	69.93
1.3	1	50.73	42.90	DEL	DEL	58.24	48.22	DEL	DEL	DEL	69.93
1.5	1	50.73	39.97	DEL	DEL	47.59	DEL	DEL	47.59	DEL	69.93
1.7	1	50.73	DEL	DEL	DEL	58.24	DEL	DEL	DEL	DEL	69.93
1.9	1	55.53	48.53	DEL	46.86	58.24	DEL	52.81	53.65	DEL	69.93
1	1.3	59.18	DEL	63.88	58.24	65.76	DEL	52.19	57.09	58.97	75.57
1.1	1.3	56.36	DEL	63.56	56.15	63.15	DEL	49.37	56.36	56.78	63.25
1.3	1.3	49.47	DEL	DEL	DEL	58.97	DEL	49.16	DEL	DEL	70.25
1.5	1.3	44.67	DEL	41.96	DEL	56.36	DEL	42.38	DEL	43.21	65.30
1.7	1.3	44.15	DEL	42.38	DEL	55.63	DEL	43.52	DEL	41.44	64.92
1.9	1.3	50.62	DEL	49.06	DEL	58.45	DEL	DEL	DEL	DEL	70.45

Table 4. The effect of Feature selection (in %)

In the above table, from 1 to 9 were numbered the attributes of this dataset. The attributes pictured in red and labeled DEL, are the ones that were deleted after the Wrapper subset evaluation ranked them as below the average. Because of the lack of redundancy in this dataset, it could be seen that the accuracy of the evaluation dropped from 98% to 65% . This drop in the accuracy was very significant and unaffordable to have.

From the Tic-Tac-Toe dataset there could be made the conclusion: if the feature selection would rank more attributes as important, and find non-redundant data inside some of the attributes, there should not be applied the rule with the deletion of the attributes that are selected when they have an above the average importance.

Vote dataset:

The Vote dataset was taken from the Title: 1984 United States Congressional Voting Records Database. This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA.

In order to simplify the recognition of all the attributes, they were renamed, by being numbered from 1 to 16:

Handicapped-infants	1	mx-missile	9
Ware-project-cost-sharing	2	Immigration	10
Adoption-of-the-budget-resolution	3	Synfuels-corporation-cutback	11
Physician-fee-freeze	4	Education-spending	12
El-Salvador-aid	5	Superfund-right-to-sue	13
Religious-groups-in-schools	6	Crime	14
Anti-satellite-test-ban	7	Duty-free-exports	15
Aid-to-Nicaraguan-contras	8	Export-administration-act-S-Africa	16

Table 5. Attributes in Vote dataset

The first evaluation that was performed on the dataset was the Wrapper Subset Evaluation. After this evaluation we observed that the only attribute that was classified as important was the physician fee freeze, i.e., **attribute number 4**. All the other attributes were evaluated as not important by the algorithm, having between 0 and 10 importance, thus having a lot of redundant information stored in.

The complete details of the evaluation are presented in Table 6, where it is also shown the accuracy of the evaluation with all attributes.

c	exp	1	2	3	4	5	6	7	8	9
1	1	67,61	64,85	87,02	96,22	94,76	85,84	85,98	90,23	88,86
1.1	1	67,13	65,37	87,97	95,75	94,28	85,84	85,74	90,23	88,61
1.3	1	67,13	65,67	87,97	95,28	94,52	85,37	85,27	90,47	88,61
1.5	1	67,13	65,37	87,97	95,04	94,04	85,61	85,74	90,47	88,61
1.7	1	67,13	64,85	88,20	95,04	94,28	85,37	85,27	90,47	88,61
1.9	1	67,13	87,97	95,04	95,04	94,04	85,37	85,27	90,71	88,61
1	1.3	61,70	57,36	87,26	94,81	95,23	82,54	85,51	90,71	89,10
1.1	1.3	62,41	57,10	87,26	94,81	95,23	81,60	85,98	90,47	89,34
1.3	1.3	63,82	56,84	87,73	94,81	95,23	80,42	85,03	91,19	88,86
1.5	1.3	61,22	57,62	86,55	94,81	95,71	80,43	85,74	91,90	89,10
1.7	1.3	61,70	56,84	86,08	94,81	95,47	79,71	85,74	92,14	89,10
1.9	1.3	60,52	56,84	86,79	94,81	95,23	81,13	84,79	92,38	88,61

c	exp	10	11	12	13	14	16	17	18
1	1	55,84	69,80	85,00	82,92	86,36	76,90	78,24	96,02
1.1	1	55,84	69,80	85,14	82,92	86,12	76,90	77,34	96,09
1.3	1	55,60	70,53	85,14	82,92	86,36	76,90	77,34	95,86
1.5	1	55,37	70,28	85,14	82,92	86,12	76,90	77,94	96,09
1.7	1	55,60	69,80	85,14	83,17	86,12	76,90	77,03	96,32
1.9	1	55,14	70,53	85,14	82,92	85,64	76,90	76,73	96,32
1	1.3	53,50	69,80	86,63	81,95	88,27	76,16	76,43	95,63
1.1	1.3	52,10	69,80	86,13	82,19	88,03	76,41	76,43	95,63
1.3	1.3	50,46	67,87	86,13	82,19	87,79	77,39	76,43	95,86
1.5	1.3	50,23	67,39	84,90	82,68	87,32	76,65	75,83	95,63
1.7	1.3	50,23	66,66	84,90	82,19	88,03	77,39	75,52	95,40
1.9	1.3	50,00	66,18	84,15	81,95	88,27	77,88	76,73	95,40

Table 6. Evaluation of attributes in Vote dataset (in %)

After the feature selection, the accuracy of the evaluation results were the same as with all of the attributes and in some cases even better.

c	exp	physic-fee-freeze	class
1	1	96.22%	95.63%
1.1	1	96.22%	95.63%
1.3	1	96.22%	95.63%
1.5	1	96.22%	95.63%
1.7	1	96.22%	95.63%
1.9	1	96.22%	95.63%
1	1.3	96.22%	95.63%
1.1	1.3	96.22%	95.63%
1.3	1.3	96.22%	95.63%
1.5	1.3	96.22%	95.63%
1.7	1.3	96.22%	95.63%
1.9	1.3	96.22%	95.63%
X	X	96.22%	95.63%

Table 7. Evaluation after deleting the redundant attributes

From this evaluation we observed that even with only one attribute remaining, and all the other containing redundant information deleted, the accuracy of the evaluation was very good. The evaluation time was almost 0 (0.02) from 0.31, thus more than 15 times less than the initial evaluation.

6. Conclusions

In this paper we investigated whether feature selection methods can improve the accuracy and the time of classification. The Wrapper Subset Evaluation algorithm was tested and two types of input data representations were used. Experiments were developed using a powerful classification technique based on kernels, named Support Vector Machine.

Two conclusions could be drawn after performing the experiments:

· If we have non redundant attributes, by deleting them the accuracy is going to have an important drop.

· If the deleted attributes are only the non important ones, and the ones with little importance and with redundant information, the accuracy is going to stay the same, and take less time, since there is going to be less data to process.

Feature selection is a really important feature of data mining, helping the system perform better and in less time, by removing the non important features from the dataset.

References

[1] V N. Vapnik, *The nature of statistical learning theory*, New York: Springer-Verlag, 2000.

[2] I. Joachims, *Text categorization with Support Vector Machines: Learning with many relevant features*, Proceedings of the European Conference on Machine Learning, Berlin: Springer, 1998.

[3] M. Hong, G. Yanchun, W. Yujie, and L. Xiaoying, *Study on classification method based on Support Vector Machine*, 2009 First International Workshop on Education Technology and Computer Science, Wuhan, China, pp.369-373, March, 7-8, 2009.

[4] X. Duan, G. Shan, and Q. Zhang, *Design of a two layers Support Vector Machine for classification*, 2009 Second International Conference on Information and Computing Science, Manchester, UK, pp. 247-250, May, 21-22, 2009.

[5] Y. Li, X. Danrui, and D. Zhe, *A new method of Support Vector Machine for class imbalance problem*, 2009 International Joint Conference on Computational Sciences and Optimization, Hainan Island, China, pp. 904-907, April 24-26, 2009.

[6] Z. Xinfeng, X. Xiaozhao, C. Yiheng, and L. Yaowei, *A weighted hyper-sphere SVM*, 2009 Fifth International Conference on Natural Computation, Tianjin, China, pp. 574-577, 14-16, August, 2009.

[7] D. Morariu, L. Vintan, V. Tresp, *Feature Selection Methods for an Improved SVM Classifier*, Proceedings of 14th International Conference on of Intelligent Systems (ICIS06), ISSN 1305-5313 Volume 14,(pp. 83-89), Prague, 2006.

[8] I. H. Witten, E. Frank, Data Mining, *Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, Elsevier Inc., 2005.

[9] R. Kohavi, *Wrappers for Performance Enhancement and Oblivious Decision Graphs*, Stanford University, 1995.

[10] T. Mitchell, *Machine Learning*, The McGraw-Hill Companies, Inc., 1997, pp. 52-78.

[11] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling, *Feature Selection Using Support Vector Machines*, The 27th Annual International ACM SIGIR Conference (SIGIR 2004), pp.234-241, 2004.

[12] T. Jebara, *Multi Task Feature and Kernel Selection for SVMs*, Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.

[13] S. Arlot, A. Celisse, *A survey of cross-validation procedures for model selection*, Statistics Surveys, Vol. 4 (2010) 40–79, ISSN: 1935-7516, p.52.

[14] Kohavi, R., *Wrappers for Performance Enhancement and Oblivious Decision Graphs*, PhD Thesis, 1995, Standford University.

[15] Zhiwei, X., Xinghua, W., *Research for Information Extraction Based on*

Wrapper Model Algorithm, 2010 Second International Conference on Computer Research and Development, May 2010, Kuala Lumpur, Malaysia, pp. 652-655.

Maria Muntean
Computer Science Department
"1 Decembrie 1918" University of Alba Iulia
E-mail: *mmuntean@uab.ro*

Honoriu Vălean
Automation Department
Technical University of Cluj Napoca
E-mail: *Honoriu.Valean@aut.utcluj.ro*

Remus Joldeş
Computer Science Department
"1 Decembrie 1918" University of Alba Iulia,
E-mail: *rjoldes@uab.ro*

Emilian Ceuca
Computer Science Department
"1 Decembrie 1918" University of Alba Iulia
E-mail: *eceuca@uab.ro*