# Feature Selection Methods for an Improved SVM Classifier

Daniel Morariu, Lucian N. Vintan, and Volker Tresp

***Abstract***—Text categorization is the problem of classifying text documents into a set of predefined classes. After a preprocessing step, the documents are typically represented as large sparse vectors. When training classifiers on large collections of documents, both the time and memory restrictions can be quite prohibitive. This justifies the application of feature selection methods to reduce the dimensionality of the document-representation vector. In this paper, three feature selection methods are evaluated: Random Selection, Information Gain (IG) and Support Vector Machine feature selection (called SVM_FS). We show that the best results were obtained with SVM_FS method for a relatively small dimension of the feature vector. Also we present a novel method to better correlate SVM kernel's parameters (Polynomial or Gaussian kernel).

***Keywords*—**Feature Selection, Learning with Kernels, Support Vector Machine, and Classification.

## I. Introduction

WHILE more and more textual information is available online, effective retrieval is difficult without good indexing and summarization of document content. Document categorization is one solution to this problem. In recent years a growing number of categorization methods and machine learning techniques have been developed and applied in different contexts.

Documents are typically represented as vectors in a features space. Each word in the vocabulary is represented as a separate dimension. The number of occurrences of a word in a document represents the value of the corresponding component in the document's vector. This document representation results in a huge dimensionality of the feature space, which poses a major problem to text categorization. The native feature space consists of the unique terms that occur into the documents, which can be tens or hundreds of thousands of terms for even a moderate-sized text collection. Due to the large dimensionality, much time and memory are needed for training a classifier on a large collection of documents. For this reason we explore various methods to reduce the feature space and thus the response time. As we'll show the categorization results are better when we work with a smaller optimized dimension of the feature space. As the feature space grows, the accuracy of the classifier doesn't grow significantly; actually it even can decreases due to noisy vector elements.

This paper represents a comparative study of feature selection methods used prior to documents classifications (Random Selection, Information Gain [4] and SVM feature selection [5]). Also we studied the influence of the input data representation on classification accuracy. We have used three type of representation, Binary, Nominal and Cornell Smart. For the classification process we used the Support Vector Machine technique, which has proven to be efficient for nonlinearly separable input data [8], [9], [11].

The Support Vector Machine (SVM) is actually based on learning with kernels. A great advantage of this technique is that it can use large input data and feature sets. Thus, it is easy to test the influence of the number of features on classification accuracy. We implemented SVM classification for two types of kernels: "*polynomial kernel*" and "*Gaussian kernel" (Radial Basis Function - RBF)*. We will use a simplified form of the kernels by correlating the parameters. We have also modified this SVM representation so that it can be used as a method of features selection in the text-mining step.

Section 2 and 3 contain prerequisites for the work that we present in this paper. In section 4 we present the framework and the methodology used for our experiments. Section 5 presents the main results of our experiments. The last section debates and concludes on the most important obtained results and proposes some further work.

## II. Feature Selection Methods

A substantial fraction of the available information is stored in text or document databases which consist of a large collection of documents from various sources such as news articles, research papers, books, web pages, etc. Data stored in text format is considered semi-structured data that means neither completely unstructured nor completely structured. In text categorization, feature selection is typically performed by assigning a score or a weight to each term and keeping some number of terms with the highest scores while discarding the rest. After this, experiments evaluate the effects that feature selection has on both the classification performance and the response time.

Numerous feature scoring measures have been proposed

Manuscript received June 22, 2006.

D. Morariu is with the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Computer Science Department, E. Cioran Street, No. 4, 550025 Sibiu, Romania, (phone: 40/0740/092202; e-mail: daniel.morariu@ulbsibiu.ro).

L. Vintan is with the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Computer Science Department, E. Cioran Street, No. 4, 550025 Sibiu, Romania, (e-mail: lucian.vintan@ulbsibiu.ro).

V. Tresp is with the Siemens AG, Information and Comunications, 81739 Munchen, Germany (e-mail: volker.tresp@siemens.com).

and evaluated: Odds Ratio [4], Information Gain, Mutual Information [4], Document Frequency, Term Strength [1], or Support Vector Machine [5], a. o.

As follows we'll present our three methods of features selection that we will further use in our work. All feature selection method use as a starting point the same vectors obtained after extraction step.

### *A. Random Selection*

In the Random feature selection method random weights between 0 and 1 are assigned to each feature. Then training and testing sets of various sizes are chosen by selecting the features according to their descending weights. These sets (with various sizes) are generated so that the larger sets are containing the smaller sets. We repeat this process for three times. After doing this we classify all of the sets and then we compute the average classification accuracy. This value will be considered the classification accuracy for random selection.

### *B. Information Gain*

Information Gain and Entropy [4] are functions of the probability distribution that underlie the process of communications. The entropy is a measure of uncertainty of a random variable. Based on entropy, as attribute effectiveness, a measure is defined in features selection, called "Information Gain", and is the expected reduction in Entropy caused by partitioning the samples according to this attribute. The Information Gain of an attribute relative to a collection of samples S, is defined as:

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{1}$$

where *Values(A)* is the set of all possible values for attribute A, and $S_v$ is the subset of S for which attribute A has the value v.

Forman in [2] reported that Information Gain failed to produce good results on an industrial text classification problem, as Reuter's database. He attributed this to the property of many feature scoring methods to ignore or to remove features needed to discriminate difficult classes.

### *C. SVM Feature Selection*

Mladenic et Al. [5], present a method for selecting features based on a linear Support Vector Machine. The authors compare 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 [3] the advantages of using the same methods in the features selection step and in the learning step are explained.

Following this idea we have used the SVM algorithm, with linear kernel, for feature selection. Thus the feature selection step becomes a learning step that trains using all features and calculates the (optimal) hyperplane that splits best the positive and negative samples. We obtain for each topic from the initial set the specified weight vector (the weight vector have the input space dimension) using linear kernels (multi-class classification). In contrast with Mladenic et. all, we normalized all weight vectors, obtained for each topics. We make an average over all weight vectors and obtain the weight vector used in the subsequent step. Using this weight vector we select only the features with a weight having an absolute value greater then a specified threshold.

## III. Support Vector Machine

The Support Vector Machine (SVM) is a classification technique based on statistical learning theory [8], [11] that was applied with great success in many challenging non-linear classification problems and was successfully applied to large data sets.

The SVM algorithm finds a hyperplane that optimally splits the training set. The optimal hyperplane can be distinguished by the maximum margin of separation between all training points and the hyperplane. Looking at a two-dimensional problem we actually want to find a line that "best" separates points in the positive class from points in the negative class. The hyperplane is characterized by a decision function like:

$$f(x) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{\Phi}(x) \rangle + b\right) \tag{2}$$

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, "*Φ(x)*" is a function that transforms the input data into a higher dimensional feature space and $\langle \cdot,\cdot \rangle$ representing the dot product. *Sgn* is the signum function. If **w** has unit length, then <**w,** Φ*(x)*> is the length of Φ*(x)* along the direction of **w**. Generally **w** will be scaled by ||**w**||. The training part the algorithm needs to find the normal vector "**w**" that leads to the largest "*b*" of the hyperplane.

## IV. Experimental Framework

### *A. The Dataset*

Our experiments are performed on the Reuters-2000 collection [10], which has 984Mb of newspapers articles in a compressed format. Collection includes a total of 806,791 documents, with news stories published by Reuters Press covering the period from 20.07.1996 through 19.07.1997. The articles have 9822391 paragraphs and contain 11522874 sentences and 310033 distinct root words. Documents are pre-classified according to 3 categories: by the *Region* (366 regions) the article refers to, by *Industry Codes* (870 industry codes) and by *Topics* proposed by Reuters (126 topics, 23 of them contain no articles). Due to the huge dimensionality of the database we will present here results obtained using a subset of data. From all documents we selected the documents for which the industry code value is equal to "System software". We obtained 7083 files that are represented using 19038 features and 68 topics. We represent documents as vectors of words, applying a stop-word filter (from a standard set of 510 stop-words) and extracting the word stem. From these 68 topics we have eliminated those topics that are poorly (less than 1% documents from all 7083 documents in the entire set) or excessively (more than 99% samples from the entire set) represented. After doing so we obtained 24 different topics and 7053 documents that were split randomly in training set (4702 samples) and evaluation set (2531 samples). In the feature extraction part we take into consideration both the article and the title of the article.

### *B. Kernel Types*

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. In practice we can see that by adding a constant bias to the kernel involves better classifying results. In this work we present results using a new idea to correlate this bias with the dimension of the space where the data will be represented. More information about this idea can be found in our previous work [7]. We consider that those two parameters (the degree and the bias) need to be correlated in order to improve the classification accuracy.

We'll present the results for different kernels and for different parameters for each kernel. For the polynomial kernel we vary the degree and for the Gaussian kernel we change the parameter C according to the following formulas (*x* and *x'* being the input vectors):

- *Polynomial*

$$k(x,x') = \left(2\cdot d + \langle x\cdot x'\rangle\right)^d \qquad (3)$$

*d* being the only parameter to be modified

- *Gaussian (radial basis function RBF)*

$$k(x,x') = \exp\left(-\|x-x'\|^2 / n\cdot C\right) \qquad (4)$$

*C* being the classical parameter and *n* being the new parameter, introduced by us, representing the number of elements from the input vectors that are greater than 0.

As linear kernel we used the polynomial kernel with degree 1. For feature selection with SVM method we used only the linear kernel.

### *C. Correlating Parameters for the Kernel*

Usually when learning with a polynomial kernel researchers use a kernel that can be expressed as like $\left(\langle \mathbf{x}\cdot\mathbf{x}'\rangle + b\right)^d$ where *d* and *b* are independent parameters. Parameter "*d"* is the kernel degree and it is used as a parameter that helps mapping the input data into a higher dimensional space. Thus, this parameter is intuitive. The second parameter "*b"* (the bias), is not so easy to infer. In all studied articles, the researchers used a nonzero *b*, but they didn't present a method for selection it. We notice that if this parameter was eliminated (i.e., chosen to be zero) the quality of the results can be poor. It is logically that there is a need to correlate the parameters *d* and *b* because the offset *b* needs to be modified as the dimension of the space modifies. Due to this, based on running laborious classification simulations presented in [7], we suggest the best correlation is "*b=2*d"*.

Also for the Gaussian kernel we modified the standard kernel used in the research community given by formula $k(x,x') = \exp(-\|x-x'\|^2 / C)$, where the parameter C is a number witch usually takes values between 1 and total numbers of features. We introduce the parameter *n* that multiplies the usually parameter *C* with a value that represents the number of distinct features having weights greater than 0 that occur in the current two input vectors, decreasing substantially the value of *C* (4). As far as we know, we are the first authors proposing a correlation between these two parameters for both polynomial and Gaussian kernels.

### *D. Representing the Data*

Because there are many ways to define the feature-weight, we represent the input data in three different formats [1], and we try to analyze their influence on the classification accuracy. In the following formulas *n(d, t)* is the number of times that term *t* occurs in document *d,* and *n(d,τ)* is the maximum frequency occurring in document *d*.

- *Binary representation* – in the input vector we store "0" if the word doesn't occur in the document and "1" if it occurs.
- *Nominal representation* – we compute the value of the weight using the formula:

$$TF(d,t) = \frac{n(d,t)}{\max_\tau n(d,\tau)} \qquad (5)$$

- *Cornell SMART representation –we compute the value of the weight using the formula:*

$$TF(d,t) = \begin{cases} 0 & \text{if } n(d,t)=0 \\ 1+\log(1+\log(n(d,t)) & \text{otherwise} \end{cases} \qquad (6)$$

## V. Experimental Results

### *A. Feature Selection for Multi-Class Classification*

For a fair comparison between the three feature selections methods used, we need to use the same number of features. For the Information Gain method the threshold for selecting the features represents a value between 0 and 1. For the other two methods the threshold represents the number of features that we want to obtain. This number must be equal with the number of features obtained through Information Gain method. If after feature selection step we obtain samples (vectors that characterize a document) that have all selected features equal to zero (those samples haven't features into the new feature set) we eliminated those samples from the sets.

In what follows we present the influence of the number of features regarding to the classification accuracy for each input data representation and for each feature selection method, considering 24 distinct classes. We present results only for a numbers of features smaller or equal to 8000. In [6] we showed that for a number of features greater than 8000, the classification accuracy doesn't increase, sometimes even decreases. Also, in [6] we present results for different value of

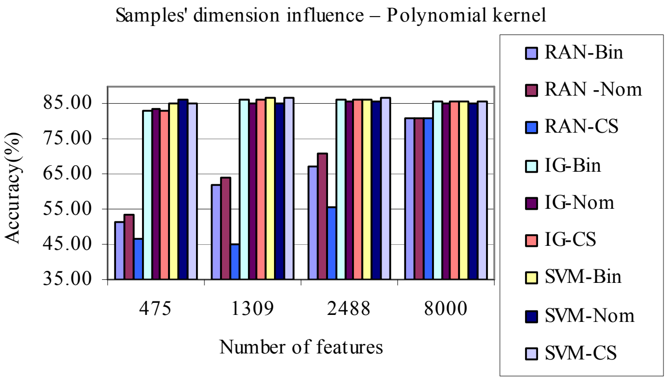


Fig. 1 Influence of the number of features on the classification accuracy using Polynomial kernel with degree equal to 2 (BIN – means Binary representation, Nom – nominal representation and CS – Cornel Smart representation)

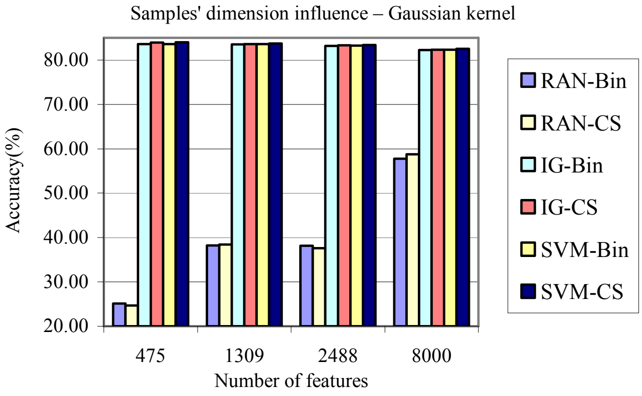


Fig. 2 Influence of the number of features on the classification accuracy using Gaussian kernel with parameter C equal to 1.3

kernel degrees. A similar comparison was presented by Mladenic in [5].

The classification performance, as it can be observed, is not improved when the number of features increases. We notice that there is a slight increase in the accuracy when we raise the percentage of features from the initial set from 4% (475 features) to 7% (1309 features) for polynomial kernel. The accuracy doesn't increase for a larger percentage of selected features. More than this, if more than 42% of the features were selected, the accuracy slightly decreases. This can occur because the additional features are noisy. As we expected, for Random features selection the value of the accuracy is very poor in comparison with the other methods. The other methods, Information Gain and SVM feature selection obtained comparable results. SVM has slightly better results in comparison with IG (Fig. 1, Fig. 2) for polynomial kernels and obtains best results using a small number of features.

The training time for polynomial kernel with degree 2 and SVM_FS method increases from 11.52 seconds for 475 features to 14.56 seconds for 1306 features and to 46.55 seconds for 2488 features. Thus for fast learning we need a small numbers of features and as it can be observed from Fig. 1 with SVM-FS method we can obtain better results with a small number of features Also the time needed for training with features selected with IG are usually greater than the times needed for training with features selected with SVM_FS (for example for 1309 features we need 14.56 seconds for SVM-FS versus 26.42 seconds for IG).

For Gaussian kernel the time is on average (for all made testes) with 20 minutes greater then the time needed for training the polynomial kernel for both features selected with IG or SVM_FS. The numbers are given for a Pentium IV at 3.4 GHz, with 1GB DRAM and 512KB cache, and WinXP.

### *B. Multi-Class Classification*

For extending the SVM algorithm from two-class classification to multi-class classification typically one of two methods is used: "One versus the rest", where each topic is

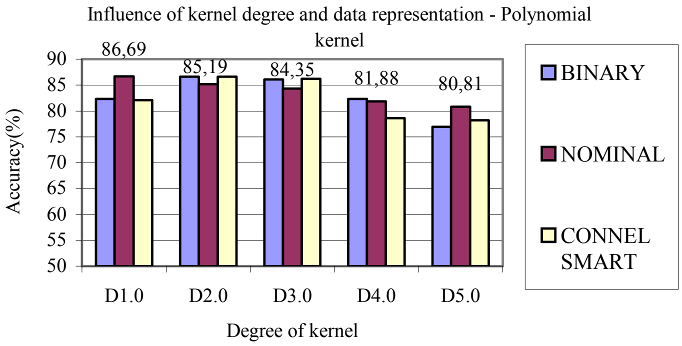


Fig. 3 Influence of data representation and degree of the kernel for polynomial kernel

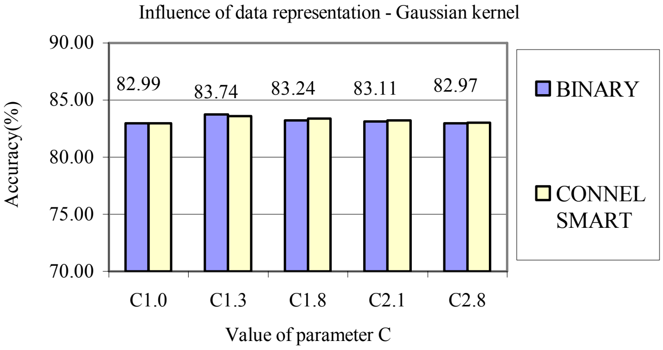


Fig. 4 Influence of data representation and parameter C for Gaussian kernel

separated from the remaining topics, and "One versus the one", where a separate classifier is trained for each pair of topics. The Reuter's database contains strongly overlapping classes and assigns almost all samples in more than one class. Therefore we chose the first method for multi-class classification. Also we tested the method "one versus the one", but the obtained results are not as good. Also the training time doesn't decrease so much because there are more decision functions to be learned even for small datasets.

In the training phase for each topic a decision function is learned. In the evaluating phase each sample is tested with each decision function and is classified in the class having the greatest value. The obtained results are compared with the known Reuter's classification.

In order to find a good combination of kernel type, kernel degree and data representation we run ten tests: five tests for a polynomial kernel with a kernel degree between 1 and 5, and respectively five tests for a Gaussian kernel with different values for the parameter C (1.0, 1.3, 1.8, 2.1 and 2.8). In [6] we report additional results. In Fig. 3 we present results obtained for polynomial kernel and SVM feature selection method with a data set with 1309 features, which was proven to be the best number (see Fig. 1, 2).

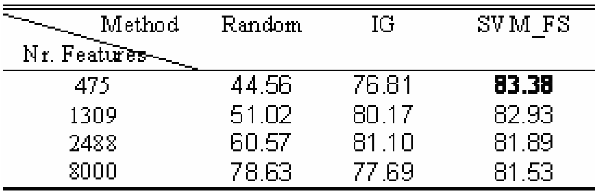
Fig. 3 shows that text files are generally linearly separable in the input space (if the input space has the right dimensionality) and the best results were obtained for a linear kernel and for a small kernel degree using a nominal representation of the input data.

For the same dimension of the feature space (1309 features), the Information Gain method achieved an average accuracy (computed for all three types of data representation) of 80.17% in comparison with the SVM feature selection method that achieved an average accuracy of 82.93% (for Random selection method an average accuracy of 51.02% was achieved). In Table I we present all average accuracies obtained and we can observe that the SVM_FS method obtains better results for each dimension of the data set. Also we can observe that the average accuracy doesn't increase when the dimension of the set increases (especially for SVM_FS). The SVM_FS method obtains best results with a small dimension of the features space (83.38% for 475 features) in comparison with IG that needs more features (2488 features for 81.10%) for obtain the best results.

In Fig. 4 we present results obtained for Gaussian kernel for two types of data representation and for five distinct value of parameter C, using a data set with 1309 features obtained with

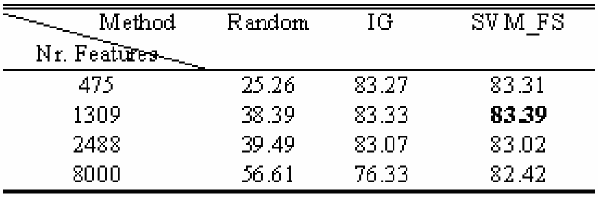
SVM_FS method. Into Gaussian kernel (Fig. 4) we add a parameter that represents the number of elements greater then zero (parameter “n” from equation 4). Nominal representation (equation 5) represents all weight values between 0 and 1. When parameter “n” is used, all the weights become very close to zero involving very poor classification accuracies (for example, due to its almost zero weight, a certain word really belonging to the document, might be considered to not belong to that document). So we don’t present here the results obtained using the nominal representation.

TABLE I
AVERAGE ACHIEVED OVER ALL DATA SETS TESTED FROM POLYNOMIAL KERNEL AND NOMINAL REPRESENTATION



| Method<br>Nr. Features | Random | IG | SVM_FS |
|---|---|---|---|
| 475 | 44.56 | 76.81 | **83.38** |
| 1309 | 51.02 | 80.17 | 82.93 |
| 2488 | 60.57 | 81.10 | 81.89 |
| 8000 | 78.63 | 77.69 | 81.53 |

TABLE II
AVERAGE ACHIEVED OVER ALL DATA SET TESTED FOR GAUSSIAN KERNEL



| Method<br>Nr. Features | Random | IG | SVM_FS |
|---|---|---|---|
| 475 | 25.26 | 83.27 | 83.31 |
| 1309 | 38.39 | 83.33 | **83.39** |
| 2488 | 39.49 | 83.07 | 83.02 |
| 8000 | 56.61 | 76.33 | 82.42 |

In Table II we compute the average over all tested values for the Gaussian kernel. As it can be observed from the table the results obtained here with SVM like feature selection method are closer than Information Gain, but in all cases are greater.

As for polynomial kernel, for Gaussian kernel SVM feature selection method obtains better results in comparison with IG. On the other hand, results obtained with IG for a Gaussian kernel are increasingly more significantly comparing with results obtained with polynomial kernel and IG (see Table I and Table II, IG columns). In comparison with results obtained using the polynomial kernel the results obtained using a Gaussian kernel are identical for SVM (see Tables I and II, SVM_FS columns).

In all presented results when we used SVM technique (in feature selection step with SVM and classification step) we used kernels presented into (3) and (4) with correlating parameters (d and b for polynomial kernel and n with input vectors for Gaussian kernel. In [8] we showed that this method assures better results in almost all cases.

## VI. CONCLUSION AND FURTHER WORK

In this paper, we investigated whether feature selection methods can improve the accuracy of document classification. Three types of feature selection methods were tested and three types of input data representations were used. Simulations were developed using a powerful classification technique based on kernels, i.e., the Support Vector Machine. In the case of multi-class classification, the best results were obtained when we chose a small (but relevant) dimension of the data set. After selecting relevant features, we showed that using between 4% to 7% from the total number of features, the classification accuracies are significantly better (from 85.19% to 86.69% for SVM_FS method). If we further increase the number of features to more than 10%, the accuracy does not improve or even decreases (86.47% for 2488 and 85.96 for 8000 features). When we used SVM_FS, better classification accuracy is obtained using a small number of features (83.38%, for 475 features representing about 3% from the total number of features) - needing small training time. Generally speaking, the SVM feature selection method was better than IG and Random methods. We have also observed that the polynomial kernel obtains better results when we used a nominal data representation and the Gaussian kernel obtains better results when we used Cornell Smart data representation. The best accuracy was obtained by the Polynomial kernel with a degree of one (86.69% and nominal representation) in comparison with Gaussian kernel that obtained only 83.74% accuracy for C=1.3 and Cornell Smart representation. Also we showed that the training classification time increases only by 3 minutes, as the number of features increases from 485 to 1309 and increases by 32 minutes when number of features increases from 1309 to 2488.

Work is ongoing to use Genetic algorithm with SVM in feature selection step in order to improve quality of selected features. An interesting natural extension of our algorithm might be an adaptation for Web mining application, in order to extract and categorized online news.

## ACKNOWLEDGEMENTS

The first two authors would like to express thanks to SIEMENS AG, CT IC MUNCHEN, Germany, especially to Dr. H. C. Hartmut RAFFLER, for his generous support that he has provided in developing this work.

## REFERENCES

[1] S. Chakrabarti, “Mining the Web- Discovering Knowledge from hypertext data”, Morgan Kaufmann Press, 2003.

[2] G. Forman, “A Pitfall and Solution in Multi-Class Feature Selection for Text Classification”, Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.

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

[4] T. Mitchell, “Machine Learning”, McGraw Hill Publishers, 1997.

[5] D. Mladenic, J. Brank, M. Grobelnik and N. Milic-Frayling, “Feature Selection Using Support Vector Machines”, The 27th Annual International ACM SIGIR Conference (SIGIR2004), pp 234-241, 2004.

[6] D. Morariu, “Classification and Clustering using Support Vector Machine”, 2nd PhD Report, University „Lucian Blaga“ of Sibiu, September, 2005, http://webspace.ulbsibiu.ro/ daniel.morariu/html/Docs /Report2.pdf.

[7] D. Morariu, L. Vintan, “A Better Correlation of the SVM kernel’s Parameters”, Proceeding of The 5th RoEduNet International Conference, Sibiu, June 2006.

[8] C. Nello, J. Swawe-Taylor, “An introduction to Support Vector Machines”, Cambridge University Press, 2000.

[9] J. Platt, “Fast training of support vector machines using sequential minimal optimization”. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods – Support Vector Learning, pages 185-208, Cambridge, MA, 1999, MIT Press.

[10] Reuters Corpus: http://about.reuters.com/researchandstandards/corpus/. Released in November 2000.

[11] B. Schoelkopf, A. Smola, “Learning with Kernels, Support Vector Machines”, MIT Press, London, 2002.