Aspects Concerning SVM Method's Scalability

Daniel MORARIU[1], Maria VINŢAN[2], and Lucian VINŢAN[1]

[1] "Lucian Blaga" University of Sibiu, Computer Science Department,
Emil Cioran street, no. 4, Sibiu 550025, Romania

[2] "Lucian Blaga" University of Sibiu, Electric and Electronic Engineering Department

{daniel.morariu, maria.vintan, lucian.vintan}@ulbsibiu.ro

Summary. In the last years the quantity of text documents is increasing continually and automatic document classification is an important challenge. In the text document classification the training step is essential in obtaining a good classifier. The quality of learning depends on the dimension of the training data. When working with huge learning data sets, problems regarding the training time that increases exponentially are occurring. In this paper we are presenting a method that allows working with huge data sets into the training step without increasing exponentially the training time and without significantly decreasing the classification accuracy.

Key words: Text Mining, Classification, Clustering, Support Vector Machine

1 Introduction

While more and more textual information is available online, effective retrieval is difficult without good indexing and summarization of documents' content. Documents' categorization is one solution to this problem. The task of documents' categorization is to assign a user defined categorical label to a given document. 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 word occurrences in a document represents the normalized value of the corresponding component in the document's vector.

In this paper we are investigating how classification accuracy is influenced using only relevant selected input vectors subset belonging to a larger data training set. The answer will show if the classification method is scalable or not. We are using classifiers based on Support Vector Machine (SVM) techniques. They are less vulnerable to degrade when the dimensionality of the feature space is increasing, and have been shown effective in many classification tasks. The SVM is actually based on learning with kernels and support vectors.

We are developing a strategy in seven steps that trains the classifiers on a reduced data set without significant accuracy decrease comparing with training on the larger initial data set but reducing the learning time. In designing this strategy we were inspired by a method presented in [12] which uses a tree structure to hierarchically group similar databases articles, at different abstract levels. Despite this strategy was not recommended by the authors to be used on text documents, we modified it in order to group the similar text documents into a single level.

In this experiment we are using an original implementation of SVM algorithm with some improvements presented by us in [3]. In order to represent the vectors and to select the relevant features we are using a feature selection method based on SVM, already presented in [4], [6]. For training and testing part we are using Reuters' database [10].

Section 2 contains the prerequisites for the work that we are presenting in this paper. In sections 3 we are presenting the proposed scalability method and in section 4 the main results of our experiments. The last section debates and concludes on the most important obtained results and proposes some further work.

2 Experimental Framework

2.1 Support Vector Machine

The Support Vector Machine (SVM) is a classification technique based on statistical learning theory [9], [8] that was applied with great success in many challenging non-linear classification problems processing 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 itself. Looking at a two-dimensional space 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:

$$f(x) = sgn(\langle \mathbf{w}, \Phi(x) \rangle + b) \tag{1}$$

where $\mathbf{w}$ is the weight vector, orthogonal to the hyperplane, "b" is a scalar that represents the margin of the hyperplane, "x" is the current tested sample, "$\Phi(x)$" is a function that transforms the input data into a higher dimensional feature space and $\langle \cdot, \cdot \rangle$ represents the dot product. *Sgn* is the sign function. If $\mathbf{w}$ has unit length, then $\langle \mathbf{w}, \Phi(x) \rangle$ is the length of $\Phi(x)$ along the direction of $\mathbf{w}$. Generally $\mathbf{w}$ will be scaled by $\|\mathbf{w}\|$. In the training part the algorithm needs to find the normal vector "$\mathbf{w}$" that leads to the largest "b" of the hyperplane.

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 separated from the remaining topics, and "One versus the one", where a separate classifier is trained for each class pair. We selected the first method for two reasons: First, our preliminary experiments show that the first method obtains better performance, which might be explained by the fact that the Reuters' database contains strongly overlapped classes and assigns almost all samples in more than one class. Second, the overall training time is shorter for the first method.

2.2 The Data-set

Our experiments were 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 data subset. From all documents we selected the documents with the industry code value equal to "System software". We obtained 7083 files that are represented using 19038 features and 68 topics. We represent a document as a vector of words, applying a stop-word filter (from a standard set of 510 stop-words [13]) and extracting the word stem [13]. From these 68 topics we have eliminated those topics that are poorly or excessively represented obtaining 24 different topics and 7053 documents that were split randomly in a training set (4702 samples) and a testing set (2351 samples).

2.3 Kernel Types

The idea of the kernel trick is to compute the norm of the difference between two vectors in a higher dimensional feature space without representing them in this space. We are using in our selected classifiers two types of kernels each of them with different parameters: polynomial and Gaussian kernels [3]. For the polynomial kernel we vary only the degree of the kernel and for the Gaussian kernel we change the parameter C according to the following formulas (x and x' being the input vectors):

Polynomial,

$$k(x, x') = (2 \cdot d + \langle x \cdot x' \rangle)^d \tag{2}$$

where d being the only parameter to be modified and representing the kernel's degree:

Gaussian (radial basis function RBF),

$$k(x, x') = \exp(-\frac{\|x - x'\|^2}{n \cdot C}) \tag{3}$$

where 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 [3].

3 Methods for determining the SVM Algorithm's Scalability

The original learning step that uses the Support Vector Machine technique to classify documents is split in the next 7 smaller steps:

1. We normalize each input vector in order to have the sum of elements equal to 1, using the following formula:

$$TF(d, t) = \frac{n(d, t)}{\sum_{\tau=0}^{19038} n(d, \tau)} \tag{4}$$

   where $TF(d, t)$ is the term's frequency, $n(d, t)$ is the number of times that term t occurs in document d, and the denominator represents the sum of terms that occur in the entire document d.
2. After normalization we compute the Euclidian distance between each input vector and each center of the group (*representative vector*) that was created up to this moment. Thus, we compute each distance and we keep the smallest obtained distance (linear clustering). If this distance is smaller than a predefined threshold we will introduce the current sample into the winner group and recompute the representative vector for that group; if not, we will create a new group and the current input vector will be the representative vector for that new group.
3. After grouping, we create a new training dataset with all representative vectors. This set will be used in the classification step. In this step we are not interested in the classification accuracy because the vectors are not the original vectors. Here, we are interested only in selecting relevant vectors (the vectors that have an effective contribution to the classification).
4. We are doing a feature selection step on this reduced set (each of them having 19038 features). For this step we are using SVM_FS method presented in [4]. After computing all weights we select 1309 features because, as we showed in [5], this number of features produces optimal results.
5. The resulted smaller vectors are used in a learning step. For this step we use polynomial kernel with degree equal to 1 and nominal data representation. We use the polynomial kernel because it usually obtains a smaller number of support vectors in comparison with the Gaussian kernel [5]. We use the kernel's degree equal to 1 because in almost all previous tests we obtained better results with this value.

6. After SVM learning step, besides the classification rules (decision functions) that are obtained, we also obtain the elements that have an effective contribution to the classification (named support vectors in the SVM algorithm). We chose all groups that are represented by those selected vectors and we develop a new set only with vectors from these groups. This new set is a reduced version of the original set containing only *relevant input vectors* that are considered to have a significant influence on the decision function.
7. This reduced vectors set will now be used in the feature selection and classification steps as the original input data.

In the second presented step all training data are grouped based on their similarity. To compute the similarity we use two different methods based on the "winner-takes-it-all" idea [1]. First method computes the center of the class (the representative vector) using arithmetic mean and the second method computes the center of the class using LVQ formula (Learning Vector Quantization [2]) in one step. Depending on the method used to compute the representative vector we are using two vector representation types. In the first method, where the representative vector is computed as an arithmetic mean, it contains the sum of all elements that are included in that group, and a value that represents the total number of samples from that group. In the second method the representative vector is computed using equation (5) for each new sample that is included in the group. The formula for computing the representative vector for the LVQ method is:

$$\vec{\boldsymbol{w}}_i(t+1) := \vec{\boldsymbol{w}}_i(t) + \alpha(\vec{\boldsymbol{x}} - \vec{\boldsymbol{w}}_i(t)) \quad (5)$$

where $\vec{\boldsymbol{w}}_\mathbf{i}$ is the representative vector for the class i, $\vec{\boldsymbol{x}}$ is the input vector and α is the learning rate. Because we want a one step learning and taking into account the small number of samples, we'll choose a relatively great value for the coefficient α.

3.1 Clustering Data Set using Arithmetic Mean

In our presented results we start with an initial set of 7083 vectors. After the grouping step we reduce this dimension at 4474 representative vectors meaning 63% of the initial set. For this reduction we used a threshold equal to 0.2. On this reduced set a classification step for selecting the relevant vectors was made. After the classification the algorithm returns a number of 874 support vectors. Taking those support vectors we created a dataset containing only 4256 relevant samples meaning approximately 60% from the initial set. Using this new reduced set we make a feature selection step, using SVM_FS method, and we select only 1309 relevant features. The reduced set is split in a training set having 2555 samples and in a testing set having 1701 samples.

3.2 Clustering Data Set using the LVQ Algorithm

In order to compare the two presented methods we tried to obtain approximately the same number of vectors with both methods. With the LVQ algorithm for a value of the threshold equal to 0.15 and a learning rate equal to 0.9 we obtained after the first step 3487 groups (representative vectors). We expect this method to work better when huge data sets will be used. In the next step, we trained a classifier in order to select from these representative vectors only those vectors that are relevant. The algorithm returns 912 support vectors and we selected only those support vectors that have the Lagrange multipliers greater than a certain threshold (in our case 0.25). We considered these support vectors as being the most relevant vectors from all representative vectors. We create a dataset that contains only 4333 samples. This number represents approximately 61% from the initial data set. The obtained set is split randomly into a training set of 2347 samples and in a testing set of 1959 samples.

4 Experimental Results

We are presenting here results only for a reduced vector dimension as number of features (1309 features). We are using this dimension because with this number of features we obtained the best results [6] and because we are interested to quantify the reduced data set's influence on the classification accuracy (scalability of SVM's method). Also in our work we are using three different representations of the input data: binary, nominal and Cornell Smart [3].

In the Fig. 1 we are presenting comparative results obtained for Polynomial kernel and nominal data representation for all four sets - the original set noted as SVM-7053, the set obtained using arithmetic mean to compute the representative vector noted as AM-4256, the set obtained using the LVQ method to compute the representative vector noted as LVQ-4333 and the set randomly obtained noted as RAN-4277. The RAN-4277 set is obtained choosing randomly a specified number of samples from the original set [5].

As it can be observed there is a small difference between the results obtained for AM-4256 and LVQ-4333. The difference in the accuracy obtained between the original set and AM-4256 set is on average equal to 1.30% for all kernel degrees. The same average difference is obtained also between the original set and LVQ-4333. When we work with a small degree of the kernel the difference between the original set and AM-4256 set is smaller than 1% but the difference between the original set and LVQ-4333 is greater (1.60%). When the kernel degree increases the results are better with LVQ-4333 comparatively with AM-4256 but usually the difference can be considered insignificant. At average over all kernels' degree and all data representations the difference between the original set and AM-4256 is 1.65% and the difference between the

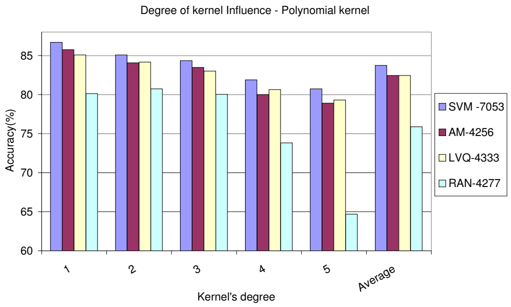

Fig. 1. Comparative results for different set dimensions - Polynomial kernel

original set and LVQ-4333 is 1.64%. We observe that for the same values of kernel's degree, which obtained the best accuracy with the original set, was also obtained the smallest difference between the original set and the reduced set. As we expected, the results obtained using randomly choused set are at average with 8% smaller than the results obtained using the entire set.

In Fig. 2 are presented the results obtained using the Gaussian kernel and Cornell Smart data representation. For this kernel the average accuracy difference between the two sets is greater than for the polynomial kernel case, being at average of 1.89% for AM-4256 and 1.95% for LVQ-4333. The difference between the original set and the randomly reduced set (RAN-4277) is also big for Gaussian kernel (being at average 7%). The smallest difference was obtained with a parameter C equal with 1.8. For this value we obtained the best results in all previous tests (using the original data set). This difference is of 1.5% for AM-4256 and of 1.75% for LVQ-4333. For this type of kernel the method based on LVQ obtains poorly results in almost all cases.

We are reducing the data in the first step at 63% and in the second step at 60% for the first method and respectively to 50% in the first step and 61% in the second step for the LVQ method. With this reduction however the lose in accuracy was about 1% for polynomial kernel and about 1.8% for Gaussian kernel. It is interesting to note that the optimal parameter values (degree or C) are usually the same for the original data set and respectively the reduced one.

Obviously, the time needed for training on a smaller number of samples decreases. For example, for polynomial kernel with degree 1 and Nominal data representation, 1031 seconds are needed to learn using all dataset and 209 seconds using the smallest set. At these 209 seconds we also have to add the time needed to select the support vectors (548 seconds) and the time

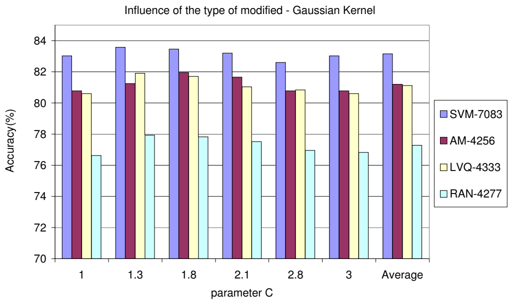

Fig. 2. Comparative results for different set dimensions - Gaussian kernel

needed for grouping data (84 seconds). The last two times occur only once for all the tests with polynomial and Gaussian kernels. The total time for polynomial kernel and degree 1 is 842 seconds. To compute these times, in both cases (with the original set and with the reduced set), we don't take into consideration the time needed for feature selection. Every time the feature selection step starts with 19038 features but in the second case we have a reduced dimension of the set (as number of vectors). Some of these times are also smaller than the first time. These values were obtained using a Pentium IV at 3.2 GHz, 1GB DRAM memory and Win XP. For the second method to obtain the representative vectors (using LVQ) the grouping part takes more time (97 seconds). The time for selecting support vector is 571 seconds. For example the time needed to compute the classification accuracy for polynomial kernel and degree 2 is 232 seconds, so the total time that can be considered is computed as:

$$t_{total\_time} = t_{group\_time} + t_{select\_SV} + t_{classify} \qquad (6)$$

where t_{group_time} is the time needed for grouping data, t_{select_SV} is the time needed for classifying data and finding support vectors and $t_{classify}$ is the time needed for classifying the reduced set. This time can also include the time needed for selecting the features. This time is not included in the classifying time using original data set so it will not be included here.

In Table 1 are presented some training times for each data set. We are interested only in training time because after training the testing time depends only of the testing set dimension and the number of support vectors. For only one sample the response is less than one second.

When we'll use the entire Reuters' data set, learning with all dataset will be impossible due to the huge time and memory needed. Our obtained results

Table 1. Training time for each data set and some training characteristics

Data set	Kernel's Characteristics	t_{group_time}	t_{select_SV}	$t_{classify}$	$t_{training\_total}$
SVM-7083	POL, D2.0, BIN	-	-	-	1532.57
SVM-7083	POL, D1.0, NOM	-	-	-	1031.21
SVM-7083	POL, D1.0, CS	-	-	-	1107.64
SVM-7083	RBF, C2.8, BIN	-	-	-	4492.95
SVM-7083	RBF, C3.0, CS	-	-	-	4481.65
AM-4256	POL, D2.0, BIN	84	548.46	263.36	895.82
AM-4256	POL, D1.0, NOM	84	548.46	209.62	842.08
AM-4256	POL, D1.0, CS	84	548.46	215.56	848.02
AM-4256	RBF, C2.8, BIN	84	548.46	511.87	1144.33
AM-4256	RBF, C3.0, CS	84	548.46	513.81	1146.27
LVQ-4333	POL, D2.0, BIN	97	571.43	289.40	957.83
LVQ-4333	POL, D1.0, NOM	97	571.43	232.53	900.96
LVQ-4333	POL, D1.0, CS	97	571.43	250.67	919.10
LVQ-4333	RBF, C2.8, BIN	97	571.43	599.14	1267.57
LVQ-4333	RBF, C3.0, CS	97	571.43	618.04	1286.47

will be useful in choosing the adequate parameters for the learning step. We don't run those tests with all databases because they usually take a lot of time. For example, for the entire Reuters database we only started the first step of feature extraction and after this step we obtained 806791 vectors, each of them having 310033 dimensions (features) and a total of 103 distinct topics.

5 Conclusions and Further Work

In the real life when working with text documents we need to work with huge sets of documents to obtain good classification accuracy. In this paper we are proposing and developing a strategy to work with large text documents sets. This strategy doesn't increase exponentially the training time, actually it decreases and doesn't loose substantially in classification accuracy. A method to reduce the number of vectors from the input set and make two learning steps in order to consider the learning step finished was developed and tested. We noticed that the classification accuracy decreases at average with only 1% for polynomial kernel and about 1.8% for Gaussian kernel when the dataset is reduced at 60% of the entire dataset. If the reduction of the set was randomly made (RAN-4277) at a dimension smaller with 40% than the original set the lost in classification is at average of 8% for the polynomial kernel and of 7% for the Gaussian kernel.

A major issue that occurs in all classification and clustering algorithms is that they are reluctant to fit in the real spaces. For instance they have a problem dealing with new documents for which none of the features are in the previous feature set (the product between the new features set and the previous feature set is an empty set). As a further improvement we will try to develop tests with families of words and use as features only a representative of each family. In this way the number of features will be significantly reduced and thus we can increase the number of files that can be classified further on. In order to achieve this we could use the WordNet database, which contains a part of the families of words for the English language.

Acknowledgements

We would like to thank to SIEMENS AG, CT IC MUNCHEN, Germany, especially to Vice-President Dr. H. C. Hartmut RAFFLER, for his generous and various supports that he has provided in developing this work.

References

1. Hung, C., Wermter, S., Smith, P., Hybrid Neural Document Clustering Using Guided Self-Organization and WordNet, IEEE Computer Society, 2003
2. Kohonen, T., Self-Organizing Maps, Second edition, Springer Publishers, 1997
3. Morariu, D., Vintan, L., A Better Correlation of the SVM Kernel's Parameters, Proceedings of the 5th RoEduNet IEEE International Conference, ISBN (13) 978-973-739-277-0, Sibiu, June, 2006
4. Morariu, D., Vintan, L., Tresp, V., Feature Selection Method for an Improved SVM Classifier, Proceedings of the 3rd International Conference of Intelligent Systems (ICIS'06), ISSN 1503-5313, vol. 14, pp. 83-89, Prague, August, 2006
5. Morariu, D., Relevant Characteristics Extraction, 3rd PhD Report, University "Lucian Blaga" of Sibiu, October, 2006, http://webspace.ulbsibiu.ro/daniel.morariu /html/Docs/Report3.pdf
6. Morariu, D., Vintan, L., Tresp, V., Evaluating some Feature Selection Methods for an Improved SVM Classifier, International Journal of Intelligent Technology, Volume 1, no. 4, ISSN 1305-6417, pages 288-298, December, 2006
7. Chakrabarti, S., Mining the Web. Discovering Knowledge from Hypertext Data, Morgan Kaufmann Publishers, USA, 2003
8. Nello C., Shawe-Taylor, J., An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000
9. Scholkopf Bernhard, Smola Alexander, Learning with Kernels, Support Vector Machine, MIT Press, London, 2002
10. Misha Wolf and Charles Wicksteed- Reuters Corpus: http://www.reuters.com/researchandstandards/corpus/ Released in November 2000 accessed in June 2005
11. Vapnik, V., The nature of Statistical learning Theory, Springer New York, 1995
12. Yu, H., Yang, J., Han, J., Classifying Large Data Sets Using SVM with Hierarchical Clusters. In SIGKDD03 Exploration: Newsletter on the Special Interest Group on Knowledge Discovery and Data Mining, ACM Press, Washington, DC, USA, 2003
13. http://www.cs.utexas.edu/users/mooney/ir-courses/ - Information Retrieval Java Application