DISTRIBUTED ENSEMBLE LEARNING IN TEXT CLASSIFICATION

Catarina Silva, Bernardete Ribeiro
University of Coimbra, CISUC, Department of Informatics Engineering, Coimbra, Portugal

Uroš Lotrič, Andrej Dobnikar
University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia

Keywords: Text Mining, Cluster Computing.

Abstract: In today's society, individuals and organizations are faced with an ever growing load and diversity of textual information and content, and with increasing demands for knowledge and skills. In this work we try to answer part of these challenges by addressing text classification problems, essential to managing knowledge, by combining several different pioneer kernel-learning machines, namely Support Vector Machines and Relevance Vector Machines. To excel complex learning procedures we establish a model of high-performance distributed computing environment to help tackling the tasks involved in the text classification problem.
The presented approach is valuable in many practical situations where text classification is used. Reuters-21578 benchmark data set is used to demonstrate the strength of the proposed system while different ensemble based learning machines provide text classification models that are efficiently deployed in the Condor and Alchemi platforms.

1 INTRODUCTION

Information overload is becoming one of the major concerns in computer science research nowadays. Individuals are increasingly complaining about the burden of excessive information, like spam email or never ending web search engine results. Therefore, Text Classification has become one of the key techniques for handling and organizing the increasing overload of digital data (Sebastiani, 2002), developing a need for fast and accurate text classifiers.

In the following paper two methods presenting state-of-the-art performances in text classification are used: Support Vector Machines (Vapnik, 1998) and Relevance Vector Machines (Tipping, 2001). Despite their widespread success, a limitation of both algorithms is their mathematical complexity, involving a quadratic programming problem on a dense square matrix which size increases with the number of samples in the data set. To circumvent the approximation error caused by approximate solutions to the quadratic problem use of committees of learning machines, also known as ensembles, is suggested.

An ensemble is started by creating base classifiers with necessary accuracy and diversity. There exist several methods to create the set of elements in an ensemble, such as, use of data mining models with different learning parameters (Joachims, 2007), use of different training samples (Sebastiani, 2002) or use of different preprocessing methods. The conjugation of their results can also be accomplished in a number of ways, like weighted average or majority voting.

To handle the computational cost, caused by using ensemble strategies on already complex learning machines, a cluster environment is used, where, with increasing number of available computing cycles the overall computing time is retained or even reduced, while the classification performance is mostly improved.

The rest of the paper is organized as follows. Next section introduces several text classification issues, including collection used in the experiments. Section three briefly explains learning machines and ensemble strategies. Section four details the deployment of text classification tasks into the cluster environment. Section five presents the results in terms of speedups and classification accuracy. Finally, conclusions and outline directions for future research are given.

Silva C., Ribeiro B., Lotrič U. and Dobnikar A. (2008).
DISTRIBUTED ENSEMBLE LEARNING IN TEXT CLASSIFICATION.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 420-423
DOI: 10.5220/0001680604200423

2 TEXT CATEGORIZATION

Text categorization can be defined as an assignment of natural language texts to one or more predefined categories, based on their content. Automatic text categorization can play an important role in information management tasks, such as text retrieval, routing and filtering. To accomplish automatic text categorization (Baeza-Yates and Ribeiro-Neto, 1999), the set of documents, typically strings of characters, has firstly to be converted to an acceptable representation that the learning machine can handle, and features are usually reduced and/or extracted. Afterwards a data mining phase takes place, as represented in Figure 1. More thoroughly, the task of text categorization can

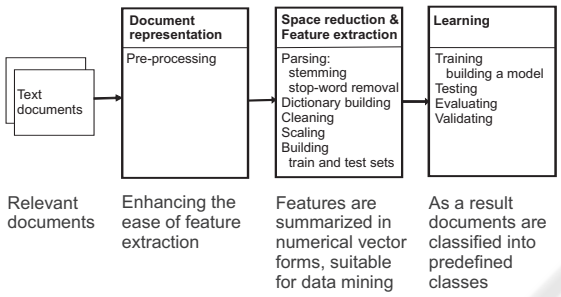

Figure 1: Automatic text Categorization.

be divided into several sub-tasks: A) pre-processing, B) parsing by applying stemming and removing stop words (Silva and Ribeiro, 2003), C) dictionary building with the terms and their document frequency, D) cleaning less frequent words, E) scaling, F) building the train and test sets, G) training, H) testing, I) application of ensemble strategies where partial classifiers are joined together to gain from synergies between them and J) evaluation of classifiers.

For the experiments the Reuters-215768 collection of articles (R21578) was used, which is publicly available at http://kdd.ics.uci.edu/databases /reuters21578/reuters21578.html. It is a financial corpus with news articles averaging 200 words each. R21578 collection has about 12000 articles, classified into 118 possible categories. We use only 10 categories (earn, acq, money-fx, grain, crude, trade, interest, ship, wheat, corn), which cover 75% of the items and constitute an accepted benchmark. R21578 is a very heterogeneous corpus, since the number of articles assigned to each category is very varying. There are articles not assigned to any of the categories and articles assigned to more than 10 categories. On the other hand there are categories with only one assigned article and others with thousands of assigned articles.

3 KERNEL-BASED LEARNING MACHINES AND ENSEMBLE STRATEGIES

Support Vector Machines (SVM) and Relevance Vector Machines (RVM) that show the state-of-art results in several problems are used in conjunction with the ensemble learning techniques for the purpose of text classification.

Support Vector Machines. were introduced by Vapnik (Vapnik, 1998) based on the Structural Risk Minimization principle, as an alternative to the traditional Empirical Risk Minimization principle. Given N input-output samples, $(\mathbf{x}_i, t_i)$, $i = 1, \ldots, N$, a general two-class or binary classification problem is to find a classifier with the decision function $y(\mathbf{x})$, such that $t_i = y(\mathbf{x}_i)$, where $t_i \in \{-1, +1\}$ is the class label for the input vector $\mathbf{x}_i$. From the multiple hyper-planes that can separate the training data without error, a linear SVM chooses the one with the largest margin. The margin is the distance from the hyperplane to the closest training examples, called support vectors. The set of support vectors also includes the training examples inside the margin and the misclassified ones.

SVM Ensemble. We explored different parameters for SVM learning (Joachims, 2007), resulting in four different learning machines: (i) linear default kernel, (ii) RBF kernel, (iii) linear kernel with trade-off between training error and margin set to 100, and (iv) linear kernel with the cost-factor, by which errors in positive examples outweight errors in negative examples, is set to 2.

Relevance Vector Machines. (RVM), proposed by Tipping (Tipping, 2001), are probabilistic non-linear models that use Bayesian theory to obtain sparse solutions for regression and classification. The RVM have an identical functional form to the Support Vector Machines, but provide probabilistic classification. The number of relevance vectors does not grow linearly with the size of training set and the models are usually much sparser, resulting in faster performance on test data at a comparable generalization error. The overall training complexity is $O(N^3)$, implying long lasting learning phase in case of huge sample sizes.

RVM Ensemble. The size and the number of the training sets used in RVM ensemble modeling depend on the available computational power, but more training examples usually results in more diversity and better performance achieved. In our case, seven

smaller training sets were constructed, each consisting of 1000 articles, randomly sampled from the available training set. From each training set a model is learned, and these models constitute the ensemble individual classifiers (Sebastiani, 2002). A majority voting scheme is implemented to determine the ensemble output decision, taking as output value the average value of the classifiers that provided the majority decision.

4 DEPLOYMENT IN THE DISTRIBUTED ENVIRONMENT

One of the important issues of our approach to parallelization of text categorization tasks was to use the existing stand alone sequential code in highest possible extent. Therefore, we targeted on Grid platforms (Berman et al., 2003) using middleware packages which allow complex scheduling of task using existing applications. For easier comparison cluster of homogenous computers was used for testing.

Distributed Environment. The homogenous computing cluster used in the experiments consists of 16 machines having 3 GHz Pentium IV processors and 1GB internal memory, internally connected with 1 Gb Ethernet. Condor (http://www.cs.wisc.edu/condor) and Alchemi (http://www.alchemi.net) middleware software packages are currently installed on the machines to automatically schedule and deploy jobs specified by users. While Condor, targeting to big computing systems, supports only console-type applications, the Alchemi Grid, pointing to small distributed systems, supports also applications with the graphical user interface.

Task Dependencies. For text categorization tasks explained in Section 2, corresponding tasks for distributed environment are created based on partitioning, communication and agglomeration principles (Quinn, 2003). Having in mind the nature of the problem and the performance issues of the underlying cluster, the task dependencies shown in Fig. 2 and Fig. 3 were obtained. The letters in circles denote the tasks given in Section 2, while the numbers refer to the partitions of the tasks which can be run in parallel. Tasks with apostrophes denote gathering of partitioned tasks from previous steps. In case of SVM, the ensemble is created from four classifiers, each having different learning parameters. Therefore, four classifiers run in parallel (tasks FGH), each of them on all ten categories. Finally, ensemble task (I), accounted

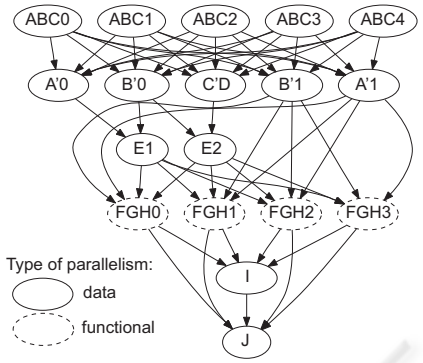

Figure 2: Task dependencies for SVM ensemble.

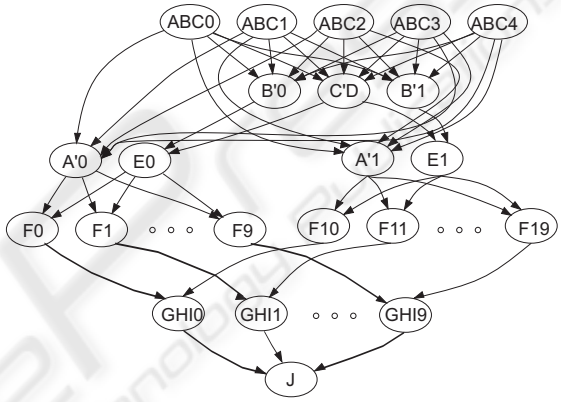

Figure 3: Task dependencies for RVM ensemble.

for synergies between the classifiers, makes a separate decision for each category.

Alternatively, RVM ensembles are build from equal classifiers using different training samples. In this case it is more convenient to partition task based on classification categories, with ensemble strategies running sequentially in each of the given partitions (tasks GHI).

5 EXPERIMENTAL RESULTS

Ensemble strategies with SVM and RVM models were applied to the R21578 collection. Experiments were conducted in both Condor and Alchemi distributed environments while a sequential approach was used for comparison. To ensure statistical significance each experiment was repeated 30 times. The results were characterized from two different aspects – the processing time and the classification performance.

As can be observed in Fig. 4 Alchemi shows improvement in processing times when compared with

sequential approach. The processing times on Alchemi platform slightly surpass those on Condor platform. The probable reason is that the Alchemi environment is much simpler, having less demanding services, thus better dealing with tasks where file transfer prevails over execution burdens. As the learning burden of RVM is much larger, the improvement in processing times is also more significant. Considering the number of processors available and that a very complex and not fully parallelizable problem was undertaken, the resulting speedups represent a real improvement towards the initial goal, i.e., to deploy a complex text classification task in a cluster environment efficiently.

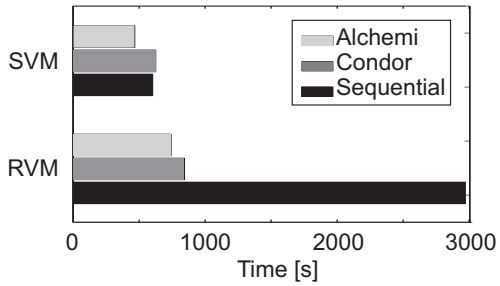

Figure 4: Ensemble processing times on R21578 collection.

Classification results are given in terms of van Rijsbergen F_1 measure (van Rijsbergen, 1979). For better assessment of proposed ensemble strategies, classification performances of single machines are also given. Regarding classification performance on R21578 collection, SVM in general provide better average results than RVM, as can be observed in Table 1. It must be stressed, however, that RVM use only a fraction of the training examples due to the computational constraints.

Table 1: F_1 performance results for single and ensemble machines on R21578 collection.

Machine	SVM	RVM
Single	0.80	0.71
Ensemble	0.84	0.69

6 CONCLUSIONS

We have proposed a strategy to deploy text classification in a distributed environment. The main contributions of the paper is in the development of a distributed environment for text classification processing and in an ensemble strategy of kernel-based learning.

The ensemble models of SVM and RVM learning machines mainly improved the known classification performances without penalizing overall processing times by using the distributed environment setup and available computing cycles. A gain from synergies between the ensemble classifiers is thus obtained. This constitutes an important step forward towards improving textual information classification.

The task and data distributions described in the paper were performed with the Condor and Alchemi platforms. The respective speedups in the several learning settings were compared with the sequential approach. It is shown that the best classification results are obtained with the SVM ensemble, followed by the RVM ensemble. In terms of speedup, the Alchemi gives the best results in all learning models.

We suggest our future work should investigate the modifications necessary for dynamic text classification with the goal to discover interesting trends among news topics. Next, a parallel programming approach with MPI and OpenMP should also be interesting in terms of possible additional speedups.

ACKNOWLEDGEMENTS

Portuguese Science and Technology Foundation Project POSI/SRI/41234/2001 and Slovenian Research Agency Projects L2-6460 and L2-6143 are gratefully acknowledged for partial financial support.

REFERENCES

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley, Wokingham, UK.

Berman, F., Fox, G. C., and Hey, A. J. G., editors (2003). *Grid Computing: Making the Global Infrastructure a Reality*. Wiley, Chichester.

Joachims, T. (2007). Svm light web page. http://svmlight.joachims.org.

Quinn, M. (2003). *Parallel Programming inC with MPI and OpenMP*. McGraw Hill.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Silva, C. and Ribeiro, B. (2003). The importance of stop word removal on recall values in text categorization. In *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pages 1661–1666, Portland.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research I*, pages 211–214.

van Rijsbergen, C. (1979). *Information Retrieval*. Butterworths.

Vapnik, V. (1998). *The Nature of Statistical Learning Theory*. Springer, Berlin.