Text Classification on a Grid Environment

Valeriana G. Roncero, Myrian C. A. Costa and Nelson F. F. Ebecken

Cidade Universitária - Centro de Tecnologia - Bloco I, Sala I-248
Ilha do Fundão – P.O. Box 68516 Rio de Janeiro, RJ - CEP 21941-972 - Brazil
{valery, myrian}@nacad.ufrj.br, nelson@ntt.ufrj.br

Abstract. The enormous amount of information stored in unstructured texts cannot simply be used for further processing by computers, which typically handle text as simple sequences of character strings. Text mining is the process of extracting interesting information and knowledge from unstructured text. One key difficulty with text classification learning algorithms is that they require many hand-labeled documents to learn accurately. In the text mining pattern discovery phase, the text classification step aims at automatically attribute one or more pre-defined classes to text documents. In this research, we propose to use an algorithm for learning from labeled and unlabeled documents based on the combination of Expectation-Maximization (EM) and a naïve Bayes classifier on a grid environment, this combination is based on a mixture of multinomials, which is commonly used in text classification. Naïve Bayes is a probabilistic approach to inductive learning. It estimates the a posteriori probability that a document belongs to a class given the observed feature values of the documents, assuming independence of the features. The class with the maximum a posteriori probability is assigned to the document. Expectation-Maximization (EM) is a class of iterative algorithms for maximum likelihood or maximum a posteriori estimation in problems with unlabeled data. The grid environment is a geographically distributed computation infrastructure composed of a set of heterogeneous resources. Text classification mining methods are time-consuming by using the grid infrastructure can bring significant benefits in learning and the classification process.

Keywords: grid computing, text classification, expectation-maximization and naïve bayes.

1 Introduction

Text mining is a relatively new practice derived from Information Retrieval (IR) [1, 2] and Natural Language Processing (NLP), Baeza-Yates *et al* [3]. The strict definition of text mining includes only the methods capable of discovering new information that is not obvious or easy to find out in a document collection, i.e., reports, historical documents, e-mails, spreadsheets, papers and others. Text mining executes several processes, each one consisting of multiple phases, which transform

or organize an amount of documents in a systematized structure. These phases enable the use of processed documents later, in an efficient and intelligent manner. The processes that compose the text mining can be visualized in fig. 1 that is summarized version of the figure model from Han *et al* [4] on page 6.

Text classification has become one of the most important techniques in text mining. The task is to automatically classify documents into predefined classes based on their content. Many algorithms have been developed to deal with automatic text classification. One of the common methods is the naïve Bayes, Mitchell [5]. Although the naïve Bayes works well in many studies [6, 7, 8], it requires a large number of labeled training documents for learning accurately. In the real world task, it is very hard to obtain the large labeled documents, which are mostly produced by humans. Nigam *et al.* [9] apply the Expectation-Maximization (EM) algorithm to improve the accuracy of learned text classifiers by augmenting a small number of labeled training documents with a large pool of unlabeled documents. The EM algorithm uses both labeled and unlabeled documents for learning. Their experimental results show that using the EM algorithm with unlabeled documents can reduce classification error when there is a small number of training data.

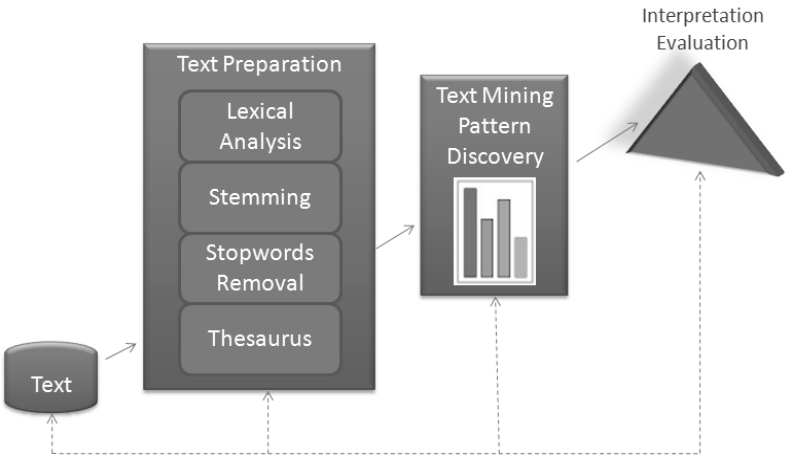

Fig. 1. shows a summary of the text mining phases.

Unfortunately, the EM algorithm is too slow when it performs on very large document collections. In order to reduce the time spent, we propose to use the grid infrastructure to improve the computational time in learning and classifying process. The text classification task uses an algorithm based on the combination of EM algorithm and the Naïve Bayes classifier, Dempster *et al* [10]. This can bring significant benefits. Implementation of text mining techniques in distributed environment allows us to access different geographically distributed data collections and perform text mining tasks in distributed way.

This paper is organized as follows. In section 2, we present an overview of text classification task with the classification algorithms. In section 3, we briefly present an overview of grid computing. Section 4 describes the distributed implementation of

naïve Bayes classifier via the EM algorithm on a grid and we briefly conclude on Section 5.

2 Text Classification

Text categorization or classification aims to automatically assign categories or classes to unseen text documents [11, 12], some classification techniques are naïve Bayes classifier [5], *k*-nearest neighbor, Yang [13], and support vector machines, Joachims [14]. The naïve Bayes algorithm requires a large number of labeled training documents, but to obtain training labels is expensive, while large quantities of unlabeled documents are readily available. The combination of EM algorithm and a naïve Bayes classifier can make use of unlabeled documents to training. This new algorithm first trains a classifier using the available labeled documents, and probabilistically labels the unlabeled documents. It then trains a new classifier using the labels for all the documents, and iterates to convergence, fig 2.

In this section, we briefly review the naïve Bayes classifier and the EM algorithm that is used for making use of unlabeled data.

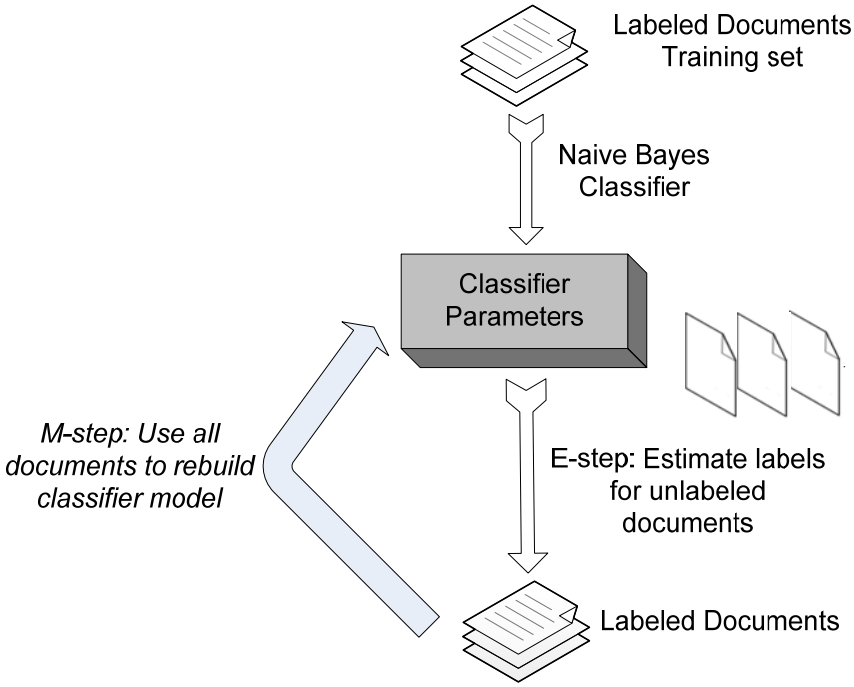

Fig. 2. Expectation-Maximization algorithm with Naïve Bayes classifier.

2.1 Naïve Bayes Classifier

Naïve Bayes Classifier is a type of Bayesian learning algorithm, which by default assumes observations are independent. It is easy to build a Naïve Bayes Classifier when you have a large number features. Researchers have shown that Naïve Bayes Classifier is competitive with other learning algorithms in many cases and in some cases it outperforms the other methods [8]. Learning in Naïve Bayes Classifier involves estimation of the parameters for a classifier, using the labeled document only. The classifier then uses the estimated parameters to classify unobserved documents.

First we will introduce some notation to describe text. Let D be a set of text documents $D = \{d_1, d_2, d_{|D|}\}$, and c_k be a possible class from a set of predefined classes $C = \{c_1, c_2, c_{|C|}\}$. We first transform the probability $P(c_k \mid D)$ using Bayes' rule,

$$P(c_k \mid D) = P(c_k) \times \frac{P(D \mid c_k)}{P(D)} \tag{1}$$

Class probability $P(c_k)$ can be estimated from training data. However, direct estimation of $P(c_k|D)$ is impossible in most cases because of the sparseness of training data.

By assuming the conditional independence of the elements of a vector, $P(D|c_k)$ is decomposed as follows,

$$P(D \mid c_k) = \prod_{j=1}^{k} P(d_j \mid c_k), \tag{2}$$

where d_j is the j^{th} element of a set of text documents D. Then eqn (1) becomes

$$P(c_k \mid D) = P(c_k) \times \frac{\prod_{j=1}^{k} P(d_j \mid c_k)}{P(D)}. \tag{3}$$

With this equation, we can calculate $P(c_k|D)$ and classify D into the class with the highest $P(c_k|D)$.

Note that the naïve Bayes classifier assumes the conditional independence of features. This assumption however does not hold in most cases. For example, word occurrence is a commonly used feature for text classification. However, obvious strong dependencies exist among word occurrences. Despite this apparent violation of the assumption, the naïve Bayes classifier exhibits good performance for various natural language processing tasks.

2.2 Expectation-Maximization Algorithm

One disadvantage of the Naïve Bayes Classifier is that it requires a large set of the labeled training documents for learning accurately. The cost of labeling documents is expensive, while unlabeled documents are commonly available. By applying the EM algorithm, we can use the unlabeled documents to augment the available labeled documents in the training process. Figure 3 shows the procedure of modified EM algorithm.

Input: Training Documents Output: Classification Model
1. Train the classifier using only labeled data. 2. Classify unlabeled documents, assigning probabilistic-weight class labels to them. 3. Update the parameters of the model. Each probabilistically labeled document is counted as its probability instead of one. 4. Go back to (2) until convergence.

Fig. 3. Modified EM algorithm.

The EM algorithm is a type of iterative algorithm for maximum likelihood or maximum a posteriori estimation in problems with incomplete data [10, 15, 16]. This algorithm can be applied to minimally supervised learning, in which the missing values correspond to missing labels of the documents, McLachlan *et al* [17]. In our task, the class labels of the unlabeled documents are considered as the missing values.

The EM algorithm consists of the E-step in which the expected values of the missing sufficient statistics given the observed data and the current parameter estimates are computed, and the M-step in which the expected values of the sufficient statistics computed in the E-step are used to compute complete data maximum likelihood estimates of the parameters [10].

The EM algorithm starts using the Naïve Bayes Classifier to initialize the parameters feature probabilities and class priors using the labeled documents. The E-step and M-step are iterated until the change in class labels for the unlabeled documents is below some threshold (i.e. the algorithm converges [16] to a local maximum). The E-step almost dominates the execution time on each iteration, since it estimates the class labels for all the training documents [9].

3 Grid Environment

A grid is a geographically distributed computation infrastructure composed of a set of heterogeneous machines, often with separate policies for security and resource use, Qi *et al* [18], that users can access via a single interface. Grids therefore, provide a common resource-access technology and operational services across widely

distributed virtual organizations composed of institutions or individuals that share resources.

Today grids can be used as effective infrastructures for distributed high-performance computing and data processing, Foster *et al* [19].

3.1 NACAD Grid Environment

The NACAD Grid uses Globus Toolkit 4 (GT4) [20] that is an open source software toolkit user for building grids. It is being developed by the Globus Alliance and many others all over the world. The Globus Toolkit is a widely used middleware in scientific and data-intensive grid applications, and is becoming standard for implementing grid systems. The toolkit addresses security, information discovery, resource and data management, communication, fault detection, portability issues and is based on grid services. Grid services is a technology based on the concepts and technologies of grids and web services and can be defined as a web service that delivers a set of interfaces that follows specific conventions. This technology was originated from the necessity to integrate services through virtual, heterogeneous and dynamic organizations, composed of distinct resources, whether within the same organization or by resource sharing.

3.2 Grid Services

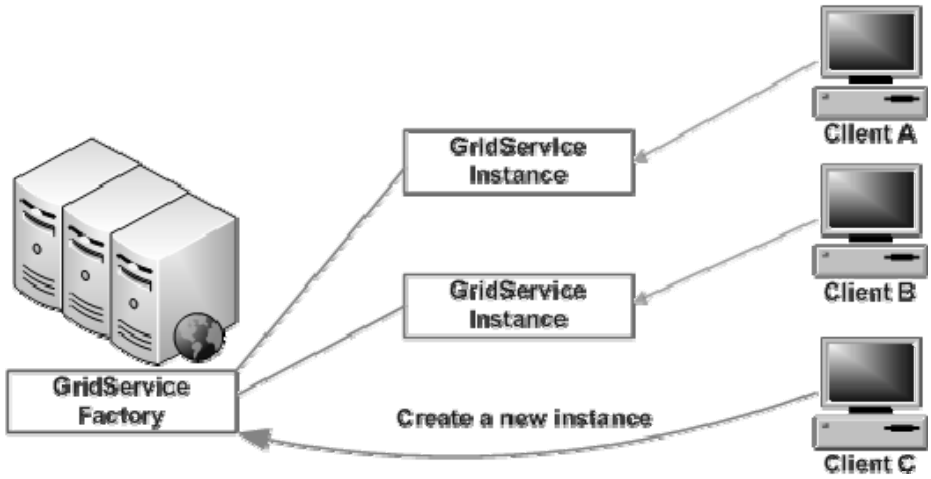

Fig. 4. Grid services

A grid service is a web service that conforms to a set of conventions (interfaces and behaviors) that defines how a client interacts with a grid service. A web service converts an application into a web-application, which is published, found, and used through the web facilitating the communication between applications.

In this work we propose to implement the classifier model as a grid service in the Aîuri Portal [21], which is a framework for a cooperative academic environment for education and research. The components are grid Text Mining services and in the future Data Mining components will be added. The classifier service will be included in the Aîuri Portal showing that many distinct algorithms can be easily added and accessed through this Portal, fig 5.

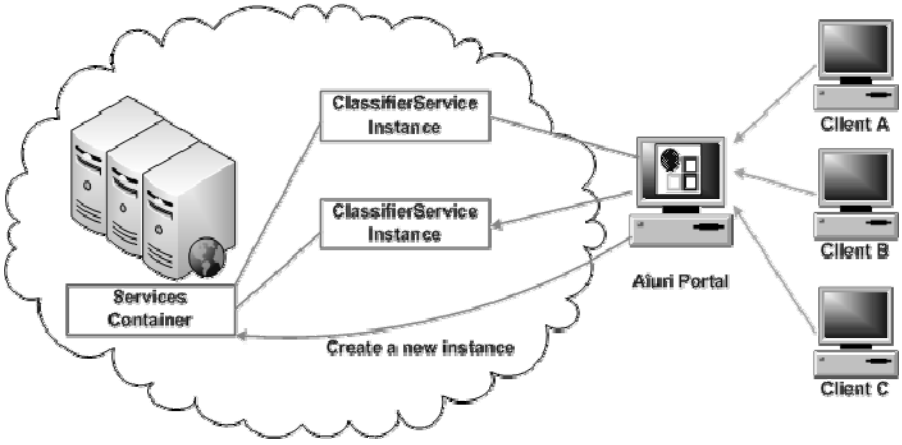

Fig. 5. The classifier service as a component service in the Aîuri Portal.

4 Naïve Bayes Classifier via the EM algorithm on a Grid Environment

The enormous amount of information stored in huge document databases in unstructured format or semi-structured format cannot simply be used for further processing by computers, which typically handle text as sequences of character strings. Text mining provides some methods, like classification, able to extract interesting information and knowledge from unstructured text. One key difficulty with text classification learning algorithms is that they require many hand-labeled documents to learn accurately. Using the Naïve Bayes Classifier via the EM algorithm we can use the unlabeled documents to increase the available labeled documents in the training process. Implementation of text mining techniques in distributed environment allows us to access different data collections that are geographically distributed and perform text mining tasks in distributed way. Figure 6 shows the distributed EM algorithm for text classification on a grid environment.

The algorithm, developed in Java, analyses text documents in XML format. The input set of documents must be in the same consistent format. The algorithm makes some assumptions about the format of input documents that are specified by the user in the input parameters file. Since the XML documents are ASCII text and are clearly marked off from each other, they can be simply put in one large file and provided as input to the algorithm. The documents are marked off from other documents by

distinguishing tags, for example, the user must specify which tag represents: a set of documents, text and on labeled documents the tag used to identify class labels. The input parameters file consist of: which are the files that will be used by the classifier, such as, train labeled and unlabeled files, test and stopwords files and stemmer dictionary; the tags meaning in the documents and some characteristics to be used by the classifier.

Input: Training Documents Output: Classification Model
1. The Portal receives from the user the parameters input file, the labeled documents and the unlabeled documents. 2. The Portal sends a request to the service container to create the grid service instance, which is composed with the Naïve Bayes Classifier and the EM application. 3. The classifier grid service builds the initial global classifier from only the labeled documents using the grid nodes. 4. Each grid node receives a pre-defined set of training documents from the Portal. 5. Iterate until the classifier parameters are still improved: 5.1. E-step: Each grid node estimates the class of each document by using the current global classifier. 5.2. M-step: Each grid node re-estimates its own local classifier given the estimated class of each document. 6. Sum up the local classifier to obtain the new global classifier and return them to all grid nodes.

Fig. 6. The distributed EM algorithm for text classification on a grid environment.

5 Summary

In this study, we propose to use a combination [9] of Expectation-Maximization (EM) [10] and a Naïve Bayes classifier on a grid environment, this combination is based on a mixture of multinomials, which is commonly used in text classification. Naïve Bayes is a probabilistic approach to inductive learning. It estimates the a posteriori probability that a document belongs to a class given the observed feature values of the documents, assuming independence of the features. The class with the maximum a posteriori probability is assigned to the document. Expectation-Maximization (EM) is a class of iterative algorithms for maximum likelihood or maximum a posteriori estimation in problems with unlabeled data. Using a grid environment we can reduce the classifier estimation processing time and distribute the documents to speed up classification task.

Acknowledgements

The authors would like to thank the Center for Parallel Computations (NACAD) at the Graduate School and Research in Engineering (COPPE), Federal University of Rio de Janeiro for providing the computational resources for this research.

References

1. Salton, G. & McGill, M.J.: Introduction to Modern Information Retrieval, McGraw-Hill Book Company (1983)
2. Baeza-Yates, R. & Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press Books (1999)
3. Kao, A. & Poteet, S.R.: Natural Language Processing and Text Mining. Springer-Verlag (2007)
4. Han, J. & Kamber M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2001)
5. Mitchell, T.M.: Bayesian Learning (Chapter 6). Machine Learning, McGraw-Hill: New York, pp. 154--200 (1997)
6. Joachimes, T., A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: Proc. of the 14th Int. Conf. on Machine Learning, pp. 143--151 (1997)
7. Lewis, D. & Ringuette, M.: A comparison of two learning algorithms for text categorization. In: 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 81--93 (1994)
8. McCallum, A. & Nigam, K.: A comparison of events models for Naive Bayes text classification. In: AAAI-98 Workshop on Learning of Text Categorization, AAAI Press, pp. 41--48 (1998).
9. Ningam, K., McCallum, A., Thrun, S. & Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning, vol. 39(2/3), pp. 103--134 (2000)
10. Dempster, A.P., Laird, N.M. & Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. In: Journal of the Royal Statistic Society, Series B, vol. 39(1), pp. 1--38 (1977)
11. Yang, Y. & Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proc. of the 14th Int. Conf. on Machine Learning, Morgan Kaufmann Publishers, pp. 412--420 (1997)
12. Mitchell, T. M.: Machine Learning. McGraw-Hill, New York (1997)
13. Yang, Y: An evaluation of statistical approaches to text categorization. In: Journal of Information Retrieval, vol. 1, pp. 67--88 (1999).
14. Joachims, T.: Text categorization with Support Vector Machines: learning with many relevant features. In: Proc. of the ECML-98, Spring Verlag, pp. 137--142 (1998)
15. Duda, R., Hart, P. & Stork, D.: Pattern Classification. Wiley-Interscience (2001)
16. Bilmes, J.A.: A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021. In: International Computer Science Institute, University of California, Berkeley (1998)

17. McLachlan, G.J. & Krishnan, T: The EM algorithm and extensions. John Wiley & Sons: New York (1997)
18. Qi, L., Jin, H., Foster, I. & Gawor, J.: HAND: Highly Available Dynamic Deployment Infrastructure for Globus Toolkit 4, In: Proc. of the 15th Euromicro Int. Conf. on Parallel, Distributed and Network-Based Processing, pp. 155--164 (2007).
19. Foster, I., Kesselman, C. & Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations, In: International Journal of Supercomputer Applications, vol. 15(3) (2001).
20. The Globus Toolkit, http://www.globus.org/toolkit/
21. Serpa, A.A., Roncero, V.G., Costa, M.C.A., and Ebecken, N.F.F: Text Mining Grid Services for Multiple Environments. In: 8th Int. Conf. on High Performance Computing for Computational Science (VECPAR 2008), Springer Lecture Notes in Computer Science, vol. 5336 pp 576--587, Toulouse, France, 2008.