Text mining on a grid environment

V. G. Roncero, M. C. A. Costa & N. F. F. Ebecken
COPPE/Federal University of Rio de Janeiro, Brazil

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 to 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 document, assuming independence of the features. The class with the maximum a posteriori probability is assigned to the document. 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, but using the grid infrastructure can bring significant benefits in the learning and classification process.

Keywords: grid computing, text classification, Expectation-Maximization, naïve Bayes.

WIT Transactions on Information and Communication Technologies, Vol 42, © 2009 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA090021

1 Introduction

Text mining is a relatively new practice derived from Information Retrieval (IR) [1, 2] and Natural Language Processing (NLP) (Kao and Poteet [3]). The strict definition of text mining includes only the methods capable of discovering new information that is not obvious or easy to find in a document collection, i.e., reports, historical documents, e-mails, spreadsheets, papers and others.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, which is a summarized version of the figure model from Han and Kamber [4, p. 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 [6–8], it requires a large number of labeled training documents for accurate learning. 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.

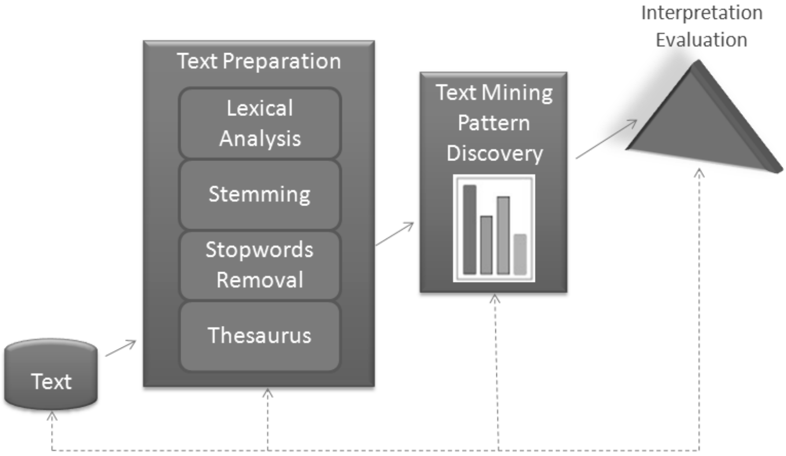

Figure 1: Summary of the text mining phases.

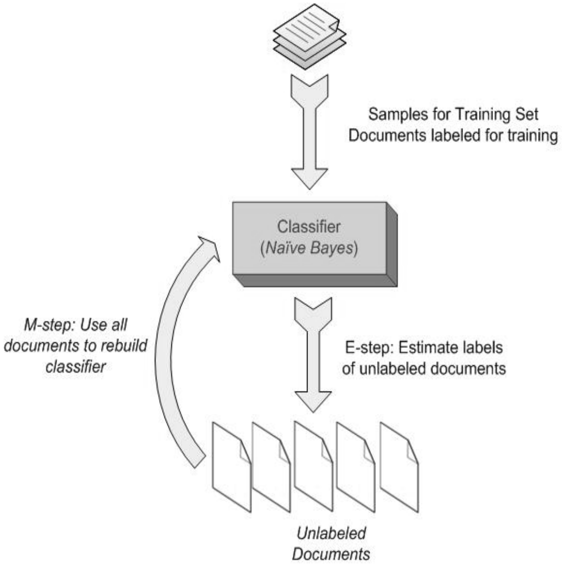
Samples for Training Set
Documents labeled for training
Classifier
(*Naïve Bayes*)
M-step: Use all documents to rebuild classifier
E-step: Estimate labels of unlabeled documents
Unlabeled Documents

Figure 2: EM algorithm with naïve Bayes classifier.

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 the learning and classifying process. The text classification task uses an algorithm based on the combination of the EM algorithm and the naïve Bayes classifier (Dempster *et al* [10]). This can bring significant benefits. The implementation of text mining techniques in a distributed environment allows us to access different geographically distributed data collections and perform text mining tasks in a distributed way.

This paper is organized as follows. In section 2, we present an overview of the 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 the naïve Bayes classifier via the EM algorithm on a grid and we briefly conclude in 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 the 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 the EM algorithm and the naïve Bayes classifier can make use of unlabeled documents in 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.

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 EM 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-weighted 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.
```

Figure 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 and Krishnan [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]).

In this study we use the Globus Toolkit 4 (GT4) [20], which 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, and portability issues.

Today, Globus and the other grid tools are used in many projects worldwide. Although most of these projects are in scientific and technical computing fields, and a growing number of grid projects in education, industry, and commerce are being implemented.

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 (fig 4).

```
1. One grid node builds the initial global classifier from only the labeled
documents and sends the global classifier to the others grid nodes.
2. Each grid node receives a pre-defined set of training documents from the
storage.
3. Iterate until convergence
    3.1 E-step: Each grid node estimates the class of each document by using
    the current global classifier.
    3.2 M-step: Each grid node re-estimates its own local classifier given the
    estimated class of each document.
    3.3. Sum up the local classifier to obtain the new global classifier and return
    them to all grid nodes.
```

Figure 4: The distributed EM algorithm for text classification on a grid.

The Globus Toolkit provides a number of components for performing data management. Data management tools (GridFTP, RFT, RLS) are concerned with the location, transfer, and management of distributed data [21]. GridFTP protocol provides a secure way to transfer data in a grid. RFT (Reliable File Transfer) is a Web Services Resource Framework (WSRF) [22] compliant web service for managing multiple data transfers. The Replica Location Service

(RLS) [23] maintains and provides access to mapping information from logical names for data items onto target names. These target names may represent physical locations of data items, or an entry in the RLS may map to another level of logical naming for the data item. The RLS is intended to be one of a set of services for providing data replication management in grids. In additional of these components, the LIGO Data Replicator (LDR) [24, 25] will be used. LDR is a collection of some components provided by the Globus project with some extra logic to pull the components together. This minimum collection of components is necessary for fast, efficient, robust, and secure replication of data. The Globus components included are: GridFTP, Globus Replica Location Service (RLS) and a metadata service developed by the LDR team but based on a prototype Globus Metadata Catalog Service (MCS) [26] for organizing useful information about the data files, especially as it pertains to when and where the data should be replicated.

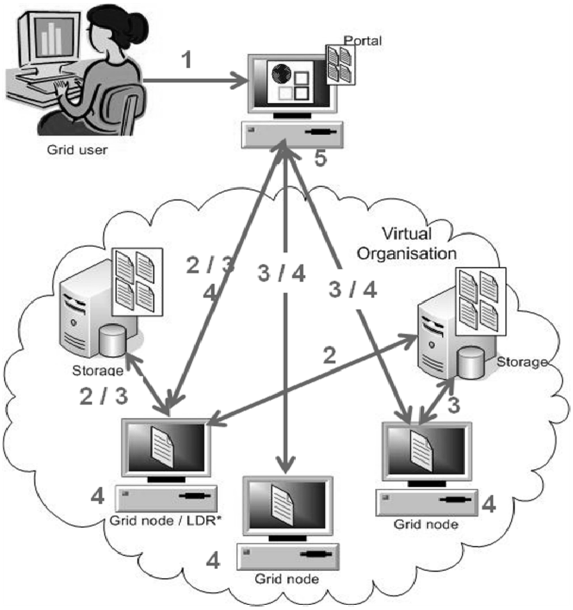

Figure 5: Distributed text mining classification process on a grid.

Figure 5 shows the distributed text mining classification process on a grid model. The grid user owns a grid certificate, which provides him the grid credentials [27] to log into the grid and submit jobs to it, which is done by means of a Portal, accessible from the user's workstation. After logged in, the user can access his documents or public documents that are stored in the grid. He submits to the Portal information about the documents that will be analyzed (1). The Portal uses the LDR queries to find out whether there is a local copy of the documents, if not, RLS tells the Portal where the documents are in the grid (2). Then the LDR system generates a request to copy the documents to the local storage system and registers the new copy in the local RLS server. One grid node builds the global classifier with the labeled documents set and sends it to the Portal. The grid nodes receive from the Portal the phases to training the local classifier with the global classifier and using the RFT service to copy the replicas of the training set of unlabeled documents from the storage to the grid nodes (3). After each grid node received the steps and its own set of training documents, it estimates the class of a subset of documents using the current classifier. Then it re-estimates its own classifier with given the estimated class of each document and sends it to the Portal (4). When estimation is concluded, the Portal sums each local classifier and returns them to all grid nodes until the classifier remains unchanged (5).

5 Summary

In this study, we propose to use a combination [9] of 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.

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. *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] Nigam, K., McCallum, A., Thrun, S. & Mitchell, T., Text classification from labeled and unlabeled documents using EM. *Machine Learning*, **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. *Journal of the Royal Statistic Society, Series B*, **39(1)**, pp. 1–38, 1977.

[11] Yang, Y. & Pedersen, J.O., A comparative study on feature selection in text categorization. *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. *Journal of Information Retrieval*, **1**, pp. 67-88, 1999.

[14] Joachims, T., Text categorization with Support Vector Machines: learning with many relevant features. *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. CA: International Computer Science Institute, ICSI-TR-97-021, 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, *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, Int. *Journal of Supercomputer Applications*, **15(3)**, 2001.

[20] The Globus Toolkit, www.globus.org/toolkit/

[21] GT4 Data Management, www.globus.org/toolkit/docs/4.0/data/

[22] The WS-Resource Framework, www.globus.org/wsrf/

[23] Replica Location Service, www.globus.org/toolkit/data/rls/

[24] LIGO Scientific Collaboration Research Group: Ligo Data Replicator. http://www.lscgroup.phys.uwm.edu/LDR/

[25] Chervenak, A., Schuler, R., Kesselman, C., Koranda, S. & Moe, B., Wide area data replication for scientific collaborations, *Proc. of the 6th IEEE/ACM Int. Workshop on Grid Computing (Grid2005)*, 2005.

[26] Metadata Catalog Service, www.globus.org/grid_software/data/mcs.php

[27] GT 4.0: Security: Pre-Web Services Authentication and Authorization, www.globus.org/toolkit/docs/4.0/security/prewsaa/