

Highly Accurate Distributed Classification of Web Documents

JingKuan Song, Hui Gao, LianLi Gao, Yan Fu
School of Computer Science and Engineering,
University of Electronic Science & Technology of China,
Chengdu, Sichuan 610054, China
Email: beautifulma163com@gmail.com

Abstract— With the rapid growth of internet, it is a scientific challenge and a massive economic need to discover an efficient and accurate text classifier for handling tons of online documents. This paper presents a distributed model for efficient web document classifications. In the model, the distributed text classifiers are trained serially with the weights on the training instances, which are adaptively set according to their previous performances. Based on the distributed model, Unequal Bagging (UBagging), an improved technique of bagging for text classifier is also proposed. Results from the experiments show that our approach could gain higher classification accuracy over traditional centralized text classifiers, and require less memory and computational time.

Index Terms—Text classification; Bagging; Distributed environment; Decision tree; Neural network

I. INTRODUCTION

Text classification is an essential component of Web data mining; it can be defined as the assignment of natural language texts to one or more predefined categories, according to their contents. The growth of the information availability, supported by the Internet, made the size of data increase dramatically. This leads to the situation of “information overload and knowledge scarcity”, which results in the necessity of better organizing the overloaded information. Therefore, usage of abundant data to make decision is becoming a field that needs to be studied most deeply nowadays[1].

Automatic text classification can play an important role in information management tasks, such as text retrieval, routing and filtering. Some other applications includes mapping queries to relevant categories and taxonomies, automated indexing of scientific articles using subject categories, selective dissemination of information to relevant user groups, tracking news event about particular topics, spam filtering, identification of document genre, authorship attribution, and etc. the two phases model.

At present, there are many algorithms of data mining, these algorithms aim at specific issues and application fields[2-5]. These algorithms are very effective in some application fields. On the other hand, they have some shortcomings more or less. A main problem is that almost every method studied only the concrete algorithm. But compromise has to be made when the algorithm is improved at the cost of losing many merits of original algorithms. Nowadays the efficiency choke point of the calculating architecture algorithms that used for the classification of web document has emerged in the

centralized environment. As the process is quite exhausted, nearly all algorithms studied and analyzed the efficiency choke point. The parallel and distributed calculating architecture is introduced to solve this problem.

II. GENERAL PROCESS OF TEXT CLASSIFICATION

Fig.1 shows the general process of text classification. Firstly, to achieve the automatic text classification, the web documents with strings of characters has to be converted to an acceptable representation that the learning machine can handle. The most common representation so far is the vector space model(VSM). Each document is indexed with the bag of the terms occurring in it, having as value the frequency the term occurred in the document. Thus, each document is represented as a point in a vector space with one dimension for every term in the vocabulary.

As observed from previous research that keywords work well as features for many text classification tasks. In the feature space representation, the text documents are represented as sequence of words. Feature selection involves tokenizing the text, indexing and feature space reduction. Text can be tokenized using term frequency (TF), inverse document frequency (IDF), or using binary representation. Using these representations the global feature space is determined from entire training document collection. However, the keywords appear in the different position of a document should account for different weight. We adjust tf-idf according to the position of the keywords. Feature selection plays a major role in achieving better classification performance. A significant amount of research has been done on feature selection for better representation of data for text classification. Nigam et al.[6] compared the performance of naive Bayes (NB)

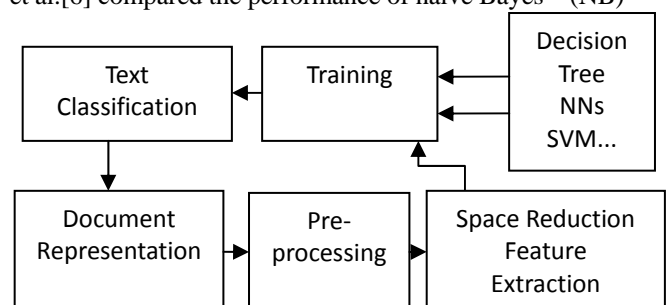


Figure 1. General Process of Text Classification

and expectation maximization (EM) using labeled and unlabeled documents. Some others use PCA to guarantee

the selected candidate attributes to be uncorrelated.

Feature space reduction in a feature selection process improves the accuracy of the learning algorithm performance, decreases the data size, controls the classification time, and avoids over-fitting of the data. There are many ways of feature space reduction such as stemming, stop-word removal, and using information gain or mutual information criteria.

Web document classification is the process of grouping web documents into one or more predefined categories based on their content. The task of text classification has been explored and several learning machines have been used. The number of classes of classifier learning techniques that have been used in text classification is bewildering. These include at the very least probabilistic methods, regression methods, decision tree and decision rule learners, neural networks, batch and incremental learners of linear classifiers, example-based methods, support vector machines, genetic algorithms, hidden Markov models, and classifier committees (which include boosting methods).

A text classifier is automatically generated by a general inductive process (the learner) which, by observing the characteristics of a set of documents. The training set is the set of documents with which the learner builds the classifier. The validation set is the set of documents used to adjust the classifier. The test set is the set on which the effectiveness of the classifier is finally evaluated. In both the validation and test phase, “evaluating the effectiveness” means applying the classifier on a set of pre-classified documents and checking the degree of correspondence between the output of the classifier and the pre-assigned categories.

III. DISTRIBUTED ENVIRONMENT

With the accelerative development of modern society information, data and database has rapidly increased and distributional database has become a tendency. What’s more, the convention data mining system’s performance must be improved to meet user’s need. That demands the abroad application of distributed data mining methods that can parallel to get the result in time.

The distributional data mining (DDM) is carries on the excavation to the distributional data set. The so-called distributional data set refers to certain partial database. They may connect through the local computer and the network. The data mining may carries on in the partial database and the overall situation database is through carries on the excavation to the partial database, carries on the obtained pattern or the knowledge the analysis integration the result set. The main goal of a distributed computing system is to connect users and resources in a transparent, open, and scalable way[7].

Since the performance is one of what we are concerning about mostly, we can design the system as a “Client-Server” model based on star topology, as depicted in Fig.2.

We use the “Client-Server” model instead of “Client-

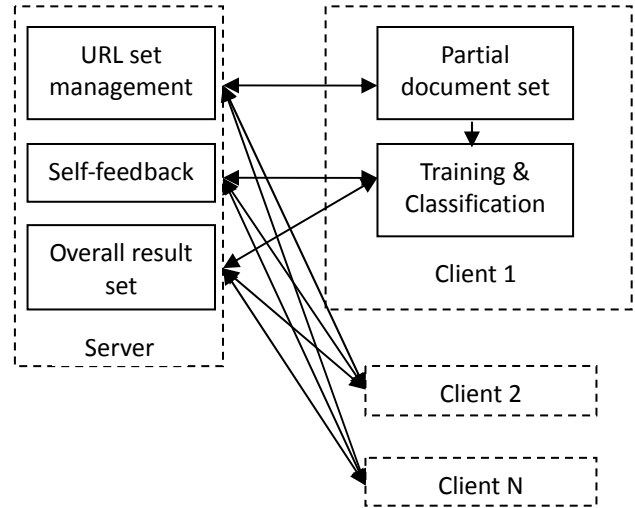


Figure 2. Distributed Environment

Client” model to minimize the communications costs across limited bandwidth interconnection network. It will also simplify the management of the system.

In the Server, we’ll have to manage the URL database and get the final result set generated by UBagging, as described in the next section. We stored all the URLs of the web document in the URL data base. Since that we will adopt gaggling algorithm, so each URL will be sent to more than one client. Considering the efficiency is as important as accuracy, we use only part of the clients to make decision for each one document, that is to say, each URL will be randomly sent to N clients, who will make the final decision to decide which category this document belongs to, and this algorithm is described in the next section.

Self-feedback is used to adjust the reliability of sub-classifier according to its performance in the classifying process, and this algorithm is described in the next section. When the clients get the partial result sets, they will send them back to the server to get the final result set. For each document, we will decide it’s category with UBagging algorithm and then the final overall result will be obtained.

In the Client, we fetch URL set from the Server in turn, and train a sub-classifier as described in the next section. The sub-classifier will be tested and will be used to classify the web documents and the result will be sent back to the server.

IV. TEXT CLASSIFICATION: UNEQUAL BAGGING

A. Classifier Reliability Evaluation

To simplify the process, we suppose the training set can be divided into two opposite types, type A and not type A, so that the result of the classifier is ‘yes’ or ‘no’.

The performance of the classifier can be measured by Sensitivity (SE) and Specificity (SP), which can be defined as follows respectively:

$$SE = ra / aa. \quad (1)$$

$$SP = rb / ab \quad (2)$$

The symbol ra is the number of the random documents that are classified under A , and the decision is correct. The aa is the total number of document that belongs to type A , rb is the number of the random documents that are not classified under A , and the decision is correct, ab is the total number of documents that do not belong to type A .

The reliability of the classifier's result, that is the posterior probability, is related not only to the performance of the classifier, but also to the prior probability, which means the probability of a random document that belongs to type A without taking account any information about the document. According to Bayes' theorem, the posterior probability (PoP) can be defined as follows:

$$PoP = PrP * SE / (PrP * SE + (1 - PrP) * SP) \quad (3)$$

PrP is the prior probability of the training set, generated from the real training set.

B. Classifier Construction

The training set was obtained from a data base of news items published over last a few months. The web documents we chose for training consists of the following 10 categories: Conflicts and War; Crime, Law and Justice; Disaster and Accident; Economy, Business and Finance; Education; Environment; Health; Politics; Religion; Social Issue. Before we use it for training, the data base was manually filtered in order to discard uncategorized items and items belonging to more than one category. This resulted in the clean data base of 1437 news items with uniquely assigned categories.

Our goal is to learn classifiers using inductive learning methods, the machine learning methods are various, and among which two of most popular ones are decision tree and neural network. From literatures, we can see that their advantages and drawbacks are almost complementary. A decision tree contains internal nodes and leaf nodes, and each internal node tests one attribute and each branch from a node makes a decision to select one test value for the attribute. The decision can be extended for general categorically value functions. To achieve better classification, we developed a combined approach for building a decision tree with the neural network as its categorically value function. We chose BP for the categorically value function of the decision tree because it's a multi-layer feed-forward neural network consisted of some hidden layers, and it can give a better solution for non-linear problems[8]. We will train the BP Neural Network with the training set before building the decision tree. For convenience, we used ID3 algorithm [10] to train the system and generate a decision tree that can classify the documents into 11 categories. In a single client, classification of a document is done by recursively testing the weights of the internal nodes of the decision tree T against those in the corresponding output vector of trained BP Neural Network until a leaf is reached.

C. Unequal Bagging

Bagging is an ensemble method that uses random re-sampling of a dataset to construct models. It's useful to improve the accuracy for the unstable algorithms like

decision tree. The different sub-classifier has the same weight when voting for the final decision. However, the results generated from different sub-classifiers have different reliability, so that the sub-classifiers should have unequal weight when making the ultimate decision. The reliability of a sub-classifier is based on its performance in the testing process. Because of the size limit or quality limit of the testing set, the reliability of a sub-classifier may not be accurate or differ from the real use. Then we use self-feedback mechanism to make the reliability more reasonable. Applying UBagging to linear classifiers makes it possible to use the coefficient of the discriminant function $C(x)$, which will make the final decision for the category that an unknown document belongs to. The improved bagging algorithm(UBagging) is as follows:

Step 1: Training & validation

Repeat for $n = 1 \dots N$

a) Take a bootstrap replicate Xn of the training set T . Xn has the same size with T , but the individual sample is selected randomly.

b) Train the training set to construct linear classifier $Cn(x)$ on Xn with the algorithm presented at the right previous part of this section.

c) Calculate the posterior probability on testing set T_2 for each $Cn(x)$ with the method described in the previous part of this section. Store the posterior probability for each $Cn(x)$ in a vector $PoP[N]$, which has the size of N .

Step 2: Classifying

Repeat for $n = 1 \dots N$

$$C(x) = PoP[n] * Cn(X) \quad (4)$$

$Pop[n]$ is the reliability of classifier n . $Cn(X)$ is the result from the classifier n .

Step 3: Self-feedback

In $C(x)$, the term with the biggest coefficient will be the final category of X .

For those sub-classifiers who made a decision in accords with the final decision, their reliability adjusted as follows:

Repeat for $n = 1 \dots P$

$$PoP(n) = PoP[n] * (1 + 1 / Tt) \quad (5)$$

P is the number of the classifiers who made the correct decision. Tt is the size of the testing set used to test the sub-classifiers.

Analogously, for those sub-classifiers who made a decision unlike the final decision, their reliability adjusted as follows:

Repeat for $n = 1 \dots Q$

$$PoP(n) = PoP[n] * (1 - 1 / Tt) \quad (6)$$

Q is the number of the classifiers who made the wrong decision. Tt is the size of the testing set used to test the sub-classifiers.

V. EXPERIMENT

The experiments described in this paper were performed in our laboratory. It consists of a cluster of 20 machines having 2.74GHz Pentium 4 processor, 1 GB RAM each. One of the machines is the server, and runs

Linux. The remaining 15 machines are clients and they can be booted in Linux or Windows XP operating systems. According to the UBagging algorithm as we described before, we'll use different proportions of the total machine to vote for the final decision. The total machine increase from 3 to 20, and we use 30 percent, 50 percent and 70 percent of the clients to vote for the final result respectively. The performance of the system is measured by the size of data set that can be dealt with in one minute, and we use the units of M/Minute. The result is shown in Fig.3. The result showed that the performance of the distributed environment increases linearly with the increase of the number of the clients.

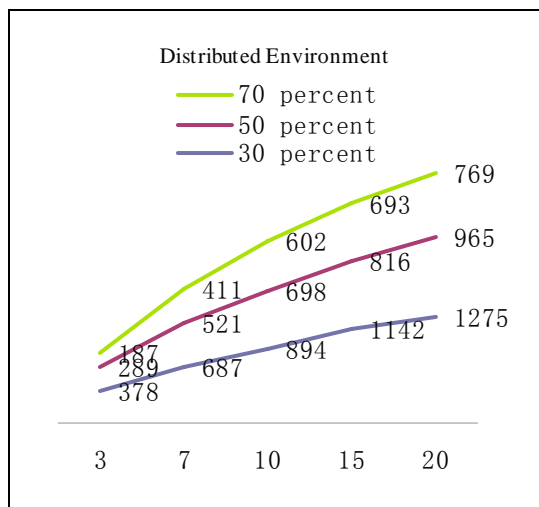


Figure 3. Performance of the Attributed Environment

TABLE 1. Accuracy compare between Bagging & UBagging

clients	Bagging			UBagging			Bagging			UBagging		
	p	r	f	p	r	f	p	r	f	p	r	f
Conflicts & War												
3	79.	89	84.	81	91.	85.	81	92.	83.	84.	92.	82
	2	.2	3	.3	2	6	.3	7	6	1	7	.1
5	82.	90	84.	83	91.	86.	83	91.	84.	84.	93.	84
	3	.2	6	.3	2	6	.1	7	2	1	9	.8
10	82.	92	84.	84	94.	86.	84	92.	85.	86.	94.	85
	6	.2	2	.3	2	6	.1	7	2	1	7	.8
15	83.	93	85.	85	94.	85.	85	93.	86.	86.	94.	89
	5	4	.2	1	.5	2	1	.7	3	8	6	.0
20	84.	92	85.	84	94.	88.	85	93.	87.	87.	93.	90
	0	5	.7	1	.5	5	1	.3	4	8	6	.2
Disaster and Accident												
3	80.	92	86.	82	91.	85.	81	92.	86.	84.	93	85.
	1	.3	3	3	2	6	.3	6	6	1	.7	2
5	81.	91	87.	83	91.	86.	83	93.	87.	85.	94	89.
	2	0	2	.2	7	6	.1	7	2	1	.7	4
10	81.	92	87.	83	92.	86.	83	92.	87.	85.	93	89.
	0	6	.2	4	.1	2	7	.1	2	2	3	.7
15	83.	92	88.	83	93.	87.	85	94.	89.	86.	93	89.
	5	3	4	0	.5	2	1	.7	3	8	3	.7
20	82	93	88.	83	93.	88.	85	94.	89.	86.	93	89.
	0	5	.2	3	.5	2	1	.7	3	5	6	.7
Education												

Classification effectiveness is usually measured in terms of the classic IR notions of precision (p) and recall (r) and the combination of them (f)[9]. P is the probability that if a random document dx is classified under ci , this decision is correct. Analogously, r is defined as the probability that, if a random document dx ought to be classified under ci , this decision is taken. The f is a kind of harmonic average of precision and recall. We will use

these three measurements to compare Bagging and UBagging with the different number of clients. The result is shown in TABLE 1. The results showed that UBagging improved accuracy in text classification task than Bagging.

VI. CONCLUSION

In this paper, we proposed a distributed model for efficient web document classifications and improved the Bagging for text classifier. According to the results of experiments, our approach could gain higher classification accuracy over traditional centralized text classifiers, and require less memory and computational time. The experiment results are encouraging and practically useful.

Although encouraging results have been obtained by using improved bagging on a distributed model, there is still much work remaining to be investigated. In the future, we will do more research to find out better algorithms on the distributed model for more acceptable accuracy.

ACKNOWLEDGMENT

This work is supported by the National High-Tech Research and Development Plan of China under Grant No. 2007AA01Z440.

REFERENCES

- [1] Jingsong Chen, Xiaoying Shi, "An incremental updating algorithm for mining association rules". Computer engineering, Vol 28, No.7, pp.106-107, July 2002.
- [2] May R. A., "Mining association rules between sets of items in large database. Proc. ACM SIGMOD int'l conf. Management of data", Washington, America, pp.207-216, 1993.
- [3] Huang J., Yin Z.B., "Improvement of Apriori algorithm for mining association rules. Journal of university of electronic science and technology", Vol 32. pp.76-79, June 2003.
- [4] Hui Cao. "An algorithm of mining association rules based on vector matrix", Computer engineering and science, Vol 26, No.11, pp.69-74, Nov. 2004.
- [5] Han J., Kamber M., Fang M., "Concept and technology of data mining", China Machine Press, Beijing, 2001.
- [6] K. Nigam, A.K. McCallum, S. Thrun, and T.M. Mitchell, "Learning to Classify Text from Labeled and Unlabeled Documents," Proc. 15th Nat'l Conf. for Artificial Intelligence, pp. 792-799, 1998
- [7] [Http://en.wikipedia.org/wiki/Distributed_computing#Goals_and_advantages](http://en.wikipedia.org/wiki/Distributed_computing#Goals_and_advantages)
- [8] Hui Gao, Yan Fu and Jianping Li, "Classification of Sensitive Web DOCUMENTS", "Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference", pp. 295-298, Dec. 2008
- [9] FABRIZIO SEBASTIANI, "Machine Learning in Automated Text Classification", ACM Computing Surveys, F. 2009