Logitboost of Multinomial Bayesian Classifier for Text Classification

Article *in* International Review on Computers and Software · January 2006

Sotiris Kotsiantis
University of Patras

P. E. Pintelas
University of Patras

Logitboost of Multinomial Bayesian Classifier for Text Classification

S. Kotsiantis[1], E. Athanasopoulou[2], and P. Pintelas[3]

Abstract – *Automated text classification has been considered as a vital method to manage and process a vast amount of documents in digital forms that are widespread and continuously increasing. In general, text classification plays an important role in information extraction and summarization, text retrieval, and question-answering. The Multinomial Bayesian Classifier has traditionally been a focus of research in the field of text learning. This paper increases the accuracy of Multinomial Bayesian Classifier with the usage of the Logitboost technique. We performed a large-scale comparison on benchmark datasets with other state-of-the-art algorithms and the proposed technique had greater accuracy in most cases.* ***Copyright © 2006 Praise Worthy Prize - All rights reserved.***

Keywords*: text mining, learning algorithms, text representation*

I. Introduction

Automatic text classification has always been an important application and research topic since the inception of digital documents. Today, text classification is a necessity due to the very large amount of text documents that we have to deal with daily. In general, text classification includes topic based text classification [1] and text genre-based classification [2]. Intuitively Text Classification is the task of classifying a document under a predefined category. More formally, if d_i is a document of the entire set of documents D and $\{c_1, c_2, ..., c_n\}$ is the set of all the categories, then text classification assigns one category c_j to a document d_i.

Sebastiani gave an excellent review of text classification domain [3]. Thus, in this work apart from the brief description of the text classification we refer to some more recent works than those in Sebastiani's article as well as few articles that were not referred by Sebastiani. Although Multinomial Bayesian method [4] is competitive, the ensembles of Multinomial Bayesian classifiers have not been a focus of research. The explanation is that Multinomial Bayes is a very stable learning algorithm and most ensemble techniques such as bagging is mainly variance reduction techniques, thus not being able to benefit from its combination.

In this study, we combine Multinomial Bayesian method with Logitboost [5], which is a bias reduction technique. As it is well known, Logitboost requires a regression algorithm for base learner. For this reason, we slightly modify Multinomial Bayesian classifier in order to be able to run as a regression method. Finally, we performed a large-scale comparison with other state-of-the-art algorithms on benchmark datasets and we actually took better accuracy in most cases.

A brief description of data-preprocessing of text data before machine learning algorithms can be applied is given in Section 2. Section 3 describes the most well known machine learning techniques that have been applied in text classification. Section 4 discusses the proposed method. Experiment results of the proposed method with other well known classifiers in a number of data sets are presented in section 5, while brief summary with further research topics are given in Section 6.

II. Data preprocessing

The task of constructing a classifier for documents does not differ a lot from other tasks of Machine Learning. The main issue is the representation of a document [6]. In Fig. 1 is given the graphical representation of the Text Classification process.

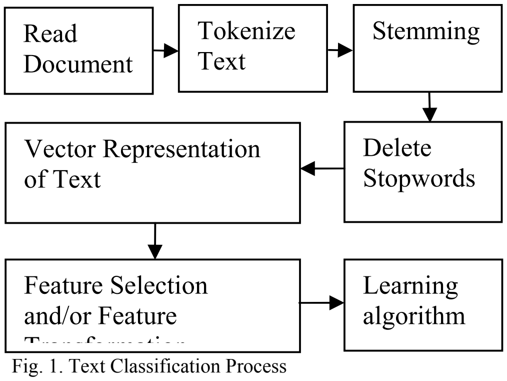
Read Document
Tokenize Text
Stemming
Delete Stopwords
Vector Representation of Text
Feature Selection and/or Feature Transformation
Learning algorithm

Fig. 1. Text Classification Process

Manuscript received September 2006, revised October 2006, accepted November 2006

In the following sub-section the document representation is presented. One particularity of the text categorization problem is that the number of features (unique words or phrases) can easily reach orders of tens of thousands. This raises big hurdles in applying many sophisticated learning algorithms to the text categorization. Thus dimension reduction methods are called for. Two possibilities exist, either selecting a subset of the original features [7], or transforming the features into new ones, that is, computing new features as some functions of the old ones [5]. We examine both in turn in Section II.2 and Section II.3.

II.1. Vector space document representations

A document is a sequence of words [6]. So each document is usually represented by an array of words. The set of all the words of a training set is called vocabulary, or feature set. So a document can be presented by a binary vector, assigning the value 1 if the document contains the feature-word or 0 if the word does not appear in the document. This can be translated as positioning a document in a $R^{|V|}$ space, were $|V|$ denotes the size of the vocabulary V .

Not all of the words presented in a document can be used in order to train the classifier [8]. There are useless words such as auxiliary verbs, conjunctions and articles. These words are called stopwords. There exist many lists of such words which are removed as a preprocess task. This is done because these words appear in most of the documents.

Stemming is another common preprocessing step. In order to reduce the size of the initial feature set is to remove misspelled or words with the same stem. A stemmer (an algorithm which performs stemming), removes words with the same stem and keeps the stem or the most common of them as feature. For example, the words “train”, “training”, “trainer” and “trains” can be replaced with “train”. Although stemming is considered by the Text Classification community to amplify the classifiers performance, there are some doubts on the actual importance of aggressive stemming, such as performed by the Porter Stemmer [3].

An ancillary feature engineering choice is the representation of the feature value [6]. Often a Boolean indicator of whether the word occurred in the document is sufficient. Other possibilities include the count of the number of times the word occurred in the document, the frequency of its occurrence normalized by the length of the document, the count normalized by the inverse document frequency of the word. In situations where the document length varies widely, it may be important to normalize the counts. Further, in short documents words are unlikely to repeat, making Boolean word indicators nearly as informative as counts. This yields a great savings in training resources and in the search space of the induction algorithm. It may otherwise try to discretize each feature optimally, searching over the number of bins and each bin’s threshold.

Most of the text categorization algorithms in the literature represent documents as collections of words. An alternative which has not been sufficiently explored is the use of word meanings, also known as senses. Kehagias et al. using several algorithms, they compared the categorization accuracy of classifiers based on words to that of classifiers based on senses [9]. The document collection on which this comparison took place is a subset of the annotated Brown Corpus semantic concordance. A series of experiments indicated that the use of senses does not result in any significant categorization improvement.

II.2. Feature Selection

The aim of feature-selection methods is the reduction of the dimensionality of the dataset by removing features that are considered irrelevant for the classification [10]. This transformation procedure has been shown to present a number of advantages, including smaller dataset size, smaller computational requirements for the text categorization algorithms (especially those that do not scale well with the feature set size) and considerable shrinking of the search space. The goal is the reduction of the curse of dimensionality to yield improved classification accuracy. Another benefit of feature selection is its tendency to reduce overfitting, i.e. the phenomenon by which a classifier is tuned also to the contingent characteristics of the training data rather than the constitutive characteristics of the categories, and therefore, to increase generalization.

Methods for feature subset selection for text document classification task use an evaluation function that is applied to a single word [11]. Scoring of individual words (Best Individual Features) can be performed using some of the measures, for instance, document frequency, term frequency, mutual information, information gain, odds ratio, $\chi 2$ statistic and term strength [7], [12], [10], [13], [11]. What is common to all of these feature-scoring methods is that they conclude by ranking the features by their independently determined scores, and then select the top scoring features.

On the contrary with Best Individual Features (BIF) methods, sequential forward selection (SFS) methods firstly select the best single word evaluated by given criterion [14]; then, add one word at a time until the number of selected words reaches desired k words. SFS methods do not result in the optimal words subset but they take note of dependencies between words as opposed to the BIF methods. Therefore SFS often give better results than BIF. However, SFS are not usually used in text classification because of their computation cost due to large vocabulary size.

Forman has present benchmark comparison of 12 metrics on well known training sets [10]. According to Forman, BNS performed best by wide margin using 500 to 1000 features, while Information Gain outperforms the other metrics when the features vary between 20 and 50. Concerning the performance of chi-square, it was consistently worse the Information Gain. Since there is no metric that performs constantly better than all others, researchers often combine two metrics in order to benefit from both metrics [10]. Novovicova et al. used SFS that took into account, not only the mutual information between a class and a word but also between a class and two words [15]. The results were slightly better.

Although machine learning based text classification is a good method as far as performance is concerned, it is inefficient for it to handle the very large training corpus. Thus, apart from feature selection, many times instance selection is needed. Fragoudis et al. [16] integrated Feature and Instance Selection for Text Classification. Their method works in two steps. In the first step, their method sequentially selects features that have high precision in predicting the target class. All documents that do not contain at least one such feature are dropped from the training set. In the second step, their method searches within this subset of the initial dataset for a set of features that tend to predict the complement of the target class and these features are also selected. The sum of the features selected during these two steps is the new feature set and the documents selected from the first step comprise the training set.

II.3. Feature Transformation

Feature Transformation varies significantly from Feature Selection approaches, but like them its purpose is to reduce the feature set size [5]. This approach does not weight terms in order to discard the lower weighted but compacts the vocabulary based on feature concurrencies.

Principal Component Analysis is a well known method for feature transformation [17]. Its aim is to learn a discriminative transformation matrix in order to reduce the initial feature space into a lower dimensional feature space in order to reduce the complexity of the classification task without any trade-off in accuracy. The transform is derived from the eigenvectors corresponding. The covariance matrix of data in PCA corresponds to the document term matrix multiplied by its transpose. Entries in the covariance matrix represent co-occurring terms in the documents. Eigenvectors of this matrix corresponding to the dominant eigenvalues are now directions related to dominant combinations can be called “topics” or “semantic concepts”. A transform matrix constructed from these eigenvectors projects a document onto these “latent semantic concepts”, and the new low dimensional representation consists of the magnitudes of these projections. The eigenanalysis can be computed efficiently by a sparse variant of singular value decomposition of the document-term matrix [18]. In the information retrieval community this method has been named Latent Semantic Indexing (LSI) [19]. This approach is not intuitive discernible for a human but has a good performance.

Qiang et al [20] performed experiments using k-NN LSI, a new combination of the standard k-NN method on top of LSI, and applying a new matrix decomposition algorithm, Semi-Discrete Matrix Decomposition, to decompose the vector matrix. The Experimental results showed that text categorization effectiveness in this space was better and it was also computationally less costly, because it needed a lower dimensional space.

The authors of [21] present a comparison of the performance of a number of text categorization methods in two different data sets. In particular, they evaluate the Vector and LSI methods, a classifier based on Support Vector Machines (SVM) and the k-Nearest Neighbor variations of the Vector and LSI models. Their results show that overall, SVMs and k-NN LSI perform better than the other methods, in a statistically significant way.

III. Machine learning algorithms

After feature selection and transformation the documents can be easily represented in a form that can be used by a ML algorithm. Many text classifiers have been proposed in the literature using machine learning techniques, probabilistic models, etc. They often differ in the approach adopted: decision trees, naıve-Bayes, rule induction, neural networks, nearest neighbors, and lately, support vector machines. Although many approaches have been proposed, automated text classification is still a major area of research primarily because the effectiveness of current automated text classifiers is not faultless and still needs improvement.

Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness [22]. However, its performance is often degraded because it does not model text well. Schneider addressed the problems and show that they can be solved by some simple corrections [23]. Mccallum and Nigam [4] proposed the NB-Multinomial classifier with even better results. Klopotek and Woch presented results of empirical evaluation of a Bayesian multinet classifier based on a new method of learning very large tree-like Bayesian networks [24]. The study suggests that tree-like Bayesian networks are able to handle a text classification task in one hundred thousand variables with sufficient speed and accuracy.

Support vector machines (SVM), when applied to text classification provide excellent precision, but poor recall. One means of customizing SVMs to improve recall, is to adjust the threshold associated with an SVM. Shanahan and Roma described an automatic process for adjusting the thresholds of generic SVM [25] with better results.

Johnson et al. described a fast decision tree construction algorithm that takes advantage of the sparsity of text data, and a rule simplification method that converts the decision tree into a logically equivalent rule set [26]. Lim proposed a method which improves performance of kNN based text classification by using well estimated parameters [27]. Some variants of the kNN method with different decision functions, k values, and feature sets were proposed and evaluated to find out adequate parameters. Corner classification (CC) network is a kind of feed forward neural network for instantly document classification. A training algorithm, named as TextCC is presented in [28].

The level of difficulty of text classification tasks naturally varies. As the number of distinct classes increases, so does the difficulty, and therefore the size of the training set needed. In any multi-class text classification task, inevitably some classes will be more difficult than others to classify. Reasons for this may be: (1) very few positive training examples for the class, and/or (2) lack of good predictive features for that class.

When training a binary classifier per category in text categorization, we use all the documents in the training corpus that belong to that category as relevant training data and all the documents in the training corpus that belong to all the other categories as non-relevant training data. It is often the case that there is an overwhelming number of non relevant training documents especially when there is a large collection of categories with each assigned to a small number of documents, which is typically an “imbalanced data problem". This problem presents a particular challenge to classification algorithms, which can achieve high accuracy by simply classifying every example as negative. To overcome this problem, cost sensitive learning is needed [29].

A scalability analysis of a number of classifiers in text categorization is given in [30]. Vinciarelli presents categorization experiments performed over noisy texts [31]. By noisy it is meant any text obtained through an extraction process (affected by errors) from media other than digital texts (e.g. transcriptions of speech recordings extracted with a recognition system). The performance of the categorization system over the clean and noisy (Word Error Rate between ~10 and ~50 percent) versions of the same documents is compared. The noisy texts are obtained through Handwriting Recognition and simulation of Optical Character Recognition. The results show that the performance loss is acceptable.

Other authors [32] also proposed to parallelize and distribute the process of text classification. With such a procedure, the performance of classifiers can be improved in both accuracy and time complexity. Recently in the area of Machine Learning the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers. In the context of combining multiple classifiers for text categorization, a number of researchers have shown that combining different classifiers can improve classification accuracy [33], [34]. Comparison between the best individual classifier and combined methods, observed that the performance of the combined methods is superior [35]. For this reason, we propose a combining technique in this study.

IV. Proposed Algorithm

As we have also mentioned, MultiNomial Bayes classifier can be effectively used in ensemble techniques that perform bias reduction, such as Logitboost. However, Logitboost requires a regression algorithm for base learner. For this reason, we slightly modify MultiNomial Bayes classifier so as to be able to run as a regression method.

In the multinomial model, a document is an ordered sequence of word events, drawn from the same vocabulary V. We assume that the lengths of documents are independent of class. We make a similar naive Bayes assumption: that the probability of each word event in a document is independent of the word’s context and position in the document. Thus, each document d_i is drawn from a multinomial distribution of words with as many independent trials as the length of d_i. This yields the familiar “bag of words” representation for documents. Define N_{it} to be the count of the number of times word w_t occurs in document d_i. Then, the probability of a document given its class is simply the multinomial distribution:

$$P(d_i \mid c_j;\theta) = P(|d_i|)\,|d_i|!\prod_{t=1}^{|V|}\frac{P(w_t \mid c_j;\theta)^{N_{it}}}{N_{it}!}$$

MultiNomial Bayes classifier assigns a probability to every possible value in the target range. The resulting distribution is then condensed into a single prediction. In categorical problems, the optimal prediction under zero-one loss is the most likely value—the mode of the underlying distribution. However, in numeric problems the optimal prediction is either the mean or the median, depending on the loss function. These two statistics are far more sensitive to the underlying distribution than the most likely value: they almost always change when the underlying distribution changes, even by a small amount. For this reason MultiNomial Bayes classifier is not as stable in regression as in classification problems.

Although the predicted variable in regression may vary continuously, for a specific application, it's not unusual for the output to take values from a finite set, where the connection between regression and classification is stronger. The main difference is that regression values have a natural ordering, whereas for classification the class values are unordered. This affects the measurement of error. For classification, predicting the wrong class is an error no matter which class is predicted (setting aside the issue of variable misclassification costs). For regression, the error in prediction varies depending on the distance from the correct value.

Naive Bayes has previously been applied to regression problems by Frank et al. [36]. We discretize the target value into a set of 10 equal-width intervals, and apply MultiNomial Bayes classifier for classification to the discretized data. For prediction, the predicted value is the expected value of the mean class value for each discretized interval (based on the predicted probabilities by MultiNomial Bayes classifier for each interval).

Finally, the proposed algorithm (LogitBoostNB-Multinomial) is summarized in (Fig. 2). In the following section, we present the experiments of the proposed technique with other well known methods.

```
Step 1: Initialization
$F^{(0)}(x) \equiv 0$  committee function:
$p^{(0)}(x) \equiv 1/2$  initial probabilities:
Step 2: LogitBoost iterations
    for m=1,2,...,10 repeat:
A. Fitting the learner
Compute working response and weights for i=1,...,n
    $w_i^{(m)} = p^{(m-1)} \cdot (1 - p^{(m-1)})$
    $z_i^{(m)} = \frac{y_i - p^{(m-1)}(x_i)}{w_i^{(m)}}$
  1. Discretize the target value into a set of 10 equal-width intervals
  2. Fit a regression version of MultiNomial Bayes classifier by
weighted least squares
  $f^{(m)} = \arg\min_f \sum_{i=1}^{n} w_i^{(m)} (z_i^{(m)} - f(x_i))^2$
B. Updating and classifier output
    $F^{(m)}(x_i) = F^{(m-1)}(x_i) + \frac{1}{2} f^{(m)}(x_i)$
    $C^{(m)}(x_i) = Sign\left(F^{(m)}(x_i)\right)$
    $p^{(m)}(x_i) = \frac{1}{1 + e^{-2F^{(m)}(x_i)}}$
```

Fig. 2. The proposed algorithm

V. Comparisons and Results

For the purpose of our study, we used well-known datasets from many domains text datasets donated by George Forman/Hewlett-Packard Labs (http://www.hpl.hp.com/personal/George_Forman/). These data sets were hand selected so as to come from real-world problems and to vary in characteristics (see Table I).

TABLE I.
DESCRIPTION OF THE DATA SETS

Data	instances	features	classes
oh0	1003	3183	6
oh15	913	3101	10
tr23	204	5833	6
re0	1504	2887	13
oh10	1050	3239	10
tr21	336	7903	6
tr11	414	6430	9
re1	1657	3759	25
tr41	878	7455	10
tr12	313	5805	8

There are various methods to determine effectiveness; however, precision, recall, and accuracy are most often used. To determine these, one must first begin by understanding if the classification of a document was a true positive (TP), false positive (FP), true negative (TN), or false negative (FN) (see Table II).

TABLE II.
CLASSIFICATION OF A DOCUMENT

TP	Determined as a document being classified correctly as relating to a category.
FP	Determined as a document that is said to be related to the category incorrectly.
FN	Determined as a document that is not marked as related to a category but should be.
TN	Documents that should not be marked as being in a particular category and are not.

Precision (πi) is determined as the conditional probability that a random document d is classified under ci, or what would be deemed the correct category. It represents the classifiers ability to place a document as being under the correct category as opposed to all documents place in that category, both correct and incorrect:

$$\pi_i = \frac{TP_i}{TP_i + FP_i} \qquad (1)$$

Recall (ρ_i) is defined as the probability that, if a random document d_x should be classified under category (c_i), this decision is taken.

$$\rho_i = \frac{TP_i}{TP_i + FN_i} \tag{2}$$

Accuracy is commonly used as a measure for categorization techniques:

$$A_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{3}$$

Furthermore, precision and recall are often combined in order to get a better picture of the performance of the classifier. This is done by combining them in the following formula:

$$F_\beta = \frac{\left(\beta^2 + 1\right)\pi\rho}{\beta^2\pi + \rho}, \tag{4}$$

where π and ρ denote precision and recall respectively. β is a positive parameter, which represents the goal of the evaluation task. If precision is considered to be more important that recall, then the value of β converges to zero. On the other hand, if recall is more important than precision then β converges to infinity. Usually β is set to 1, because in this way equal importance is given to each presicion and recall.

The most commonly used C4.5 algorithm was the representative of the decision trees in our study. Naive Bayes Multinomial algorithm was the representative of the Bayesian networks. The RIPPER algorithm was the representative of the rule-learning techniques. A well-known learning algorithm for building a neural network - RBF algorithm (Mitchell, 1997) - was the representative of the Artificial Neural Networks. Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the Support Vector Machines. It must be mentioned that we used for the most of the algorithms the free available source code by the book [37].

In our experiments, we used a hybrid sequential forward selection (SFS) to select the best single words evaluated by information gain; then, add ten word at a time until the number of selected words reaches the maximum accuracy in the learning curve. SFS method does not result in the optimal words subset but it takes note of dependencies between words as opposed to the BIF methods. Therefore SFS give better results than BIF. Generally, SFS are not usually used in text classification because of their computation cost due to large vocabulary size. For this reason, we add ten words at a time, instead of one, in an attempt to decrease computation cost.

In order to calculate the classifiers' accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier was trained on the union of all of the other subsets. Then, cross validation was run for each algorithm and the average value of the 10-cross validation was calculated.

In Table III and Table IV, we present the average accuracy of each classifier and the number of features that were used for the induction. In the same tables, we also represent with "*v*" that the proposed LogitBoost-NB-Multinomial algorithm *looses* from the specific algorithm. That is, the specific algorithm performed statistically better than LogitBoost-NB-Multinomial according to t-test with $p<0.05$. Furthermore, in Table III and Table IV, "*" indicates that LogitBoost-NB-Multinomial performed statistically better than the specific classifier according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results (*Draws*).

TABLE III.
COMPARING THE PROPOSED ALGORITHM WITH OTHER WELL KNOWN ALGORITHMS (I)

classifier		**oh0**	**tr23**	**re0**	**oh10**	**tr11**
LogitBoost NB-Multinomial	Acc.	88,33 *	94,60 *	84,17	79,90 *	87,68
	Feat.	400	250	350	350	100
SMO	Acc.	83,15 *	74,01 *	76,99 *	76,76 *	73,91*
	Feat.	490	330	300	350	450
C4.5	Acc.	85,54 *	92,64	78,05 *	79,52	79,95 *
	Feat.	310	120	430	300	110
RIPPER	Acc.	83,15 *	93,62	77,52 *	74,76 *	80,67 *
	Feat.	250	100	410	360	100
RBF	Acc.	84,94 *	82,84 *	64,76 *	73,42 *	82,60 *
	Feat.	150	50	50	100	50
NB Multinomial	Acc.	89,93	87,74 *	78,92 *	80,00	85,74
	Feat.	400	100	300	520	80

TABLE IV
COMPARING THE PROPOSED ALGORITHM WITH OTHER WELL KNOWN ALGORITHMS (II)

classifier		**re1**	**tr41**	**tr12**	**oh15**	**tr21**
LogitBoost-NB-Multinomial	Acc.	84,49*	94,87*	86,58*	83,57*	92,26*
	Feat.	200	300	110	350	100
SMO	Acc.	78,39*	81,43*	74,44*	80,06 *	81,54*
	Feat.	300	300	200	200	250
C4.5	Acc.	82,80	92,71	84,34	80,94*	87,5*
	Feat.	160	320	400	290	100
RIPPER	Acc.	81,83 *	91,11*	82,74*	78,97*	90,77
	Feat.	180	90	100	330	210
RBF	Acc.	66,20*	89,52*	80,19*	77,65*	80,95*
	Feat.	300	50	20	100	100
NB Multinomial	Acc.	85,33	94,07	84,02	84,55	88,39*
	Feat.	480	300	110	360	100

Thus, the proposed algorithm is significantly more accurate than single NB-Multinomial in 3 out of the 10 data sets, while it has not significantly higher error rates than NB-Multinomial in any data set. Moreover, the proposed algorithm is significantly more accurate than SMO and RBF algorithms in all data sets. LogitBoost-NB-Multinomial is significantly more accurate than C4.5 in 5 out of the 10 data sets, while it has not significantly higher error rates than C4.5 in any data set. Finally, the proposed algorithm is significantly more accurate than RIPPER in 8 out of the 10 data sets, while it has significantly higher error rates than RIPPER in none data set.

In brief, we managed to improve the performance of the Naive Bayes MultiNomial Classifier obtaining better accuracy than other well known classifiers.

VI. Conclusion

The text classification problem is an Artificial Intelligence research topic, especially given the vast number of documents available in the form of web pages and other electronic texts like emails, discussion forum postings and other electronic documents.

It has observed that even for a specified classification method, classification performances of the classifiers based on different training text corpuses are different; and in some cases such differences are quite substantial. This observation implies that a) classifier performance is relevant to its training corpus in some degree, and b) good or high quality training corpuses may derive classifiers of good performance. Unfortunately, up to now little research work in the literature has been seen on how to exploit training text corpuses to improve classifier's performance.

In this work, we managed to improve the performance of the Naive Bayes MultiNomial Classifier. We combined Naive Bayes MultiNomial with Logitboost. However, as it is well known, Logitboost requires a regression algorithm for base learner. For this reason, we slightly modify Naive Bayes MultiNomial classifier in order to run as a regression method. We performed a large-scale comparison with other a state-of-the-art algorithms on 10 standard benchmark datasets and we took better accuracy in most cases.

In future research it would be interesting to find a more sophisticated algorithm for choosing the number of equal-width intervals, for the application of NB-Multinomial to the discretized data. How many intervals should be generated? Depending on the application, the trend of the error of the class mean or median for a variable number of classes can be observed. Too few intervals would imply an easier classification problem, but put an unacceptable limit on the potential performance; too many intervals might make the classification problem too difficult.

Reuters Corpus Volume I (RCV1) is an archive of over 800,000 manually categorized newswire stories recently made available by Reuters, Ltd. for research purposes [38]. Using this collection, we can compare more extensively the proposed algorithm.

Acknowledgements

The Project is Co-Funded by the European Social Fund & National Resources - EPEAEK II.

References

[1] Y. Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(1/2):67–88, 1999.

[2] Ikonomakis, M., Kotsiantis, S. Tampakas, V., Text Classification Using Machine Learning Techniques, WSEAS Transactions on Computers, Issue 8, Volume 4, August 2005, pp. 966-974.

[3] Sebastiani F., "Machine Learning in Automated Text Categorization", ACM Computing Surveys, vol. 34 (1),2002, pp. 1-47.

[4] Mccallum, Andrew, Nigam, Kamal (1998), A Comparison of Event Models for Naive Bayes Text Classification, In AAAI-98 Workshop on Learning for Text Categorization.

[5] Han X., Zu G., Ohyama W., Wakabayashi T., Kimura F., Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination, LNCS, Volume 3309, Jan 2004, pp. 463-468

[6] Leopold, Edda & Kindermann, Jörg, "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?", Machine Learning 46, 2002, pp. 423 - 444.

[7] Brank J., Grobelnik M., Milic-Frayling N., Mladenic D., "Interaction of Feature Selection Methods and Linear Classification Models", Proc. of the 19th International Conference on Machine Learning, Australia, 2002.

[8] Madsen R. E., Sigurdsson S., Hansen L. K. and Lansen J., "Pruning the Vocabulary for Better Context Recognition", 7th International Conference on Pattern Recognition, 2004

[9] Kehagias A., Petridis V., Kaburlasos V., Fragkou P., "A Comparison of Word- and Sense-Based Text Categorization Using Several Classification Algorithms", JIIS, Volume 21, Issue 3, 2003, pp. 227-247.

[10] Forman, G., An Experimental Study of Feature Selection Metrics for Text Categorization. Journal of Machine Learning Research, 3 2003, pp. 1289-1305

[11] Soucy P. and Mineau G., "Feature Selection Strategies for Text Categorization", AI 2003, LNAI 2671, 2003, pp. 505-509

[12] Torkkola K., "Discriminative Features for Text Document Classification", Proc. International Conference on Pattern Recognition, Canada, 2002.

[13] Sousa P., Pimentao J. P., Santos B. R. and Moura-Pires F., "Feature Selection Algorithms to Improve Documents Classification Performance", LNAI 2663, 2003, pp. 288-296

[14] Montanes E., Quevedo J. R. and Diaz I., "A Wrapper Approach with Support Vector Machines for Text Categorization", LNCS 2686, 2003, pp. 230-237

[15] Novovicova J., Malik A., and Pudil P., "Feature Selection Using Improved Mutual Information for Text Classification", SSPR&SPR 2004, LNCS 3138, pp. 1010–1017, 2004

[16] Fragoudis D., Meretakis D., Likothanassis S., "Integrating Feature and Instance Selection for Text Classification", SIGKDD '02, July 23-26, 2002, Edmonton, Alberta, Canada.

[17] Zu G., Ohyama W., Wakabayashi T., Kimura F., "Accuracy improvement of automatic text classification based on feature transformation": Proc: the 2003 ACM Symposium on Document Engineering, November 20-22, 2003, pp.118-120.

[18] Ke H., Shaoping M., "Text categorization based on Concept indexing and principal component analysis", Proc. TENCON 2002 Conference on Computers, Communications, Control and Power Engineering, 2002, pp. 51-56.

[19] Qiang W., XiaoLong W., Yi G., "A Study of Semi-discrete Matrix Decomposition for LSI in Automated Text Categorization", LNCS, Volume 3248, Jan 2005, pp. 606-615.

[20] Wang Qiang, Wang XiaoLong, Guan Yi, A Study of Semi-discrete Matrix Decomposition for LSI in Automated Text Categorization, Lecture Notes in Computer Science, Volume 3248, Jan 2005, Pages 606 – 615.

[21] Ana Cardoso-Cachopo, Arlindo L. Oliveira, An Empirical Comparison of Text Categorization Methods, Lecture Notes in Computer Science, Volume 2857, Jan 2003, Pages 183 - 196

[22] Kim S. B., Rim H. C., Yook D. S. and Lim H. S., "Effective Methods for Improving Naive Bayes Text Classifiers", LNAI 2417, 2002, pp. 414-423

[23] Schneider, K., Techniques for Improving the Performance of Naive Bayes for Text Classification, LNCS, Vol. 3406, 2005, 682-693.

[24] Klopotek M. and Woch M., "Very Large Bayesian Networks in Text Classification", ICCS 2003, LNCS 2657, 2003, pp. 397-406

[25] Shanahan J. and Roma N., Improving SVM Text Classification Performance through Threshold Adjustment, LNAI 2837, 2003, 361-372

[26] D. E. Johnson, F. J. Oles, T. Zhang, T. Goetz, "A decision-tree-based symbolic rule induction system for text categorization", IBM Systems Journal, September 2002.

[27] Heui Lim, Improving kNN Based Text Classification with Well Estimated Parameters, LNCS, Vol. 3316, Oct 2004, Pages 516 - 523.

[28] Zhenya Zhang, Shuguang Zhang, Enhong Chen, Xufa Wang, Hongmei Cheng, TextCC: New Feed Forward Neural Network for Classifying Documents Instantly, Lecture Notes in Computer Science, Volume 3497, Jan 2005, Pages 232 – 237.

[29] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., "SMOTE: Synthetic Minority Over-sampling Technique," Journal of AI Research, 16 2002, pp. 321-357.

[30] Y. Yang, J. Zhang and B. Kisiel., "A scalability analysis of classifiers in text categorization", ACM SIGIR'03, 2003, pp 96-103

[31] Vinciarelli A., "Noisy Text Categorization, Pattern Recognition", 17th International Conference on (ICPR'04) , 2004, pp. 554-557

[32] Verayuth Lertnattee, Thanaruk Theeramunkong, Parallel Text Categorization for Multi-dimensional Data, Lecture Notes in Computer Science, Volume 3320, Jan 2004, Pages 38 - 41

[33] Bao Y. and Ishii N., "Combining Multiple kNN Classifiers for Text Categorization by Reducts", LNCS 2534, 2002, pp. 340-347

[34] Sung-Bae Cho, Jee-Haeng Lee, Learning Neural Network Ensemble for Practical Text Classification, Lecture Notes in Computer Science, Volume 2690, Aug 2003, Pages 1032 – 1036.

[35] Bi Y., Bell D., Wang H., Guo G., Greer K., "Combining Multiple Classifiers Using Dempster's Rule of Combination for Text Categorization", MDAI, 2004, 127-138.

[36] E. Frank, Trigg L., Holmes G. and Witten I.H. (2000), Technical Note: Naive Bayes for regression, Machine Learning, 41(1) 5-26, October.

[37] Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo, 2000.

[38] Lewis D., Yang Y., Rose T., Li F., "RCV1: A New Benchmark Collection for Text Categorization Research", Journal of Machine Learning Research 5, 2004, pp. 361-397.

Authors' information

[1] Department of Mathematics University of Patras, Greece
& Department of Computer Science & Technology, University of Peloponnese
sotos@math.upatras.gr.

[2] Department of Mathematics University of Patras, Greece
evelina@ math.upatras.gr

[3] Department of Mathematics University of Patras, Greece
pintelas@math.upatras.gr

Sotiris Kotsiantis received a diploma in mathematics, a Master and a Ph.D. degree in computer science from the University of Patras, Greece. He is an adjunct lecturer in the Department of Computer Science and Technology at the University of Peloponnese, Greece His main research interests are in the field of machine learning, data mining and knowledge representation. He has more than 60 publications to his credit in international journals and conferences.

Evelina Athanasopoulou received a diploma in mathematics, a Master in computer science from the University of Patras, Greece. Her main research interests are in the field of machine learning and text classification.

Panayiotis Pintelas is a Professor in the Department of Mathe-matics, University of Patras, Greece. His research interests are in the field of educational software and machine learning. He has more than 130 publications in international journals and conferences.