Integrating Global and Local Application of Discriminative Multinomial Bayesian Classifier for Text Classification

Emmanuel Pappas and Sotiris Kotsiantis

Abstract. The Discriminative Multinomial Naive Bayes classifier has been a center of attention in the field of text classification. In this study, we attempted to increase the prediction accuracy of the Discriminative Multinomial Naive Bayes by integrating global and local application of Discriminative Multinomial Naive Bayes classifier. We performed a large-scale comparison on benchmark datasets with other state-of-the-art algorithms and the proposed methodology gave better accuracy in most cases.

1 Introduction

Text classification has been an important application since the beginning of digital documents. Text Classification is the assignment of classifying a document under a predefined category. Sebastiani gave a nice review of text classification domain [17].

In this study, we attempted to increase the prediction accuracy of the Discriminative Multinomial Naive Bayes [19] by integrating global and local application of Discriminative Multinomial Naive Bayes classifier. Finally, we performed a large-scale comparison with other state-of-the-art algorithms on benchmark datasets and the proposed methodology had enhanced accuracy in most cases.

A brief description of data pre-processing 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

Emmanuel Pappas
Hellenic Open University, Greece
e-mail: mpappas@net314.eu

Sotiris Kotsiantis
Department of Mathematics, University of Patras, Greece
e-mail: sotos@math.upatras.gr

A. Abraham and S.M. Thampi (Eds.): Intelligent Informatics, AISC 182, pp. 49–55.
springerlink.com

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.

2 Data Preprocessing

A document is a sequence of words [2]. So each document is typically represented by an array of words. The set of all the words of a data set is called vocabulary, or feature set. Not all of the words presented in a document are useful in order to train the classifier [13]. There are worthless words such as auxiliary verbs, conjunctions and articles. These words are called stop-words. There exist many lists of such words which can be removed as a preprocess task. Stemming is another ordinary preprocessing step. A stemmer (which is an algorithm which performs stemming), removes words with the same stem and keeps the stem or the most general of them as feature [17].

An auxiliary feature engineering choice is the representation of the feature value [26]. Frequently, a Boolean indicator of whether the word took place in the document is satisfactory. Other possibilities include the count of the number of times the word is presented 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.

The aim of feature-selection methods is the reduction of the dimensionality of the data by removing features that are measured irrelevant [3]. This transformation procedure has a number of advantages, such as smaller dataset size, smaller computational requirements for the text classification algorithms and considerable shrinking of the search space. Scoring of individual words can be carried out using some measures, such as document frequency, term frequency, mutual information, information gain, odds ratio, χ^2 statistic and term strength [5], [15], [18]. What is universal to all of these feature-scoring methods is that they bring to a close by ranking the features by their independently determined scores, and then select the top scoring features. Forman presented benchmark comparison of twelve metrics on well known training sets [3]. Since there is no metric that performs constantly better than all others, researchers often combine two metrics [6].

Feature Transformation varies considerably from Feature Selection approaches, but like them its purpose is to reduce the feature set size [26]. This approach compacts the vocabulary based on feature concurrencies. Principal Component Analysis is a well known method for feature transformation [23]. In the text mining community this method has been also named Latent Semantic Indexing (LSI) [1].

3 Machine Learning Algorithms

After feature selection and transformation the documents can be without difficulty represented in a form that can be used by a ML algorithm. Many text classifiers have been proposed in the literature using different machine learning techniques such as Naive Bayes, Nearest Neighbors, and lately, Support Vector Machines. Although many approaches have been proposed, automated text classification is

still a major area of research mainly because the effectiveness of current automated text classifiers is not perfect and still needs improvement.

Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness [8]. However, its performance is often degraded because it does not model text well. Schneider addressed the problems and show that they can be resolved by some simple corrections [16]. In [25], an auxiliary feature method is proposed as an improvement to simple Bayes. It determines features by a feature selection method, and selects an auxiliary feature which can reclassify the text space aimed at the chosen features. Then the corresponding conditional probability is adjusted in order to improve classification accuracy.

Mccallum and Nigam [14] proposed the NB-Multinomial classifier with good results. Klopotek and Woch presented results of empirical evaluation of a Bayesian multinet learner based on a new method of learning very large tree-like Bayesian networks [9]. The study suggests that tree-like Bayesian networks can handle a text classification task in one hundred thousand variables with sufficient speed and accuracy.

In learning Bayesian network classifiers, parameter learning often uses Frequency Estimate (FE), which determines parameters by computing the appropriate frequencies from dataset. The major advantage of FE is its competence: it only needs to count each data point once. It is well-known that FE maximizes likelihood and therefore is a characteristic generative learning method. In [19], the authors proposed an efficient and effective discriminative parameter learning method, called Discriminative Frequency Estimate (DFE). The authors' motivation was to turn the generative parameter learning method FE into a discriminative one by injecting a discriminative element into it. DFE discriminatively computes frequencies from dataset, and then estimates parameters based on the appropriate frequencies. They named their algorithm as Discriminative Multinomial Bayesian Classifier.

Several authors have shown that support vector machines (SVM) provide a fast and effective means for learning text classifiers [7], [10], [21], [24]. The reason for that is SVM can handle exponentially many features, because it does not have to represent examples in that transformed space, the only thing that needs to be computed efficiently is the similarity of two examples.

kNN is a lazy learning method as no model needs to be built and nearly all computation takes place at the classification stage. This prohibits it from being applied to large datasets. However, k-NN has been used to text categorization since the early days of its research [4] and is one of the most effective methods on the Reuters corpus of newswire stories – a benchmark corpus in text categorization.

A problem of supervised algorithms for text classification is that they normally require high-quality training data to build an accurate classifier. Unfortunately, in many real-world applications the training sets present imbalanced class distributions. In order to deal with this problem, a number of different approaches such as sampling have been proposed [12], [20].

4 Proposed Methodology

The proposed model simple trains a Discriminative Multinomial Bayesian Classifier (DMNB) classifier during the train process. For this cause, the training time of the model is that of simple DMNB. During the classification of a test document the model calculate the probabilities each class and if the probability of the most possible class is at least two times the probability of the next possible class then the decision is that of global DMNB model. However, if the global DMNB is not so sure e.g. the probability of the most possible class is less than two times the probability of the next possible class; the model finds the k nearest neighbors using the selected distance metric and train the local simple DMNB classifier using these k instances. Finally, in this case the model averages the probabilities of global DMNB with local DMNB classifier for the classification of the testing instance. It must be mentioned that local DMNB classifier is only used for a small number of test documents and for this reason classification time is not a big problem. Generally, the proposed ensemble is described by pseudo-code in Fig 1.

Training:
Build Global DMNB in all the training set
Classification:
1. Obtain the test document
2. Calculate the probabilities of belonging the document in each class of the dataset.
3. If the probability of the most possible class is at least two times the probability of the next possible class then the decision is that of global DMNB model else
    a. Find the k(=50) nearest neighbors using the selected distance metric (Manhattan in our implementation)
    b. Using as training instances the k instances train the local DMNB classifier
    c. Aggregate the decisions of global DMNB with local DMNB classifier by averaging of the probabilities for the classification of the testing instance.

Fig. 1 Integrating Global and Local Application of Naive Bayes Classifier (IGLDMNB)

Combining instance-based learning with DMNB is inspired by improving DMNB through relaxing the conditional independence assumption using lazy learning. It is expected that there are no strong dependences within the k nearest neighbors of the test instance, although the attribute dependences might be strong in the whole dataset. Fundamentally, we are looking for a sub-space of the instance space in which the conditional independence assumption is true or almost true.

5 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.

For our experiments we used Naive Bayes Multinomial algorithm and Discriminative Multinomial Naive Bayes classifier. The Sequential Minimal Optimization (or SMO) algorithm was the representative of the Support Vector Machines in out study. It must be mentioned that we used for the algorithms the free available source code by the book [23]. In order to calculate the classifiers' accuracy, the whole training set was divided into 10 mutually exclusive and equal-sized subsets and for each subset the learner was trained on the union of all of the other subsets. Then, the average value of the 10-cross validation was calculated.

In Table 1, we present the average accuracy of each classifier. In the same tables, we also represent with "*v*" that the proposed IGLDMNB algorithm *looses* from the specific algorithm. That is, the specific algorithm performed statistically better than IGLDMNB according to t-test with $p<0.05$. Furthermore, in Table 1, "*" indicates that IGLDMNB 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 1 Comparing the proposed algorithm with other well known algorithms

Data-set	IGLDMNB	DMNB	SMO	NB-Multinomial
oh0	92.14	91.23	81.96*	89.03*
oh10	84.58	83.81	74.86*	81.24*
oh15	85.22	84.77	72.72*	83.78
re0	83.99	83.78	75.47*	80.38
re1	83.10	82.86	74.29*	83.35
tr11	89.27	86.23*	74.17*	84.79*
tr12	91.06	86.91*	74.46*	83.05*
tr21	92.52	91.93	79.46*	63.37*
tr23	93.29	91.17*	74.12*	71.55*
tr41	96.97	96.36	87.02*	94.42*

The proposed method is significantly more accurate than single NB-Multinomial in 7 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 algorithm in all data sets. Finally, the proposed method is significantly more accurate than simple DMNB [21] in 3 out of the 10 data sets, while it has not significantly higher error rates than DMNB in any data set.

In brief, we managed to improve the performance of the Discriminative Multinomial Bayesian Classifier obtaining better accuracy than other well known classifiers. We have implemented the proposed algorithm in a software tool (see Fig. 2). The tool expects the training set as an Attribute-Relation File Format. The class attribute must be in the last column. After the training of the model (from few seconds to few minutes to complete), one is able to predict the class of the new text.

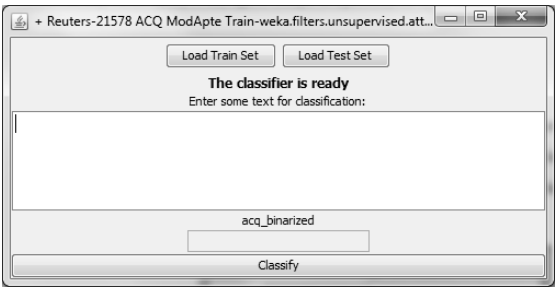

Fig. 2 A screenshot of the implemented tool

6 Conclusion

The text classification problem is a machine learning research topic, specially given the vast number of documents available in the form of web pages and other electronic texts like discussion forum postings, emails, and other electronic documents [22]. In this work, we managed to improve the performance of the Discriminative Multinomial Bayesian Classifier. 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. 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 [11]. Using this collection, we can compare more extensively the proposed algorithm.

References

1. Cardoso-Cachopo, A., Oliveira, A.L.: An Empirical Comparison of Text Categorization Methods. In: Nascimento, M.A., de Moura, E.S., Oliveira, A.L. (eds.) SPIRE 2003. LNCS, vol. 2857, pp. 183–196. Springer, Heidelberg (2003)
2. Figueiredo, F., Rocha, L., Couto, T., Salles, T., Gonçalves, M.A., Meira Jr., W.: Word co-occurrence features for text classification. Information Systems 36(5), 843–858 (2011)
3. Forman, G.: Feature selection for text classification. In: Computational Methods of Feature Selection, pp. 257–276. Chapman and Hall/CRC (2007)
4. Guo, G.D., Wang, H., Bell, D., Bi, Y.X., Greer, K.: Using kNN model for automatic text categorization. Soft Computing 10(5), 423–430 (2006)
5. Feng, G., Guo, J., Jing, B.-Y., Hao, L.: A Bayesian feature selection paradigm for text classification. Information Processing Management (2011) ISSN 0306-4573, 10.1016/j.ipm.2011.08.002
6. Chen, J., Huang, H., Tian, S., Qu, Y.: Feature selection for text classification with Naïve Bayes. Expert Systems with Applications 36(3), Part I, 5432–5435 (2009)
7. Joachims, T.: Learning to classify text using support vector machines. Kluwer Academic, Hingharn (2002)

8. Kim, S.-B., Rim, H.-C., Yook, D., Lim, H.-S.: Effective Methods for Improving Naive Bayes Text Classifiers. In: Ishizuka, M., Sattar, A. (eds.) PRICAI 2002. LNCS (LNAI), vol. 2417, pp. 414–423. Springer, Heidelberg (2002)
9. Kłopotek, M.A., Woch, M.: Very Large Bayesian Networks in Text Classification. In: Sloot, P.M.A., Abramson, D., Bogdanov, A.V., Gorbachev, Y.E., Dongarra, J., Zomaya, A.Y. (eds.) ICCS 2003. LNCS, vol. 2657, pp. 397–406. Springer, Heidelberg (2003)
10. Leopold, E., Kindermann, J.: Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? Machine Learning 46, 423–444 (2002)
11. Lewis, D., Yang, Y., Rose, T., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5, 361–397 (2004)
12. Liu, Y., Loh, H.T., Sun, A.: Imbalanced text classification: A term weighting approach. Expert Systems with Applications 36, 690–701 (2009)
13. Madsen, R.E., Sigurdsson, S., Hansen, L.K., Lansen, J.: Pruning the Vocabulary for Better Context Recognition. In: 7th International Conference on Pattern Recognition (2004)
14. Mccallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI 1998 Workshop on Learning for Text Categorization (1998)
15. Ogura, H., Amano, H., Kondo, M.: Feature selection with a measure of deviations from Poisson in text categorization. Expert Systems with Applications 36, 6826–6832 (2009)
16. Schneider, K.-M.: Techniques for Improving the Performance of Naive Bayes for Text Classification. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 682–693. Springer, Heidelberg (2005)
17. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
18. Shang, W., Huang, H., Zhu, H., Lin, Y.: A novel feature selection algorithm for text categorization. Expert Systems with Applications 33, 1–5 (2007)
19. Su, J., Zhang, H., Ling, C., Matwin, S.: Discriminative Parameter Learning for Bayesian Networks. In: ICML 2008 (2008)
20. Sun, A., Lim, E., Liu, Y.: On strategies for imbalanced text classification using SVM: A comparative study. Decision Support Systems 48(1), 191–201 (2009)
21. Vikramjit, M., Wang, C.-J., Banerjee, S.: Text classification: A least square support vector machine approach. Applied Soft Computing 7(3), 908–914 (2007)
22. Yu, B.: An evaluation of text classification methods for literary study. Literary and Linguistic Computing 23(3), 327–343 (2008)
23. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann (2011) ISBN 978-0-12-374856-0
24. Zhang, W., Yoshida, T., Tang, X.: Text classification based on multi-word with support vector machine. Knowledge-Based Systems 21(8), 879–886 (2008)
25. Zhang, W., Gao, F.: An Improvement to Naive Bayes for Text Classification. Procedia Engineering 15, 2160–2164
26. Zhang, W., Yoshida, T., Tang, X.: A comparative study of TF*IDF, LSI and multi-words for text classification. Expert Systems with Applications 38(3), 2758–2765 (2011)