Classification of Movie Reviews Using Complemented Naive Bayesian Classifier

Siva RamaKrishna Reddy V.[1], D V L N. Somayajulu[1], Ajay R. Dani[2]
NIT Warangal, India[1]
Prithvi Information Solutions Limited, India[2]

Abstract

Text classification is an important research area as it enables the computers to work intelligently process unstructured data. This unstructured data is a rich source of information for industries. Most of such opinion rich data (more than 85%) is in text format. In this work we have observed the effect of different machine learning algorithms on the text data including the Naïve Bayes. Our main focus is on improving the classification efficiency of Naïve Bayes using its complemented version with less sensitivity. The results show that the feature selection procedure from our previous work combined with these algorithms results in significant improvement of classification efficiency and reduced over-fitting compared to the previous work.

1. Introduction

In many cases our decisions are influenced by the opinions of others. Before the internet awareness became widespread, many of us used to ask our friends or neighbors for opinion of an electronic good or a movie before actually buying it. With the growing availability and popularity of opinion-rich resources such as online review websites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others.

Unfortunately, 85% of these opinion rich resources are available in unstructured format. It has encouraged the analysts to develop an intelligent system that can automatically categorize or classify these text documents. A lot of research has been carried out, and each of them belongs to one of the following two approaches:

- Unsupervised approaches manually derive or impose some rules on the data to extract useful information.
- Supervised machine learning approaches use the statistical models such as Naïve Bayes, SVM and Bayesian networks etc.,

The proposed approach in this paper is the supervised approach. Here a series of text documents or reviews which have been previously categorized (manually) and a classifier (model) is trained on these documents. Later this trained classifier is used to categorize new (unclassified) documents. The work documented here is an extension of [2]. Previously, the emphasis was on Pre-Processing i.e. converting these unstructured training documents into structural. Here it is mainly concentrated on classification.

The movie-review dataset [1] has been used for experimental purposes. A sequence of pre-processing steps will be carried out to convert these documents into structural format i.e. Term-by-Document matrix (TbyD), as most of the machine learning algorithms are designed to work on structural data rather than on the unstructured data. A set of four good classifiers including the simple and complemented Naive Bayesian, Support Vector Machines (SVM), Bayesian Networks and Discriminative Frequency Estimate with Bayesian Networks are selected for validation.

The remainder of this work is presented as follows; section 2 covers the related work done so far to the problem followed by the proposed approach where pre-processing, feature selection and classifiers employed in this work were explained briefly. In the later section 4 and 5 covers the methodology used, the results and observations respectively.

2. Related Work

This section briefly reviews the previous work related to the automated text classification, especially in the movie review domain.

Pang and Lee [10] did the foremost work of analyzing the concept of sentiment among online documents such as online reviews, feedbacks, how to classify them. They first used some manually tagged words which are good indicators for each of the classes' positive and negative classes independently and classified the documents using these selected

words as features. And later based on the statistics they extracted some more words and classified the documents using them. They came to an important conclusion that binary TbyD works better than frequency TbyD. They have also shown few good observations using bigrams (2 word combinations), Parts-Of-Speech tagging etc.

In Contrast to their previous work [10], where they used words as the features and Pang and Lee [11] proposed the following process (1) label the sentences in the document's either subjective or objective, discarding the latter; and then (2) apply a standard machine-learning classifier to the resulting extract. This can prevent the polarity classifier from considering irrelevant or even potentially misleading text.

Konig and Brill [15] studied ways to reduce the amount of human effort required to build a text classifier. In this approach, they seek to leverage human input beyond individual labels for text documents, by exploiting two salient characteristics of text classification tasks: (1) a key property of text classification is that often knowledge of only a small subset of a documents' text is sufficient to make a correct classification decision. (2) The other property is the fact that humans are typically proficient at reasoning over classification rules specified in terms of text fragments, i.e. when presented with a set of possible rules. During classification phase a hybrid approach i.e. a pattern based classifier followed by a statistical based classifier is employed.

In Rennie et al [18], the false assumptions made by the simple naïve bayes are being observed such as the imbalance in training data and the feature independency. A complemented version of the same has been proposed to tackle these two false assumptions.

Prabowo et al [17] compared all the machine learning techniques applied on the text categorization problem where they have discussed other rule-based classification models and their own hybrid model. In comparison to all these works this approach also concentrated more on the feature selection and the classification algorithm Discriminative partitioning with Bayesian Networks (DPBN) [14].

3. Proposed Approach

To address several of the typical problems in Automated Text Classification, as well as to satisfy the aims of this analysis itself, a modular system is proposed. It includes separate components for acquiring candidate keywords for texts, reducing the dimensionality of the keyword datasets, and using the resultant examples of labeled documents to classify documents.

3.1. Pre-processing and Feature Selection

Pre-processing is a very important step in text classification. This phase explains the process of converting text documents to structured format by tokenizing them into individual words along with the removal of some useless punctuation marks. Later it proceeds to the stop-word or noise words elimination and the important process of stemming.

Feature selection for text classification is a well-studied problem. Its goals are improving classification effectiveness, computational efficiency, or both [16]. Aggressive reduction of the feature space has been repeatedly shown to lead to little accuracy loss, and to a performance gain in many cases.

Here the pre-processing and feature selection are being carried as a sequence of steps described below:

- **String tokenization and punctuation removal: A**ll the words present in the text are extracted using the popular Bag-of-Words (BOW) approach. At the same time all the punctuation symbols and special characters were eliminated. The special characters '?' and '!' are not eliminated as these two characters are used to express negative feelings in evaluative texts.
- **Stop words and stemming:** After extracting all words, some of the noise words such as the prepositions, conjunctions etc., were eliminated from them. Later the remaining words are passed through a process of stemming [3] where the inflated or derived words are reduced into their respective stems or root words. *Example*: CONNECTED, CONNECTING, CONNECTION (inflected) are reduced to CONNECT (Root/Stem).
- **TbyD Construction:** Now with the remaining words a Term-by-Document (TbyD) matrix has been constructed with both the frequency and binary values separately.
- **Feature Selection:** This feature selection process is otherwise seen as the dimensionality reduction of TbyD. This has been applied in a sequence of two steps. First eliminating words by document frequency and then applying an attribute selection measure like Information Gain. Eliminating words based on document frequency, refers to the procedure that we calculate the frequency of each word in all the documents and then retain only those words which satisfy certain threshold value.
- **Information Gain (IG):** The goal of applying the IG to the set of feature vector T is to find the best subset T' which maximizes the classification efficiency [13]. Information for any attribute or feature is its measure of purity. It represents the amount of information that this feature carries and helps in classifying a new instance based on this word alone. On the other hand entropy is the measure of impurity. Information for the entire Documents *D* is calculated as follows.

$$Info(D) = -\sum_{i=1}^{m}(Pi * \log 2(Pi)) \quad (3.1)$$

Where, P_i is the probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{id}|/|D|$. *Info(D)* is just the average amount of information needed to identify the class label of a tuple in D. Now we have to calculate information of each word in D and it is calculated as follows.

$$InfoA(D) = \sum_{j=1}^{v}\left(\frac{|Dj|}{|D|} * Info(Dj)\right) \quad (3.2)$$

Where v is the different number of values word A can take. *InfoA(D)* is the expected information required to classify a tuple from D based on the partitioning by A. Smaller the expected information required, greater the purity of the partitions. Information gain is defined as the difference between the original information requirement and the new requirement and is formulated as follow.

$$Gain(A) = Info(D) - InfoA(D) \quad (3.3)$$

Gain(A) tells us how much would be gained by classifying the new instance based on A.

3.2. Machine Learning Algorithms

The goal of classification is to build a set of models that can correctly predict the class of the different objects. The input to these methods is a set of objects (i.e., training data), the classes which these objects belong to (i.e., dependent variables), and a set of variables describing different characteristics of the objects (i.e., independent variables). Once such a predictive model is built, it can be used to predict the class of the objects for which class information is not known a priori. Below is the brief overview on some classification algorithms that have been used in data mining and machine learning areas. These are used as base algorithms in our research.

Naïve Bayesian (NB) Algorithm: Naïve Bayesian algorithm [7] has been widely used for document classification, and shown to produce very good performance. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. NB algorithm computes the posterior probability that the document belongs to different classes and assigns it to the class with the highest posterior probability. The posterior probability of class is computed using Bayes rule and the testing sample is assigned to the class with the highest posterior probability. The naive part of NB algorithm is the assumption of word independence that the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category.

Imagine that each document belongs to one of the set of different classes and each document can be modeled as a set of words *Wi.*

$$P(D \mid C) = p(C) * \prod_{i=1}^{n} P(Wi \mid C) \quad (3.4)$$

Where *P(D|C)* is the prior probability of a document D for a particular class C and $P(W_i|C)$ is the joint probability of the word W_i for the class C.

Complemented Naïve Bayes (CNB): Naïve Bayes is often used as a baseline in text classification because of its speed and easiness to implement. Its severe assumptions make such efficiency possible but also adversely affect the quality of the overall classification. Even though performance of the Naïve Bayes is good it makes several poor assumptions such as data independence and the uneven training data for a particular class (skewed data). Complemented Naïve Bayes [18] is one of the Naïve Bayes variant which tackles the poor assumptions made by the parent Naïve Bayes classifier such as the Uneven Training size (The most occurring class in training data dominates during actual classification) and the independence (All features or attributes are treated individually) assumptions.

Skewed data refers to having more training examples for one class than another which causes the decision boundary weights to be biased. This in turn induces the classifier to unwittingly prefer one class over the other. To counter this problem Complement Naïve Bayes proposes a probability estimate parameter which uses data from all classes except *c* which is shown in equation (3.5). Which is more effective compared to traditional Naïve Bayes' estimate.

$$\overline{\theta c} i = \frac{N\overline{c} i + \alpha i}{N\overline{c} + \alpha} \quad (3.5)$$

Where $N\overline{c}i$ is the number of time word *i* occurred in documents in classes other than *c* and $N\overline{c}$ is the total number of word occurrences in classes other than *c*, αi and α are the smoothing parameters. This correction is related to the one-versus-all-but-one technique which is used in multi-label classification CNB performs better than one-versus-all-but-one and regular Naïve Bayes since it eliminates the biased regular Naïve Bayes weights.

Support Vector Machines (SVM): Support Vector Machine (SVM) [6] is a popular technique for classification. In recent years, the SVM has become an effective tool for pattern recognition, machine learning and data mining, because of its high generalization performance. The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes.

The foundations of SVM have been developed by Vapnik, and are gaining popularity due to many attractive features, and promising empirical performance. The formulation of SVM embodies the Structural Risk Minimization (SRM) principle, as opposed to Empirical Risk Minimization (ERM) which is commonly employed with other statistical methods. SRM minimizes the error on the training data. Thus, SVMs are known to generalize better. The SRM technique consists of finding the optimal separation surface between classes due to the identification of the most representative training samples called the support vectors. SVM attempts to position a decision boundary so that the margin between the two classes is maximized.

The idea of SRM is to find a hypothesis h which reflects the well-known trade-off between the training error and the complexity of the space. SVM learns from the training set to find a decision surface (classifier) in the vector space of data points, that best separates the data points into two classes (relevant and non-relevant).

Discriminative Partitioning using Bayesian Networks (DPBN): Discriminative Frequency Estimate (DFE) learns parameters by discriminatively computing frequencies from data. Empirical studies show that the DFE algorithm integrates the advantages of both generative and discriminative learning: it performs as well as the state-of-the-art discriminative parameter learning method ELR in accuracy, but is significantly more efficient.

DFE discriminatively computes frequencies from data, and then estimates parameters based on the appropriate frequencies. Empirical studies show that DFE inherits the advantages of both generative and discriminative learning [14]. In DFE calculation when we count an instance, we can apply the current classifier to it, and then update the corresponding entries based on how well the current classifier predicts on the instance. And given an instance e, we can compute the difference between true probability *P(c|e)* and the predicted probability $\hat{P}(c \mid e)$ generated by the current parameters, where c is the true class of e, and then updating the corresponding entries based on the difference *L(e)*. *L(e)* is the prediction loss and defined as follows.

$$L(e) = P(c \mid e) - \hat{P}(c \mid e) \quad (3.6)$$

DFE can be viewed as a discriminative version of Bayesian Networks. It is computationally efficient, converges quickly, does not suffer from the over-fitting problem, and performs competitively with the other state-of-the-art Machine learning classification algorithms.

4. Experimental Setup

According to the machine learning models discussed in 3.2, to conduct the experiments we have selected the movie review dataset [1] which consists of movie reviews collected from IMDB which are manually classified into two classes positive and negative. It has been standardized and consists of a total of 2000 documents, 1000 documents for each class. It is necessary that a machine learning algorithm must be build on a initial set of available documents called the training set and it is equally essential that the model generated must be tested on a different set of similar documents called the testing set for its prediction accuracy.

Two different kinds of evaluation schemes are followed in this work

K-fold cross validation: Here the entire data is divided into *K* equal sets and a set from them is selected alternatively to be the testing set while the remaining *(K-1)* sets are combined to be the training set. At the end the average of efficiency from the *K* iterations gives the final efficiency of the model

X:Y validation: Now the data is split into two parts not necessarily of equal size, X% of the data for training set and Y% of the data for testing set such that X+Y is 100.

The machine learning methods used in our work are the direct implementations used from WEKA. A setting different from the default settings for each classifier has been selected after carrying out experimentations on this data. Naive Bayes classifier is selected with the kernel estimator set to *true*. For DPBN number of iterations is set to *10*. LibSVM implementation of SVM is used from the WEKA with linear kernel and normalized data. And complemented naive bayes is used with a sensitivity of *0.1*.

A total of eight combinations of evaluation schemes have been followed which are shown in the results.

5. Results and Observations

Experiments conducted using the movie review dataset [1] investigated all the eight combinations of evaluation schemes with the four different classification algorithms mentioned earlier. The results were documented in two separate figures, where figure 1. shows the graphical representation of the four classifiers efficiencies using the binary Term-by-Document (TbyD) and figure 2. shows the same with frequency TbyD.

A close look at figure 1 shows that complemented naive bayes (CNB) and discriminative partitioning using bayesian networks (DPBN) are a step ahead in classification efficiencies when compared with the Support Vector Machines (SVM) and simple naive bayes (NB). When CNB and DPBN are considered

clearly the former one comes out as the winner. If 10-fold cross validation is considered as the standard then CNB tops with 94.85% accuracy followed by DPBN (93.25), NB (89.25) and SVM (88.55). The important observation from the results is that we observe the efficiencies of all the four classifiers at 20:80 and 10:90 evaluation schemes, CNB is performing well with even a less training set, the remaining three are also performing up to the mark. The main reason behind this less occurrence of over-fitting is from the efficient feature selection.

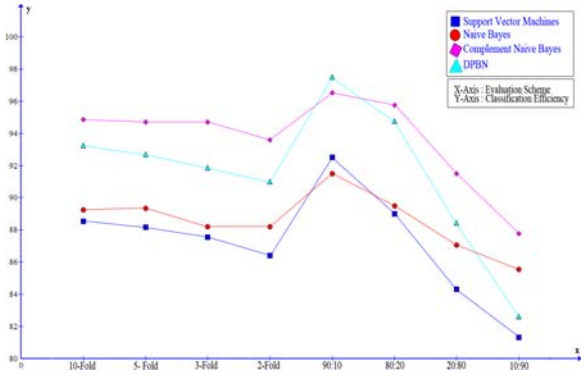

Figure 1. Graphical representation of Classifier Efficiencies for the Binary TbyD

It may be noted that Pang and Lee [10] has shown 86.4% efficiency, Konig and Brill [15] got around 91% efficiency with 90:10 evaluation scheme.

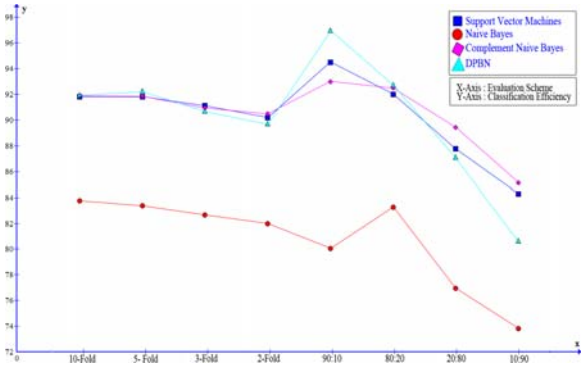

Figure 2. Graphical representation of Classifier Efficiencies for the Frequency TbyD

Now considering the frequency TbyD results from figure 2, the overall results when compared with the binary TbyD are a bit disappointing. The important observation here is that SVM joins the league of DPBN and complemented naive bayes, these three are performing almost similarly. Simple naive bayes falls flat with the frequency TbyD.

6. Conclusion

A different hybrid feature selection procedure has been employed in this work. A lot of experimentation is carried out on test data to find the better hybrid combination of pre-processing and feature selection procedures. It has been identified that SVM works very well on frequency term-by-document rather than on Binary term-by-document. Our hybrid feature selection procedure has been helpful in reducing the over-fitting problem. It is shown that Complemented Naïve Bayes classifier with low sensitivity is giving the best results at all evaluation schemes and more importantly it is giving a good efficiency at 10% and 20% documents as training sets.

So far, the work presented is on binary classification and it will be very helpful in applying this approach to the multilabel and multiclass problems.

7. Acknowledgements

We are very grateful to Kavitha Bhatt and Bala Subramanyam for their valuable inputs.

8. References

[1] Movie review data (Review Polarity Dataset Version 2.0) http://www.cs.cornell.edu/People/pabo/movie-review-data/review_polarity.tar.gz
(Access Date: 6th Feb, 2011)

[2] SRK Reddy. V; Somayajulu. DVLN; Ajay R. Dani; Sentiment Classification of Text Reviews Using Novel Feature Selection with Reduced Over-fitting, ICITST, Nov 2010.

[3] Porter, M.F. An Algorithm for Suffix Stripping, Program, 14(3): 130-137, 1980.

[4] Salton, G.; Buckley, C. Term-Weighting Approaches in Automatic Text Retrieval, Information Processing & Management Vol.24, No.5, pp. 513-523, 1988.

[5] Sebastiani, F. Machine Learning in Automated Text Categorization, ACM Computing Surveys, Vol. 34, No.1, pp. 1-47, 2002.

[6] Joachims, T. Text Categorization with Support Vector Machines: Learning with many Relevant Features, LS-8, Report 23, 1998.

[7] Pop, I. An approach of the Naïve Bayes Classifier for the document classification, General Mathematics, Vol. 14, No.4, pp. 135-138, 2006.

[8] Friedman, N. Geiger, D. Goldszmidt, M. Bayesian Network Classifiers, Machine Learning 29: pp. 131-163, 1997.

[9] Mullen, T. Collier, N. Sentiment Analysis using support vector machines with diverse information sources

in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 412–418, 2004.

[10] Pang, B. Lee, L. and Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86, 2002.

[11] Pang, B. and Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in Proceedings of the Association for Computational Linguistics (ACL), pp. 271–278, 2004.

[12] Francis, A. L. Taming Text: An Introduction to Text Mining, Casualty Actuarial Forum, and winter 2006.

[13] Coenen, F. Leng, P. Sanderson, R. Wang, Y. J. Statistical Identification of Key Phrases for Text Classification, In MLDM, LNAI 4571, pp. 838-853, 2007.

[14] Su, J. Zhang, H. Ling, C. X. Matwin, S. Discriminative Parameter Learning for Bayesian Networks, in the proceedings of 25th ICML, 2008.

[15] Konig, A. C. Brill, E. Reducing the Human Overhead in Text Categorization, KDD, 2006.

[16] Pang, B. Lee, L. Opinion Mining and Sentiment Analysis, in Foundations and Trends in Information Retrieval Vol.2, Nos. 1-2, pp. 1-135, 2008.

[17] Prabowo, R. Thelwall, M. Sentiment Analysis: A Combined Approach, Journal of Informatics, Vol. 3 No. 2, pp. 143-157, 2009.

[18] Rennie, J.D.M. Shih, L. Teevan, J. Karger, D.R. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proceedings of 20th ICML, Washington DC, 2003.

