COMPARATIVE STUDY OF TEXT CLASSIFICATION METHODS

Khalid Hussain zargar[1], Dr. Manzoor Ahmad Chachoo[2]
Department of computer science Mewar University Rajasthan
[2]Associate Professor, Department of Computer Science University of Kashmir

Abstract: The problem of text classification is the recent research area in data mining. The various applications of Text classification include information retrieval, document clustering, summarization, information extraction and so on. Machine learning techniques have been considered vital to manage the process of text categorization as vast amount of documents are in digital form and continuously increasing. In this paper we will examine various classification methods like Naïve Bayes, K- nearest neighbor and support vector machine and seek answers whether we should consider old classification algorithms or opt for new brand of support vector classification methods. Also we will seek the strengths and weaknesses of these methods on a set of binary text classification problems.
Keywords: Text classification, nearest neighbor, Naïve bayes, Support vector machines, Feature selection, precision, Recall

I. INTRODUCTION

A. Text Categorization: Since most of the text in the modern era is in digital form. The goal of text categorization is the classification of documents into predefined categories. The categories are just symbolic labels with no additional knowledge of their meanings. There are various algorithms defined for the task in the literature. The task of text categorization has the following characteristics.A set of documents are given (training set) for which the class labels are already known.The documents are preprocessed and transformed into a representation suitable for training the classifier. The tasks include tokenization, filtering, stemming etc. In tokenization a document is treated as a string, and then partitioned into a list of tokens. Removing stop words, such as "the", "a", "and", etc are frequently occurring, so the insignificant words need to be removed. After filtering the stemming algorithm is applied that converts different word form into similar canonical form. A categorization algorithm is selected, applied on the training set and a model is created to classify the test set of unlabelled documents. This process of classification is called supervised learning. In supervised learning the task of constructing a classifier depends a lot on the representation of a document. A common choice is to represent a document as a bag of words, another approach of document representation takes into account the frequency with which words appear in a specific document.

B. Feature Selection:
The main issue to tackle in text classification is the feature space. The need for feature selection to improve the learning ability of the algorithm has gained importance. The idea of Feature Selection is to select subset of features from the original documents. Not all the words presented in the document can be used to train the classifier. Feature selection is performed by keeping the words with greater information about the particular class and ignoring those with less information. The main aim of feature selection is the reduction of dimensionality of dataset by removing features that are irrelevant for classification. Another benefit of feature selection is its tendency to reduce overfitting. Various feature selection methods like chi-square test, information gain have been used for the task. In this paper we used the simple and effective term and document frequency approach where 'n' top terms with highest weights are selected.

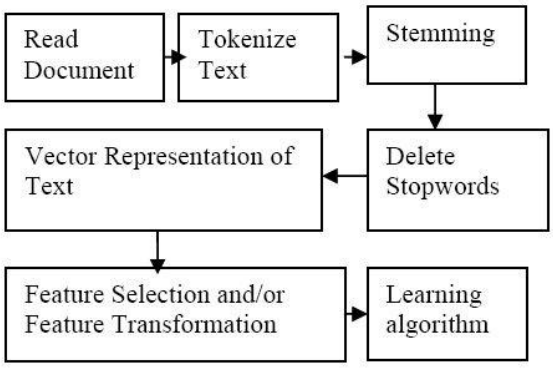


Fig. 1. Text Classification Process

C. Classification Algorithims:
Naïve Bayes Algorithim: In text classification, our goal is to find the best class for the document. The first supervised learning method we introduce is the Naïve Bayes or NB model, a probabilistic learning method. The probability of a document d being in class c is computed as

$$P(c|d) \propto P(c) \prod_{1 \le k \le nd} P(t_k|c)$$

where $P(tk|c)$ is the conditional probability of term tk occurring in a document of class c.We interpret $P(tk|c)$ as a measure of how much evidence tk contributes that c is the correct class. $P(c)$ is the prior probability of a document occurring in class c. If a document's terms do not provide clear evidence for one class versus another, we choose the one that has a higher prior probability. $(t1, t2, \ldots, tn_d)$ are the tokens in d that are part of the vocabulary we use for classification and n_d is the number of such tokens in d.
The best class in NB classification is the most likely or *maximum posteriori* (MAP) class

$$C_{map} = \underset{c \in C}{\operatorname{argmax}}\, P(c|d) = \underset{c \in C}{\operatorname{argmax}}\, P(c) \prod_{1 \le k \le nd} P(t_k|c)$$

D. KNN Classification algorithim:
Simplest of all algorithims for predicting the class of test example. K- nearest neighbor *algorithim is based on learning by analogy that is by comparing a given test example with training examples that are similar to it. All the training examples are stored in a n-dimensional space. When given an unknown example kNN searches the pattern space for the k training examples that are closest to the unknown examples. These k training examples are the k nearest neighbours. An example is classified by majority vote of its neighbours with the example being assigned to the class most common amonest its k nearest neighbours.there are two major design choices to make the value of k and the similarity function to use. The most common choice for k is a small odd integer to avoid ties. In this paper weighted term frequencies are used in the similarity function. Similarity function takes one training document and the test example as parameter. It returns a value that corresponds to the amount of similarity between these documents. The similarity function is given by*

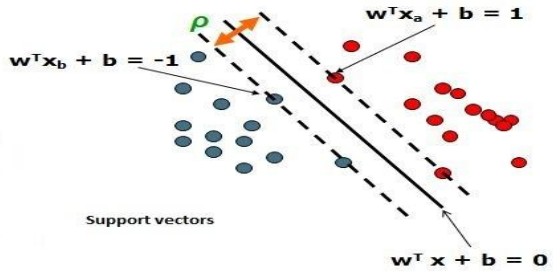
$$Sim(X,Dj)= [\sum ti \in (X \cap Dj) xi *] dij] / [||X||] * [||dj||]$$

X is a vector that keeps all terms in a new document
Dj is the training document
Ti is the term found in both documents or vectors
Xi and dij are weighted term frequencies for term I and the two documents

E. Support vector Machines:
Support vector classification proposed by Vapnik is a supervised learning technique for creating a decision function with a training dataset. SVM takes a set of input data and predicts for each given input which of the two possible classes comprises the input, making the SVM a non probabilistic binary linear classifier SVMs are a class of algorithms that combine the principles of statistical learning theory with optimisation techniques and the idea of a kernel mapping. They were introduced by Boser et al. 1992), and in their simplest version they learn a separating hyperplane between two sets of points so as to maximise the margin (distance between plane and closest point). A single SVM can only separate two classes—a positive class $L1$ (indicated by $y = +1$) and a negative class $L2$ (indicated by $y = ¡1$). Given a set of training data $\{x_1, x_2, \ldots, x_l\}$ in some space R^p and their labels $\{y_1, y_2, \ldots, y_l\}$, where $y_i \in \{-1, +1\}$, estimate a prediction function 'f' such that it can classify an unseen data point x. The SVM learning algorithm aims to find a linear function of the form

$$f(x) = W^T X + b = W .X + b.$$

A better classification performance is achieved by the hyperplane that has the largest distance to the nearest training data points of any class(margin) . The larger the margin the lower the generalization error of the classifier. The decision function is fully specified by a subset of training samples, the support vectors. Assuming that all data is atleast distance 1 vector from the hyperplane, the the following two constraints follow for a training set (xi,yi)



$W .X + b >=1$ if $y_i = 1$
$W .X + b <=1$ if $y_i = -1$

Here w is the vector perpendicular to the hyperplane an drepresents the orientation of hyperplane in d dimension space, b is the position of the hyperplane in d dimension space and x is abinary vector representing the new document to classify . Once the weights are learned, new items are classified by computing W. X

F. Experimental Data: For our experiments we use Reuters corpus volume1 dataset . Of the 91 catagories only eight top catagories are used. In these experiments we used 10 fold cross validation . The training set includes 2100 examples and a test set of 228 examples.

G. Parameter optimization: In our comparative study, we will compare classification algorithms when the feature space size and the number of training documents is varying on a large number of binary text classification methods. As some algorithms like the SVM and the k nearest neighbor classifier can accept different parameters, we decided to perform some experiments in order to limit the subsequent comparative study to a selection of parameter-optimized classifiers. To reduce the number of parameters, we make the Naive Bayes conditional independence assumption. We assume that attribute values are independent of each other given the class. the Precision/Recall-Breakeven Point is used as a measure of performance to stay (at least to some extend) compatible with previously published results. The precision/recall-breakeven point is based on the two well know statistics recall and precision widely used in information retrieval.

H. Experimental Results:
In our experiments we examine the classifier learning abilities by varying the number of training documents K-nearest neighbor and Naïve bayes tend to reach the best performance on a medium sized training document space than the SVM classifier which deals best with very large number of features. In our experiments various parameters of SVM are considered in an attempt to optimize the performance of this algorithim. The parameter C (default 200) (relative importance of complexity of model and error) and various kernel functions were tried as well. None of these lead to interesting improvements in terms of performance. The nearest neighbor classifier was tried with various values for K (1, 10, 25 ,50) but the algorithim achieved better results for k=25.

	Naïve Bayes		KNN for k=25		SVM	
Category name	**Precision(%)**	**Recall(%)**	**Precision(%)**	**Recall(%)**	**Precision(%)**	**Recall(%)**
Wheat	50.47	50.47	68.81	74.06	64.61	97.64
Acq	98.26	99.39	99.10	99.76	99.64	97.39
Cotton	21.74	12.82	31.73	37.43	75	100
Rubber	80.00	54.05	96.97	84.49	62.96	43.24
Barley	13.21	18.92	65.26	68.51	98.06	68.43
Coffee	87.39	87.39	87.61	89.19	90.24	97.42
Fuel	85.71	46.15	76.87	76.92	51.73	76.21
Corn	48.33	48.07	45.30	51.25	79.43	91.08
Overall Accuracy (mikro)	86.36% +/-1.26%		Overall accuracy 90.44% +/- 1.83(mikro)		Overall accuracy 96.83% +/- 2.04(mikro)	

Precision /Recall breakeven for the dataset Reuters corpus volume1 (top 8 catagories) for the text classification algorithims a) Naïve Bayes b) KNN c) Support vector Machines

As the table given above illustrates that Support Vector Machines outperform the KNN and Naïve Bayes algorithim. The promising performance achieved by the SVM as proposed by Vapnik et al are because of

- High dimensional input space
- Few irrelevant features
- Document vectors are sparse
- Most text categorization problems are linearly saperable

II. CONCLUSION

We have applied the different classification algorithims to the problem of document classification. These methods were evaluated individually. Since document categorization involves feature selection, we have studied the effect of varying the training document size in addition to reducing the feature space. All the classifiers performed reasonably well. In particular the performance of Naïve Bayes improves as the number of features increases. With K-nearest neighbor the value of k has the reasonable effect on the performance of the classifier. However regarding our experimental dataset (RCV1) the optimal value of k was near to the one published previously. Finally, with respect to the best feature sizes, Support vector machines best performance with all features selected.

III. RELATED WORK

Our experimental results were found almost consistent with the results published earlier. For example in [Dum98], Platt's SVM SMO algorithm was presented to outperform the naive Bayes Although Naive Bayes and the k nearest neighbors classifier are multi-class classifiers, the SVM are by default binary classifiers. Moreover, comparisons were done on multi-class classification tasks [Yan99b; Zha01] or on the averaged performance of the set of one against all classification tasks [Dum98; Zha01].Other studies found the naive Bayes to perform worse than SVM and the k nearest neighbors [Yan99b; Zha01]. Also the dataset used in these comparative studies were older versions of the datasets used by us for our experiments.

REFERENCES

[1] F. Sebastiani, "Text categorization", Alessandro Zanasi (ed.) Text Mining and its Applications, WIT Press, Southampton, UK, pp. 109-129, 2005.
[2] A. Dasgupta, P. Drineas, B. Harb, "Feature Selection Methods for Text Classification", KDD'07, ACM, 2007
[3] Susan Dumais John Platt David Heckerman, "Inductive earning Algorithms and Representations for Text Categorization", Publishedby ACM, 1998.
[4] McCallum, A. and Nigam K., "A Comparison of Event Models for Naive Bayes Text Classification". AAAI/ ICML -98 Workshop on Learning for Text Categorization
[5] Joachims, T. "Text categorization with support vector machines: learning with many relevant features". In Proceedings of ECML-98, 10th European Conference on Machine Learning (Chemnitz,DE), pp. 137–142 1998.
[6] F. Sebastiani, "Machine Learning in Automated Text Categorization", ACM 2002.
[7] Lewis, D. D. and Ringutte, M. (1994) A comparison of two learning algorithms for text categorization. In Third Annual Symp. on Document Analysis and Information Retrieval, Las Vegas, NV, pp. 81–93.
[8] Shanahan J. and Roma N., Improving SVM Text Classification Performance through Threshold Adjustment, LNAI 2837, 2003, 361- 372
[9] Y. Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(1/2):67–88, 1999
[10] Thorsten Joachims, "A Statistical Learning Model of Text Classification for Support Vector Machines".
[11] Manabu Sassano," Virtual Examples for Text Classification with Support Vector Machines". Fujitsu Laboratories Ltd.
[12] Hein Ragas Cornelis H.A. Koster, "Four text classification algorithms compared on a Dutch corpus" SIGIR 1998: 369-370 1998.
[13] Gongde Guo, Hui Wang, David Bell, Yaxin Bi and Kieran Greer, "KNN Model-Based Approach in Classification", Proc. ODBASE pp- 986 – 996, 2003
[14] Gongde Guo, Hui Wang, David Bell Yaxin Bi , and Kieran Greer ," Using kNN Model-based Approach for Automatic Text Categorization". Belfast, BT7 1NN, UK
[15] Chen donghui Liu zhijing, "A new text categorization method based on HMM and SVM", IEEE2010.
[16] S. M. Kamruzzaman, Chowdhury Mofizur Rahman: "Text Categorization using Association Rule and Naive Bayes Classifier" CoRR, 2010
[17] Yang, Y. (1997). An evaluation of statistical approaches to text categorization.Technical Report CMU-CS-97-127, Carnegie Mellon University

