An Efficient Mining Model For Enhancing Text Classification

[1]Mahesh Kadiyala, [2]G Jacob Jayaraj
[1]Dept. of Software Engineering, S V College of Engineering & Technology, Moinabad, AP, India
[2]Dept. of CSE, S V College of Engineering & Technology, Moinabad, AP, India

Abstract
Text classification is a supervised technique and it uses labeled training data to learn the classification task and then automatically classifies the test text documents using the learnt system. In information retrieval and management task categorization plays a vital role. The procedure of text categorization includes the sequence of the steps text preprocessing, feature extraction and then classification. In our task, two machine learning algorithm (Naïve Bayes and KNN) are used for categorization. This work includes some noise reduction techniques to reduce the dimensionality of feature vector that improves the efficiency of categorization. For better categorization each document had undergone by some feature weighting and reduction techniques for the enhancement of the classification task. After that, the classifier trained by self made data set and the results analyzed by Reuters corpus document set.

Keywords
Stop Word Removal, Stemming, Feature Weighting and Selection, K-NN, Naïve Bayesian

I. Introduction

In text classification, usually the dimensionality of the feature vector is huge because the input document set consist vast data and consists so many terms. To reduce this severity, in preprocessing, we remove the unnecessary words which does not support the classification task (example articles, verbs and prepositions etc.,). For text categorization (By supervised learning procedure) we assign some documents from predefined categories (for example business, health, movies etc., from Reuter's corpus). In day to day life, the digital documents in the web are increases, the number of terms (i.e., features) that are in those documents are large in number but a few are informative. It is the severe problem which degrades the efficiency in Information Retrieval (IR) procedures. This work also includes stemming process [5] which reduces the dimensionality of feature space and the stochastic dependence between terms. A better feature selection procedure reflects the effectiveness on classification and computational efficiency. The reminder of this work is organized as follows. Section II, gives some back ground about feature selection. Section III, presents feature weightening and implementation procedure of different feature selection methods and KNN and Naïve Bayes classifiers. Section IV, describes about experimental results and analysis. Finally conclusion of this work illustrated in section V.

II. Related Work

By Feature selection approach we remove the redundant and irrelevant features from the corpus, i.e., selecting a subset of features from the training set and uses that set as feature set for text classification. Some supervised feature selection approaches (IG, MI, OR, CHI, NGL, GSS) [6] were used for this task. In our work, To reduce the noisy of data with respect to term frequency (TF) [6], document frequency (DF) [6], are implemented by giving user input threshold. By selecting the most probable features from total features (by the threshold value k) analysis of results with respect to dimensionality size possible is in our work.

III. Feature Weightening

In this process, each feature (single word or term or token) is assigned with a score based on some score computing function and higher scored (weighted) terms are selected. The score computing functions includes some mathematical definitions and probabilistic approaches which are estimated by some static information in the documents across different categories. Some example notations based on probabilities are:
P(t): The probability of a document x containing term t;
P(Ci): The probability of a document x that not belonging to the category Ci;
P(t, Ci): The joint probability of a document x containing term t and belonging to the category Ci;
P(Ci/t): The probability of a document x that belonging to the category Ci under the condition that it contains term t;
P(t/Ci): The probability of a document x that containing the term t under the condition that it belongs to the category Ci;

A. Document Frequency

The number of documents in which a team occurs is DF.

$$DF = \sum_{i=1}^{m} (A_i)$$

We computed the DF for every unique tern in the training corpus and the features which have less DF than the predefined threshold were to be removed.

B.Mutual Information (MI)

The MI and IG are gives similar results for binary problems. But we implemented multi class problem solving procedure (i.e., global feature selection approach) so that these two techniques gives different results.

C. Chi Square

It measures the independence of a feature and a category. 'Null hypothesis here is that' the feature and category are completely independent, i.e., that the feature is useless for categorizing documents.

D. GSS Coefficient

It is simplified Chi Square function and named by the inventors Galvotti, Sebastiani and Simi.

$$GSS(F) = \max_{k=1}^{|c|} GSS(F, C_k)$$

E. Odds Ratio

It compares the odds of a feature occurring in one category with the odds for it occurring in another category. OR is positive score if the feature more often occur in one document than other, it is negative if the feature occurs more in the other and it is zero if the feature is equal in both i.e., ln(1)=0.

$$OR(F,C_k) = ln\frac{P(F|C_k)(1-P\left(F\middle|\overline{C}_k\right))}{P\left(F\middle|\overline{C}_k\right)(1-P(F|C_k))} = ln\frac{\left(\frac{N_{F,C_k}}{N_{C_k}}\right)(1-\frac{N_{F,\overline{C}_k}}{N_{\overline{C}_k}})}{\left(\frac{N_{F,\overline{C}_k}}{N_{\overline{C}_k}}\right)(1-\frac{N_{F,C_k}}{N_{C_k}})}$$

F. NGL Coefficient

It is a variant of the Chi Square metric, also called as correlation coefficient and named by the last names of inventors Ng, gho and Low.

G. Information Gain

This method implemented on the constraints of class membership function (presence/absence) and by that how much information gained. The IG of a term t is given as IG(t)=H©-H(C/T) where T={present, absent} and C={c+, c-}.

H. The KNN and Naïve Bayes Classifiers

The advantage of KNN in this model is, by choosing different constraints in every level of classification task we may compare the results with respect to N(N is variable) most matched values. By computing the Euclidean distance we implemented this classifier.

The Naïve Bayes classifier is both binary and multi class classifier. This classifier works on Bayes rule and works as follows. There are classes, say Ck for the data to be classified. Each class has a probability $P(C_k)$ that represents the prior probability of classifying an attribute into Ck and the values of $P(C_k)$ can be estimated from the training dataset. For n attribute values, v_j, the goal of classification is clearly to find the conditional probability $P(C_k | v_1 \wedge v_2 \wedge \ldots \wedge v_n)$. By Bayes rule, this probability is equivalent to

$$\frac{P(v_1 v_2 \ldots v_n|C_k)P(C_k)}{P(v_1 v_2 \ldots v_n)}$$

The central assumption of naïve Bayesian classification is that, within each class, the values v_i are all independent of each other. Then by the laws of independent probability,

$P(v_i | \{\text{all the other values of } v_j\}, C_k) = P(v_i | C_k)$ and therefore

$P(v_i | \{\text{all the other values of } v_j\}, C_k) = P(v_i | C_k)$ and therefore

$P(v_1 \wedge v_2 \wedge \ldots \wedge v_n | C_k) = P(v_1 | C_k)P(v_2 | C_k)\ldots P(v_n | C_k)$.

Each factor on the right-hand side of this equation can be determined from the trained data, because (for an arbitrary v_i),

$P(v_i | C_k) \approx [\#(v_i \wedge C_k)] / [\#(C_k)]$

Where '#' represents the number of such occurrences in the training set data. Therefore, the classification of the test set can now be estimated by,

$P(C_k | v_1 \wedge v_2 \wedge \ldots \wedge v_n)$ which is proportional to

$P(C_k)\ P(v_1 | C_k)\ P(v_2 | C_k)\ P(v_3 | C_k) \ldots P(v_n | C_k)$.

IV. Results and Analysis

A. Data Set1: Self Made Data Set

To train the classifiers, we used a small self-made corpus because the running time needed to be as short as possible. I collected articles in online from the New York Times, Washington post and CNN.com out of the standard category. "Business", "Science", "Sports", "Health", Education", "Movies", and "Travel". I collected 150documents with the following categories. Sports(30), Health(30), Science(27), Business(23), Education(24), Travel(6), Movies(10), with an average 702 words per document.

B. Data Set 2: The Reuters 21578 Corpus

The second corpus included in the model is the frequently used reuters21578 corpus. The corpus is freely available in the internet. By using the freely available tool SX we convert the 22 SGML documents to XML documents. We then deleted some single characters which (decimal values below 30, for example white spaces) were rejected by the validating XML parser. The results shown below in the tables are generated under the constraints where classifier is KNN, N=30, Threshold=1, Threshold step size=0.1, method of summation is sum and DF=<1, TF=<1 for the test documents Reut2-000.xml, Reut2.001.xml, Reut2.002. xml respectively. Many observations possible, for example, by table1, table2 and table3, Chi Square is much far away with respect to IG and MG values and when feature space size increases the similarity between IG and MI diminished.

Table 1: Dimensionality of Feature Space=250

	Chi Square	IG	MI
Break even	0.112	0.765	0.716
11 point	0.253	0.831	0.745
Avg precision	0.196	0.869	0.762

Table 2: Dimensionality of Feature Space=500

	Chi Square	IG	MI
Break even	0.000	0.787	0.786
11 point	0.125	0.851	0.844
Avg precision	0.137	0.891	0.872

Table 3: Dimensionality of Feature Space=1000

	Chi Square	IG	MI
Break even	0.266	0.713	0.561
11 point	0.390	0.743	0.574
Avg precision	0.345	0.77	0.588

1. KNN Classifier Performance

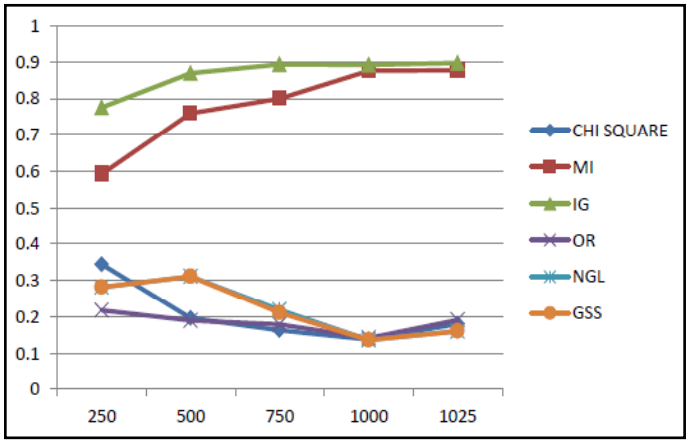


Fig. 1: KNN Classifier Results for the Test Document Reut. 002. xml at Constraints k=30, TF<1 and DF<1.

2. Naïve Bayes Classifier

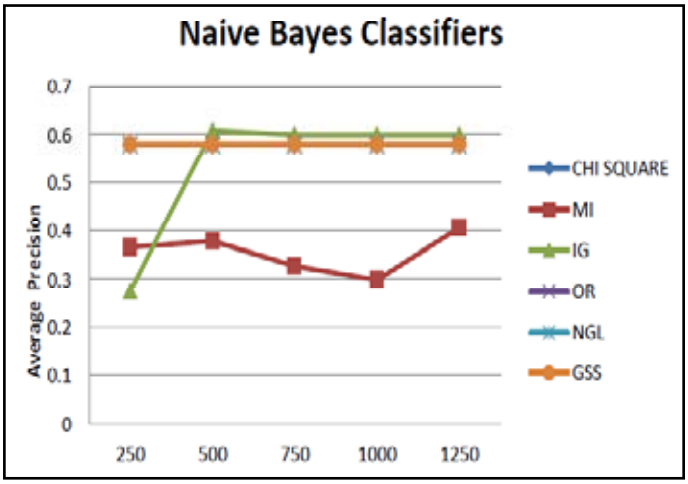


Fig. 2: Naïve Bayes Classifier Results for the Test Document Reut.002.xml at Constraints TF<1 and DF<1

Many observations for the categorization problem by two classifiers (KNN and Naïve Bayes) resulted. For most results we may conclude that Naïve Bayes classifier gives better results even though the performance of two classifiers is efficient. From fig. 1 we conclude that, KNN classifier gives different results for each feature selection technique but from fig. 2 Naïve Bayes classifier gives similar results for all feature selection techniques except for IG and MI. We observe that KNN classifier given similar results for GSS, Chi Square and NGL techniques at feature space size=1000.

V. Conclusion

We did a framework to which more classifiers (for example Neural Network classification, other Statistical methods, Case and Rule based systems) can easily be added, and to gives the users an opportunity to compare and evaluate different preprocessing techniques like stemming, term weighting, and dimensionality reduction. For a better text classification model implementation, every step of the model should yield better results. By this system we proved that Lancaster stemmer (generated unique terms 7288) better then Porter's stemmer (generated unique terms 8697), comparison of Weighting techniques Chi Square, MI and IG. From fig. 1 and fig. 2, We also proved that under some constraints Naïve Bayes classifier gives better results than KNN.

References

[1] "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification", IEEE transactions on knowledge and data engineering, Vol. 23, No. 3, march 2011.
[2] H.kim, p. Howland, H. park,"Dimension Reduction in Text Classification with Support Vector Machines", J. Machine Learning Research, Vol. 6, pp. 37-53, 2005.
[3] F.Sebastani,"Machine Learning in Automated Text Categorization", ACM Computing Surveys, Vol. 34, No. 1, pp.1-47, 2002.
[4] "The of Word clusters for Text Classification", 23rd European Colloquium on Information retrieval Research, 2001.
[5] M. F. Porter,"An algorithm for suffix stripping", 1980.
[6] Oystern Lohre Garnes, Kjetil Norvag, Robert Neumayer, "Feature Selection for Text Categorization", Norwegian University of Science and Technology, June 2009.



Mr. Mahesh Kadiyala is doing M.Tech in SV College of Engineering & Technology in the field of Software Engineering since 2010-2012 and he was completed his B.Tech in Gokula Krishna College of Engineering in the field of Information Technology since 2006-2010.



Mr. G Jacob Jayaraj received his M.Tech in Jawaharlal Nehru Technological University College of Engineering, Hyderabad in the field of Computer Science Engineering in 2004.He completed his A.M.I.E in Institute of Mechanical Engineers (Mumbai), in1998, and now he is presently working as Associate Professor (CSE Dept) in SV College of Engineering & Technology since 2006 to till date.