Categorization of Documents using ROCCHIO Feedback Algorithm with TFIDF Classifier using Entropy for Web Mining

V. V. N. A. Bhargavi

Abstract: **Text categorization is the task of assigning a given text document to one or more predefined categories. High availability of digital data and its day-to-day increase gives rise to the need of faster and better text classifiers. This paper mainly focuses on classifying data in context of text categorization. The paper reports a study conducted on 20 newsgroup dataset, using ROCCHIO feedback algorithm with TFIDF classifier in the context of document categorization. It proposes a new text categorization technique based on the ROCCHIO feedback algorithm with the prototype vectors created using TFIDF based on entropy and shows the significant increase in the accuracy with entropy based approach. Feature selection is added to this result to improvise the categorization. The results achieved using this algorithm are very promising when compared to conventional methods with features chosen on the basis of bag-of-words text. Drastic improvement was observed when subject was given considerable importance. The paper also focuses on 10-fold cross validation to show the data independent classifier accuracy. Finally it projects on a theoretical analysis of the performance comparison of various algorithms like NBC,K-NN and SVM.**

Keywords: **TFIDF, Entrop, Mutual gain,Cross Validation, ROCCHIO.**

1. INTRODUCTION

Text categorization is the key technique for handling and organizing electronic data. Document categorization is the process of grouping documents into different categories or classes. In recent years, automated classification of text into predefined categories has attracted considerable interest, due to the increasing volume of digital documents and the need to organize them. Document classification appears in many applications, including e-mail filtering, mail routing, spam filtering, search of topics from large databases, news monitoring, selective dissemination of information to information consumers, automated indexing of scientific articles, automated population of hierarchical catalogues of web resources, identification of document genre, authorship attribution, survey coding and so on. One of the main tasks of document categorization is assigning the document to a set of predefined categories which is known as Text categorization. This task involves understanding the behavior of the dataset, contents of the data set and previous understanding of the topic. It is a useful technique in information retrieval and machine learning.

In most traditional text categorization examples, bag-of-words representation of data was used for processing. Bag-of-words data is created for each document. Each bag-of-data thus created consists of vectors with real numbers. The main contributions of this paper are as follows: The paper proposes a new text categorization technique based on ROCCHIO feedback algorithm with the prototype vectors created using TFIDF based on entropy. This is referred to as TFIDF classifier based on entropy and mutual gain for each categorical document. The resulting text categorization system equally performs and outperforms traditional text categorization ROCHHIO technique, as extensively verified through experiments. Section II gives a short description of the related work which has been done on this topic. It briefly describes the overview of the tasks performed in those papers. Section III gives a brief description of the text categorization problem and introduces its working definition used throughout this paper. Section IV describes the feature extraction process from the Bag-of-words representation of a document. TFIDF classifier is described in section V. Empirical results and a theoretical comparison of various methods can be found in the sections VI and VII. The analysis of what is learned during this project can be found in sections VIII.

2. TEXT CATEGORIZATION

The problem of document categorization is to classify documents into a fixed number of known classes. The TFIDF classifier described in the paper finds an approximate class definition automatically from the training data and then classifies a document based on similarity measure between the document and the class definition. Mutual gain is calculated using entropy based similarity measure. Classification of documents is done based on TFIDF classifier combined with entropy gain measure. Representing documents as a bag-of-words is a

Assistant Professor, IT, Gokaraju Rangaraju Institute of Engineering & Technology,Hyderabad
bhargavi.varala@yahoo.com

common technique in data mining. In addition to using these words as indexing terms it is usually assumed that the ordering of the words in a document does not matter. The documents are thus no longer represented as sequences. Term frequency *TF* (word, document, frequency) and inverse document frequency *IDF* (word, document) is calculated.

The working definition of text categorization used in this paper assumes that the number of categories is fixed and known each document is assigned to exactly one of them. To put it more formally, there is a set of classes *C* and a set of training documents *D*. Furthermore, there is a mapping *T(d)* which belongs to the category *'C'* which is the true function that assigns documents to a class. *T (d)* is known for the documents in the training set. The information contained in the training examples can be used by the learning algorithm to find a model or hypothesis *H(d)* belongs to *C* which identifies or approximates *T(d). H(d)* is the class which the learned hypothesis assigns document d to and it can be used to classify new documents. The objective is to find a hypothesis which maximizes accuracy, the percentage of times a hypothesis makes a correct prediction.

3. FEATURE EXTRACTION

3.1 Representation

A standard approach to text categorization makes use of the classical text representation technique that maps a document to a high dimensional feature vector, where each entry of the vector represents the presence or absence of a feature. This approach loses all the word order information only retaining the frequency of the terms in the document. Each document can be in multiple, exactly one, or no category at all. Using machine learning, the objective is to learn classifiers which perform the category assignments automatically. The first step in text categorization is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. This leads to an attribute value representation of text. A simple way to transform a text document into a feature vector is using a "bag-of-words" representation, where each feature is a single word. Each distinct word w corresponds to a feature, with the number of times word w occurs in the document as its value. And the value of each feature is the term frequency *TF(w, d)* which is equal to the number of times the word occurred in the document. Inverse document frequency IDF is the number of documents in which the term occurs, irrespective of how many times it occurs in each document.

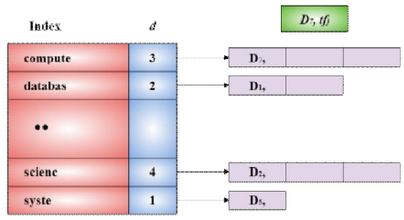

Figure 1: Term Frequency and Inverse Document Frequency

3.2 Feature Selection

Feature selection, also known as subset selection or variable selection is a process commonly used in machine learning. Feature selection has been an active research area in pattern recognition, statistics and data mining communities. The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. Feature selection can significantly improve the comprehensibility of the resulting classifier. The process of feature selection is done in 5 steps:

1. Pruning the frequent words i.e. stop words. This process involves excluding certain common words such as 'a', 'an', 'the' etc. which commonly occur in all the documents many of the times. It is important to exclude this high-frequency word because it may misclassify the documents since they exist nearly in all of the documents. All the stop words which are of length less than 3 are discarded in order to get efficient results.

2. Pruning of infrequent words. It helps removing spelling mistakes and unimportant words such as, 'aaa', '@cit', etc. All these infrequent words have been manually removed for effective computation of results. There are some infrequent words which are relevant to the categories. Removing such words is not a good idea because they may uniquely classify the documents. Hence a manual work has been done on this data to remove only irrelevant and unwanted or unimportant words.

3. Giving extra weightage to the subject of the dataset. Subject line in this dataset is grabbed. Each word in the subject line is given more importance for easy classification of the testing data. Various values are given to this subject line in order to test the efficiency of the algorithm.

4. Stemming of words is done. A stemmer is a program which determines a stem form of a given inflected word form. The stem need not be identical to the morphological root of the word, it is usually

sufficient that related words map to the same stem, even if this stem is not in itself a valid root. The proposed algorithm in this paper uses porter stemmer for stemming.

5. Choosing the words which have higher mutual information with the target concept. The mutual information of a word is calculated as the reduction in entropy achieved when the presence of an article in a category is conditioned on the occurrence of the word in the document. The formula for Mutual information of a word w with the target concept T is given in Eqn(1)

$$\begin{aligned} I(w,T) &= E(T) - E\left(\frac{T}{w}\right) \\ &= -\sum_{c \in C} P(T(d) = c) logP(T(d) = c) \\ &+ \sum_{c \in C} P(T(d) = c, w = 0) logP(T(d) \\ &= c \,/\, w = 0) \\ &+ \sum_{c \in C} P(T(d) = c, w = 1) logP(T(d) \\ &= c \,/\, w = 1 \ldots Eqn(1) \end{aligned}$$

Where *P(T(d)=c)* is the probability that an arbitrary document *d* is in category *c*. And *P(T(d)=c, w=0)* is the probability that a document *d* is in Category *c* and that it doesn't contain word *w*. *P(T(d)=c/w=0)* is the probability that a document *d* is in *C* given that it doesn't contain word *w*.

4. TFIDF CLASSIFIER

The main idea of the TF/IDF classifier is to represent each document by a vector in which each entry corresponds to a feature of the class. Then represent each class by a vector obtained by combining vectors of all the documents in the class [2].

A TFIDF classifier represents documents as vectors. Each one is characterized by a set of distinct words D_j = $(w_{j1}, w_{j2}....w_{jn})$ where *n* is the number of features of the class *j* and w_{jk}, the weight of the word *k*. W*jk* is defined in Eqn(2)

$$W_{jk} = TF(w, D) * IDF\ (k) \ \ldots Eqn(2)$$

Where *TF (w, D)* is the term frequency i.e. the number of times the word wk occurs in the topic *j*. Let *DFk* be the number of documents in which word w appears and |*D*| be the total number of documents. Inverse document frequency can be calculated using this formula.

$$IDF\ (k) = log\left(\frac{|D|}{DFk}\right) \ldots Eqn(3)$$

The weighting function assigns high values to the class related words. Conversely it will assign low weights to words appearing in many classes or those that are rare. The TFIDF measure assigns a weight to each unique word in a document representing how topic specific that word is to its document or class. The term frequency is the number of times that word appears in the class. The inverse document frequency component computes the log of ratio of the total number of classes, to, the number of classes containing word. This weighting function assigns high values to topic specific words, which are those words that appear in relatively high frequency within one category but appear in relatively few other categories. Words that occur in many classes, or that occur with low frequency, are deemed more general and are hence assigned low weights.

The proposed algorithm uses ROCCHIO relevance feedback machine learning classifier for text categorization. It uses standard TFIDF weighted vectors to represent text documents (normalized by maximum term frequency). For each category *c*, *a* prototype vector has been computed by summing the vectors of the training documents in the category. The prototype vector for a category *'c'* is calculated as shown below in Eqn(4)

$$\bar{c} = \sum_{d \in c} \bar{d} \ \ldots Eqn(4)$$

The resulting set of prototype vectors of all the classes forms the learned model or training data. Given some new document represented by weighted vector, the topic similarity between a class and the new document can be computed with the cosine measure. The similarity between a class *c* and *a* document *d* represented by a vector *d* is calculated by the following cosine similarity.

$$H(d') = \max_{c \in C} \cos(\vec{c}, \vec{d}) \ \ldots Eqn(5)$$

The class given by H(d') is the class the document d' is classified into. Since smaller the angle, larger the cosine it signifies that we are choosing the class whose vector has the smallest angle with the document vector d'. Hence the decision rule for the TFIDF classifier can be written as

$$H(d') = \max_{c \in C} \frac{\cos(\vec{c}, \vec{d})}{\|\vec{c}\|\|\vec{d}\|} \ \ldots Eqn(6)$$

5. EXPERIMENTS AND RESULTS

20 Newsgroup data set is used for experimenting the proposed algorithm. This data set consists of 20000 messages taken from 20 news groups. Following are the 10 newsgroups which are used for experimentation.

alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball

5.1 Traditional Rocchio Approach

Given data set was first implemented using traditional ROCCHIO algorithm. The data set was first stemmed and then all the stop words are removed. Traditional ROCCHIO algorithm for training data works as follows:

```
1. Assume the set of categories is {c1, c2.cn}
2. For i from 1 to n let pi = <0, 0. 0>
   (init.prototype vectors)
3. For each training example <x, c(x)>??D
   Let d be the frequency normalized TF/IDF term
   vector for doc x
   Let i = j: (cj = c(x))
   (Sum all the document vectors in ci to get
   pi)
   Let pi = pi + d(One vector per category)

For testing data,
1. Given test document x
2. Let d be the TF/IDF weighted term vector for
   x
3. Let m = .2 (init. maximum cosSim)
4. for i from 1 to n:
(compute similarity to prototype vector)
Let s = cosSim (d, pi)
if s > m
    let m = s
    let r = ci (update most similar class
    prototype)
Return class r
```

Words contained in the Subject field are given more weightage than other words which are occurring in the document. For e.g. the weight of the word in the subject field is twice the value of the normal weight of that word which is calculated using TFIDF measure. Like wise, the behavior of the test data has been analyzed giving different weights to the subject words in both training and testing data. Results show that subject words when given less value works better when compared to the weights when given with more value. Graph below shows the behavior of the algorithm when given different values to the subject.

5.2 Entropy Based Method

Given data set was then combined with mutual gain based on entropy for each category. Entropy is a measure that measures the disorder of a set of data. Entropy is a decision tree machine learning algorithm which determines which attribute is the best classifier. But in our algorithm, entropy plays a major part in classification unlike selecting a best attribute as a classifier. Entropy of a category is probability of the number of words occurring in that category given total number of words in all the categories. The two decision tree paths left and right is finding out the entropy of the word occurring in that category and entropy of the word occurring not in that category. Entropy of the word occurring in the category is the probability of number of times that word occurring in that category given total number of times all the words occurring in that category. The negation of the above statement is the entropy of the word not occurring in that category. Mutual gain is obtained by subtracting both the above entropies from the entropy of that category. This mutual gain value is multiplied with the weight of that word to get a new measure entropy weight of that word

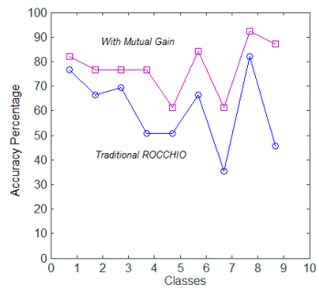

Figure 2: Traditional ROCCHIO Vs Mutual Gain

X-axis in the figure represents all the 10 categories in the order specified above. Y-axis specifies the accuracy percentage level for each category. In the above graph, pink lined graph specifies the mutual gain information method (TFIDF * information gain) proposed in the paper where as blue lined graph specifies traditional TFIDF method. The graph clearly indicates that the proposed weighted algorithm (TFIDF + mutual gain) produces better results than traditional algorithm (ROCCHIO TFIDF measure).

5.3 10-Fold Cross Validation:

Cross validation is a common approach to evaluating the fitness of a model generated via data mining, where the data is divided into a training set and a test subset to respectively build and then test the model. Common cross validation techniques in use are holdout method, k-fold cross validation and the leave-one-out method. In our validation methodology we use k-fold validation paradigm specifically 10 fold validation scheme. In 10-fold cross validation, the original data is partitioned into 10 sub samples. Of the 10 sub samples, a single sub sample is retained as the validation data for testing the model, and the remaining 9 sub samples are used as training data. The cross validation process is then repeated 10 times (the folds) with each of the 10 sub samples used exactly once as the validation data. 10 results from the folds then can be

averaged (or otherwise combined) to produce a single estimation. On average, 10-fold cross validation shows the data independent classifier accuracy. However it performed differently for few data sets. Graph below shows the mutual gain weighted measure results based on 10-fold cross validation methodology. X-axis consists of 10 fold-cross validation data sets and y-axis consists of accuracy percentage. Graph clearly indicates that the result is very good for some training and testing data while the result is medium for other sub samples of training and testing data .So the module parameter which is the subject here can be tuned to further enhance the data independent classifier accuracy. Incremental evaluation has been performed on the dataset starting from the small sized training data set to large sized training data set. Our algorithm shows a drastic improvement in the percentage of accuracy achieved when the training documents are increased from 100 to 500. The feature set considered for the comparison contained 100 features.

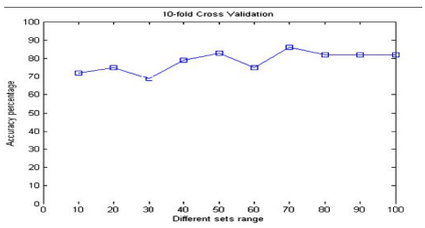

Figure3.10-fold Cross Validation

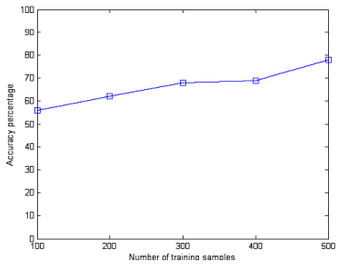

Figure 4: Increase in accuracy with training samples

The graph above shows the improvement of our approach when implemented on the larger dataset. X-axis shows the number of training samples and Y-axis shows the accuracy percentage. As the number of samples increases, our algorithm performs better and produces better results. We also compared the accuracy of our algorithm depending upon the number of features being considered. The increase in the number of the features also had a considerable impact on the accuracy of the classifier. The below graph shows the accuracy of our algorithm with the increasing values of the features:

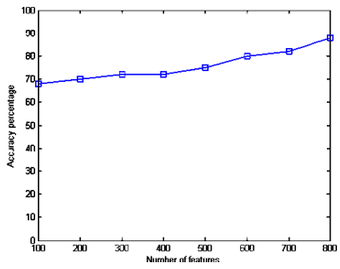

Figure5: Increasing features.

6. THEORETICAL COMPARISON OF VARIOUS ALGORITHMS

A theoretical analysis for performance comparison of various Machine Learning Methods for document categorization like Naive Bayes (NB), K-nearest neighbor(KNN) and support vector machines (SVMs) along with the TFIDF classifier is made. NBC is a probability-based approach, the basic concept of it is to find the probability that a particular word is in the particular class. If we want to apply K-NN method to classify e-mails, the emails of the training set have to be indexed and then convert them into a document vector representation. When classifying a new e-mail, the similarity between its document vector and each one in the training set has to be computed. Then, the categories of the k nearest neighbors are determined and the category which occurs most frequently is chosen. Support vector machine (SVM) has become very popular in the machine learning community because of its good generalization performance and its ability to handle high dimensional data by using kernels. An e-mail may be represented by a feature vector x that is composed of the various words from a dictionary formed by analyzing the collected e-mails. An e-mail classification can be analyzed by performing a simple dot product between the features of an email and the SVM model weight vector, $y = w.x+b$, where y is the result of classification, w is weight vector corresponding to those in the feature vector x, and b is the bias parameter in the SVM model that determined by the training process. From the theoretical analyses, the following observations are made:

1) SVMs consistently achieve good performance on text categorization tasks, outperforming existing methods substantially and significantly[3]. With their ability to generalize well in high dimensional feature spaces, SVMs eliminate the need for feature selection, making the application of text categorization considerably easier. Another advantage of SVMs over the conventional methods is their robustness. Furthermore, SVMs do not require any parameter tuning, since they can find good parameter settings

automatically. All this makes SVMs a very promising and easy-to-use method for learning text classifiers.

2) KNN performs the worst among all considered methods. However, if the more preprocessing tasks are utilized (i.e., stemming and stopping are applied together), then better KNN performs.

3) Stemming does not make any significant improvement for all algorithms in performance, though it decreases the size of the feature set. On the other hand, when the stopping procedure will be employed, that is, ignoring some words that do not carry meaning in natural language, we can get better performance. The phenomenon is obvious especially in K-NN

4) We may try to improve the accuracy result by combing some of the methods. One approach is to integrate TF-IDF and NB methods.

7. CONCLUSION AND FUTURE WORK

Text Categorization is the key technique for handling and organizing electronic data. Document categorization is the process of grouping documents into different categories or classes. Bags-of-words data representation is used for text mining generally. Feature selection is used to choose a subset of input variables by eliminating features with little or no predictive information. Proposed algorithm is the classification task which mainly uses entropy based mutual information gain when combined with TFIDF weighted approach. Precision has been increased using this method when compared to traditional ROCCHIO method which only used TFIDF classifier to classify the documents. We implemented our algorithm on three weights. One weighting measure being traditional tf-idf. Traditional ROCCHIO algorithm results dealt with that procedure. The other one is the tf-idf time's entropy of each word of a particular category. The weighing scheme is idf time's entropy of a particular word in that category. With the experiment results obtained, it shows that third measure outperforms the second measure which unusual. When we analyzed the data, we understood that this is because of the stop-words in the documents. We could manually remove many stop-words for better performance of entropy based tf-idf measure but it is a time consuming task. They are many things which we need to improvise on.

8. REFERENCES

[1]. T.Mitchell. Machine Learning, McGraw Hill, 1997.

[2]. T.Joachims (1996). A probabilistic analysis of the ROCCHIO algorithm with TFIDF for text categorization, Computer Science Technical Report CMU-CS-96- 118.Carnegie Mellon University.

[3]. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, University at Dortmund, LS VIII, 1997.

[4]. Manu Arey, Naveen Ramamurthy, Y.Alp Aslandogan. Topic Identification of Textual data.

[5]. Fei Yu, Jiyao An and Hong Li, .Intelligence Text Categorization Based on Bayes Algorithm,. Proceedings of 2004 International Conference on Information Acquision, 2004, pp. 347-350.

[6]. Qi-Rui Zhang,Ling Zhang, Shou-Bin_Dong,Jing-Hua Tan, .Document Indexing In Text Categorization,. Proceedings of 2005 International Conference on Machine learning and Cybernetics, 2005, pp. 3792-3796.