Text Document Classification based-on Least Square Support Vector Machines with Singular Value Decomposition

M.Ramakrishna Murty
Associate Professor
Department of CSE
GMRIT, Rajam,
Srikakulam(Dist) A.P. India

J.V.R Murthy
Professor
Department of CSE
JNTU Kakinada
A.P India

Prasad Reddy P.V.G.D
Professor
Dept of CS&SE
Andhara University
Visakhapatnam, A.P,India

ABSTRACT
Due to rapid growth of on-line information, text classification has become one of key technique for handling and organizing text data. One of the reasons to build taxonomy of documents is to make it easier to find relevant documents, content filtering and topic tracking.

LS-SVM is the classifier, used in this paper for efficient classification of text documents. Text data is normally high-dimensional characteristic, to reduce the high-dimensionality also possible with SVM.

In this paper we are improving classification accuracy and dimensionality reduction of a large text data by Least Square Support Vector Machines along with Singular Value Decomposition.

Keywords
Text classification, Least-Square Support Vector Machines. Singular Value Decomposition.

1. INTRODUCTION
Text classification task involves identifying the main themes of a document by placing the document into pre-defined set topics. An example would be to automatically label each incoming documents with a topic like "sports", "politics", "Education" and "Business". [11] Whatever the specific method employed, a data mining classification task starts with a *training set D* = (*d*1, . . . , *dn*) of documents that are already labelled with a class $L \in \mathbf{L}$(e.g. sport, politics). The task is then to determine a *classification model*

$$f : D \rightarrow \mathbf{L}\ f(d) = L \quad (5)$$

which is able to assign the correct class to a new document *d* of the domain.

Large collections of documents are becoming increasingly common. The first point for applying classification methods on unstructured text document collection is to create structured format called vector space model.

In the vector-space model, each document is represented as a vector within the vector space. Each dimension of the vector corresponds to one word, and the value of each component is the relative frequency of that word within the document. As a result, an m-by-n term-document matrix A is constructed, where m is the number of unique terms in the text collection and n is the number of documents in the collection. Within this matrix the term Ai,j is the frequency of the ith term within the jth document.

In the process of vector space model construction applies basic preprocessing procedure like eliminate stopwords, stemming words to extract, unique content bearing words or keywords from the set of documents.[1] All these keywords used to construct vector space model, which is conceptually represented by a vector of keywords extracted from the documents.

Vector space model contains word-by-document matrix, whose rows are words and columns are documents, with associated weights representing the importance of the keywords in the document and within the entire document collection. [15] Thus for large document collections, both row and column dimensions of the matrix are quite large as well as sparse. According to the size of the VSM the most critical problem for text classification is the high dimensionality. High dimensionality is challenging for classification of documents.

The way to reduce the high-dimensionality of the vector space model or a term-document matrix is Singular Value Decomposition (SVD), in this the term-document matrix is decomposed into three smaller matrices. Consider these three matrices for reducing high dimensionality. We take document-document matrix for further classification. Support Vector Machine is the procedure used for effective document classification used in this paper.

In this paper our proposal is to reduce the dimensionality of the term-document matrix using SVD to improve the quality of the classification.

We organize the remaining parts of this paper as follows. In section 2 we review preprocessing approach in text data analysis. In section 3 singular value decomposition approach, section 4 classification and section 5, 6 covers our approach and experimental results. Section 7 is conclusion and future work.

2. PREPROCESSING

In order to obtain all words that are used in a given text, a *tokenization* process is required, i.e. a text document is split into a stream of words by removing all punctuation marks and by replacing tabs and other non-text characters by single white spaces. This tokenized representation is then used for further processing. In order to reduce the size of the set of words describing document can be reduced by filtering and stemming.

In this section, we describe our proposed preprocessing method for creating the optimistic vector space model. Our proposed preprocessing method leads to the optimal creation of the vector space model with less time complexity.

In our preprocessing approach we collect all the stopwords, which are commonly available. Now uses the ASCII values of each letter without consider case(either lower case or upper case) and sum the each letter corresponding ASCII value for every word and generate the number. Assign number to corresponding word, and keep them in sorted order.

Suppose for example the word "and", corresponding ASCII value of a=97,n=111and d=101then the total word "and" value is 309.similarily for word "to" is 127+122=249. But in this approach there is chance that the ascii sum of the two word's values can be same as shown with the below example , the word "ask" sum value is 97+115+107=319 and the word "her" sum value is 104+101+111=319.

Solution for above mentioned problem is during the comparison we can compare with the ascii sum value and in the corresponding array we can take stopwords string. So that we can compare with the string and confirm that will be no loss of key words and also we should create a subset of strings with same ascii sum so that it is enough to compare with only that subset.

For searching of ASCII values we used for individual letters used interpolation search method to get quick corresponding value.

The above proposal incorporates that into a porter stemming algorithm for stemming that gives effective preprocessing of document. The Porter stemmer is divided into five steps, in step1 removes the i-suffixes and step 2 to 4the d-suffixes.

Composite *d*-suffixes are reduced to single *d*-suffixes one at a time. So for example if a word ends ***icational***, step 2 reduces it to ***icate*** and step 3 to ***ic***. Three steps are sufficient for this process in English. Step 5 does some tidying up.

2.1 Keyword Selection Method

After filtering and stemming there is also further decrease the number of words that should be used indexing or keyword selection procedure can be used. In this case, only the selected keywords are used to describe the documents. [1] A simple method for keyword selection is to extract keywords based on their entropy. E.g. for each word *t* in the vocabulary the entropy can be computed:

$$w(t) = 1 + \frac{1}{log_2|D|}\sum_{d \in D} p(d,t) log_2\ P(d,t) \quad (1)$$

where $P(d,t) = \frac{tf(d,t)}{\sum_{i=1}^{n} tf(d,t)}$

Equation (1) gives the entropy of the given word. Here the entropy gives a measure how well a word is suited to separate documents by keyword search. For instance, words that occur in many documents will have low entropy. The entropy can be seen as a measure of the importance of a word in the given domain context. [3] As index words a number of words that have high entropy relative to their overall frequency can be chosen, i.e. of words occurring equally often those with the higher entropy can be preferred to place in the vector space model.

2.2 Vector Space Model

Vector Space Model has been standard model of representing documents in information retrieval. The basic idea is to represent each document as a vector of certain keywords word frequencies. The vector space model represents documents as vectors in *m* dimensional space, i.e. each document *d* is described by a numerical feature vector $w(d) = (x(d, t1), \ldots, x(d, tm))$.

The main task of the vector space representation of documents is to find an appropriate encoding of the feature vector. Each element of the vector usually represents a word (or a group of words) of the document collection, i.e. the size of the vector is defined by the number of words (or groups of words) of the complete document collection. The simplest way of document encoding is to use binary term vectors, i.e. a vector element is set to one if the corresponding word is used in the document and to zero if the word is not.

Vector space model is useful because it provides an efficient, quantitative representation of each document. VSM contains the number of occurrences of word *j* in document *I*, say, f_{ji}, and the

number of documents which contain the word *j*, say, d_j. A common approach uses the solution fji X dj method

Using these counts, [9] we can represent the *i*th document as a *w*-dimensional vector x_i as follows. For $1 \leq j \leq w$, set the *j*th component x_i, to be the product of three terms

$$x_{ji} = t_{ji} . g_j . s_i \text{-- (2)}$$

where t_{ji} is the *term weighing component* and depends only on f_{ji} , while g_j is the *global weighing component* and depends on d_j, and s_i is the *normalization component* for x_i. Intuitively, t_{ji} captures the relative importance of a word in a document, while g_j captures the overall importance of a word in the entire set of documents.

The objective of such weighing schemes is to enhance discrimination between various document vectors for better retrieval effectiveness.

In this paper we use the *term frequency-inverse document frequency*. This scheme uses

$$t_{ji} = f_{ji}, \quad g_j = \log(d/d_j) \text{ and}$$

$$s_i = \sum^{w}_{j=1}(t_{ji} (t_{ji} g_j)^2)^{-½}. \text{–(3)}$$

Note that this normalization implies that $\|x_i\| = 1$, i.e., each document vector lies on the surface of the unit sphere in R^w. Intuitively; the effect of normalization is to retain only the *proportion* of words occurring in a document.

This ensures that documents dealing with the same subject matter (that is, using similar words), but differing in length lead to similar document vectors.

3. SINGULAR VALUE DECOMPOSITION

Documents are represented using the vector-space model. But we observed VSM is sparse and high dimensional. [4] The high dimensional data does not provide better results for grouping. To reduce the high dimensionality, we used Singular Value Decomposition approach.

[6] Singular Value Decomposition is a mathematical method, which says that a rectangular matrix A_{mXn}be a document-term matrix with positive real value entries. The rank reduced singular value decomposition is performed on the matrix to determine patterns in the relationship between the term and concepts contained in text.

3.1 Decomposition method using SVD

In this method the document-term matrix $A_{m x n.}$ splits into three smaller rank matrices in the following way.

Calculate U:

Step1: Calculate $A.A^T$

Step2: Eigen values of AA^T are calculate

Step3: Eigen vectors for corresponding Eigen values are calculated.

Step4: Those Eigen vectors are placed in a matrix. That is U.

Calculate V:

Step1: Calculate A^TA

Step2: Eigen values of A^TA are calculate.

Step3: Eigen vectors for corresponding Eigen values are calculated.

Step4: Those Eigen vectors are placed in a matrix. That is V

Calculate S:

Step1: Calculate square root of Eigen values of AA^T or A^TA.

Step2: Place these values as diagonal in the decreasing order. Put the remaining values as zero. This is S

[10] After apply the above calculation the document-terrn matrix A splits into three smaller rank matrices USV^T where $U^TU=I$, $V^TV=I$; the columns of U are orthonormal Eigen vectors of AA^T the columns of V are othonormal eigenvectors of A^TA, and S is a diagonal matrix containing the square roots of Eigen values from U or V descending order.

After split the term-document matrix A where U is term matrix size m x m, S is a diagonal matrix of size m x n, and V^T is a transposed matrix of V of size n x n containing in its rows the document vector as shown like Figure 1.

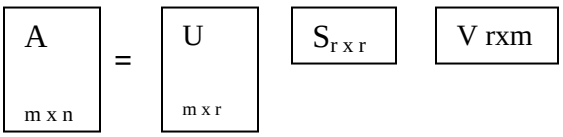

Figure 1. The decomposition of matrix A into USD^T with SVD

In the real world data is in the high-dimensional form, i.e, a document-term matrix A is having large number of rows and columns. The SVD is used to construct a low-rank matrix. In the SVD process of dimensionality reduction is makes similar items appear more similar, and unlike items more unlike.

Classification with Support Vector Machine can apply on the low dimensional data significantly reduce the computational burden.

4. CLASSIFICATION

[16] Document classification is an area that deals with the supervised grouping of text documents into meaningful groups, usually representing topics in the document collection. [7]

Document classification has many applications, such as clustering of search engine results, data analysis, e-mail filtering and mail routing etc.

Conceptually, each of the classification algorithms has a training phase, [8] in which class profiles are learned from some sufficiently large sample of documents together with their classification (the train set), and an application phase, in which the class profiles are used to assign the most probable class to new documents.

Given a set of documents D and a set of classes C, a classifier for the class c_i is a function f_i=D={0,1} that approximates an unknown function f_i:D→{0,1}which expresses the relevance of the documents for the class c_i [13] Given a set of documents classes C and example documents for each class, construct a classifier which, given a document d, finds the class(es) to which d is most similar.

4.1 Support Vector Machines

Support Vector Machine is a supervised learning technique from the field of machine learning applicable to both classification and regression. [4] The standard SVM takes a set of input data and predicts for each given input which of two possible classes.

[14] Support vector machine are learning systems that use hypothesis space of linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias determine from statistical learning theory

Given l training examples $\{x_i,y_i\}$,i=1,…l, where each example has d input($x_i \varepsilon R^d$),and a class label with one of two values ($y_i \in \{-1,1\}$). [5] Now all hyperplanes in R^d are parameterized by a vector (w) and a constant(b), expressed in the equation.

$$w.x+b=0 \text{—(4)}$$

where w and b are parameters of the model.

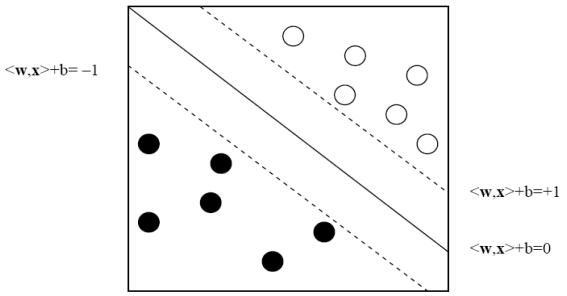

Figure2 Linearly separable classification

Figure 2 shows a two-dimensional training set consisting of dark and light squares. A decision boundary that bisects the training examples into their respective classes is illustrated with a solid line. Any example located along the decision boundary must satisfy equation (4).

[12] Let us consider a two-class classification problem the unique information concerning classes being represented by a finite sequence of labeled data,

$$S=\{ (x,y)/x_i=(x_i^{(1)},\ldots..x_i^{(d)})^T \varepsilon R^d,$$

$$y_i=\{-1,1\}, i=1,N\} \text{-----(5)}$$

the first component of each pair(x_i,y_i) of S represents an instance coming from the class of label y_i.

4.1.1 Linear separable case

The sequence is linearly separable if there exists a linear discriminant function $f:R^d \rightarrow R$ that separates the examples of S that is

$$f(x)=b+w1x^{(1)}+\ldots\ldots+wdx^{(d)} \text{ -----(6)}$$

for each $x=(x^{(1)},\ldots\ldots x^{(d)})^T \varepsilon$ R, such that for any $(x_i,y_i) \varepsilon$ S, $f(x_i)>0$ if $y_i=1$ and $f(x_i)<0$ if $y_i=1$.

The linear separability can be expressed by the existence of the parameter $b \varepsilon R$ and $w \varepsilon R^d$ such that $w^T zi+b>0$ where $Z_i=y_i x_i$ and $w=(w1,\ldots\ldots wd)^T$.In such a case we say the hyperplane

$$H_{wb}: w^T x+b=0\text{------(7)}$$

separates without errors S.

4.2 Least Square SVM

[2] Least Squares Support Vector Machines (LS-SVM) are re-formulations to the standard SVMs. The cost function is a regularized least squares function with equality constraints, leading to linear Karush Kuhn-Tucker systems. [17] The solution can be found efficiently by iterative methods like the Conjugate Gradient (CG) algorithm. LS-SVMs are closely related to regularization networks, Gaussian processes and kernel-fisher discriminant analysis, but additionally emphasize and exploit primal-dual interpretations.

Links between kernel versions of classical pattern recognition algorithms and extensions to recurrent networks and control and robust modeling are available. A Bayesian evidence framework has been applied with three levels of inference allowing for probabilistic interpretations, automatic hyper-parameter selection, model comparison and input selection.

5. OUR APPROACH

Higher dimensionality of vector space model of text data leads to computational burden and inefficient classification results. In this work we found that to reduce the high dimensionality by singular value decomposition and improve the quality of the text document grouping using Support Vector Machines.
The whole classification system is based on LS-SVM. Our approach is entirely with following steps.

A. Model the training documents into document vector

- First tokenization, In this process text documents are split into a stream of words by removing all pantuation marks and by replacing tabs and other non-text characters by single white spaces.
- Remove all stopwords which are commonly available. We followed the ASCII based approach, the ASCII value of each letter without consider case(either lower case or upper case) and sum the each letter corresponding ASCII value for every word and generate the number. Assign number to corresponding word, and keep them in sorted order.
- Use the key words and their frequencies to form a term-document matrix A.
- Impose SVD on matrix A and find a decomposed i.e reduced form of the document matrix.

B. Input the document vectors to the SVM classification system. Adjust the parameters in SVM and choose Radical Basis Function (RBF) as the kernel function of the SVM:

$$S(x, xi) = \exp\{\frac{-\|x-xi\|^2}{-2}\} \text{---(8)}$$

C. Apply the classification system formed in the above step which documents belongs to respective class.

6. EXPERIMENTAL RESULTS

In our experiment we have taken standard documents, these documents contain all together 15809 words. These words reduced after elimination of stopwords into 1840, after stemming remaining words are 323.

In this section we show the experimental results. We conducted several experiments for document preprocessing, dimensionality reduction and classification. Standard documents were used for our experiments.

After preprocessing we used Singular Value Decomposition. SVD divided the term-document matrix into three matrices. Figure 3 & 4 shows the classification results of given documents trained and classification.

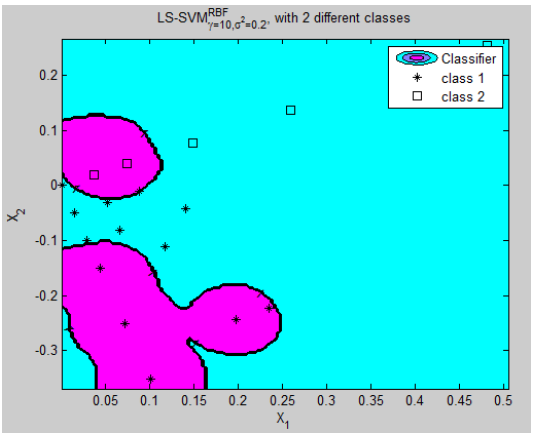

Figure 3 classification of two different documents using LS-SVM

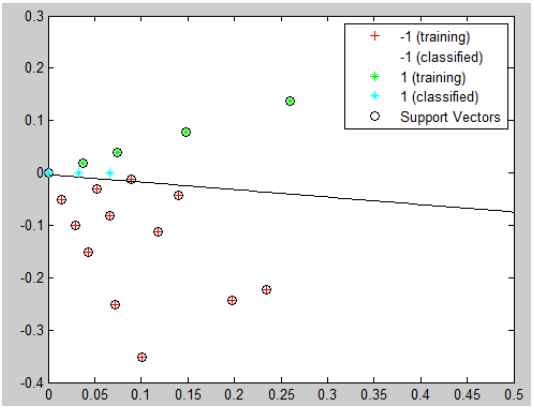

Figure 4 Trained and classification of documents using LS-SVM

7. CONCLUSION AND FUTURE WORK

In this paper, we used preprocessing method stemming with ASCII based, to eliminate the stopwords and find keywords from the verbs and nouns from the document. Finding keywords we used entropy based approach which is best to find the keywords in the input documents. Used SVD method is to reduce the dimensionality of the input term-document matrix.

This paper proposes new algorithm called LS-SVM which combines the advantages of LSI and SVM. The experiment results also confirm that LS-SVM is a very practical and effective method for classification of documents.

In future work, we will continue our focus on improving the efficiency and scalability of our preprocessing and classification schemes especially in the multiple theme documents.

8. ACKNOWLEDGMENTS

My sincere thanks to my college management; they have been providing such an environment to do research work.

9. REFERENCES

[1] Xianfei Zhang, Zhigang Guo,Bicheng Li An efficient algorithm of News Topic Tracking Global Congress on intelligent systems,2009.

[2] Suykens J.A.K., Vandewalle J., "Least squares support vector machine classi□ers," *Neural Processing Letters*, **9**(3), 293-300, 1999.

[3] Duff, R. Grimes, and J.Lewis. Sparse matrix test problems. ACM Trans Math Soft, page 1-14,1989

[4] A Practical Guide to Support Vector Classification Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin

[5] Robert Burbidge, Bernard Buxton " An Introduction to Support Vector Machines for Data Mining "

[6] Crammer, K. & Singer, Y., On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, **2**, pp. 265–292, 2001.

[7] Bloehdorn, S. & Hotho, A. (2004). Text classification by boosting weak learners based on terms and concepts. In *Proc. IEEE Int. Conf. on Data Mining (ICDM 04)*, (pp. 331–334). IEEE Computer Society Press.

[8] Noam Slonim and Naftali Tishby, " The power of word clustering for test classification", ACM SIGIR 2001.

[9] X.J, and W.Xo, Document Clustering with prior knowledge in Proc of ACM/SIGIR conference research and development Information Retrival,2006.

[10] Jaiwai, Han, Michelia Kamber Data Mining concepts and Techniques, Morgan Kaufmann publisher 2000.

[11] Manu kunchada, Text mining techniques &applications Charles river media,2006.

[12] C. Cortes and V. Vapnik. Support-vector networks .Machine Learning, 20:273-297, November 1995.

[13] Kjersti Aas and Line Eikvil TextCategorisation: ASurvey. Norwegian Computer Center Blinderri Oslo Norway,June 1999.

[14] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines. Cambridge University Press, 2000.

[15] D. S´anchez, M.J. Mart´ın-Bautista, I. Blanco C. Justicia de la Torre "Text Knowledge Mining: An Alternative to Text Data Mining" IEEE International Conference on Data Mining Workshops 2008.

[16] Vishal Gupta , Gurpreet S. Lehal "A Survey of Text Mining Techniques and Applications" Journal of Emerging technologies in web intelligence, vol,1 no1 Auguest 2009.

[17] Dustin Boswell "Introduction to Support Vector Machines" August 6, 2002