Volume 2, Issue 11, November 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Feature Selection with LSI &PDDP for Machine Learning KNN Classification

K.Gayathri,.
Dept. of. Computer Science
Nirmala College of Arts and Science
Coimbatore, India,

Dr.A.Marimuthu
Dept. of Computer Science
Government Arts College
Coimbatore, India,

Abstract: Text categorization has become one of the key techniques for handling and organizing text data. In practical text classification tasks, the ability to interpret the classification results is an important as the ability to classify exactly. This article will focus on the feature selection, for reducing the dimensionality of the vectors. We propose a new algorithm capable of partitioning a set of documents or other samples based on an embedding in a high-dimensional Euclidean space(i.e in which every document is a vector of real numbers, and then we apply classification techniques like knn for categorizations the data and finally evaluate the results by using precisions, etc.,

Keywords: Text Categorization, Pre-processing, LSI, PDDP, KNN.

I.INTRODUCTION

Text categorization is a supervised learning task in which documents are assigned to categories based on the training on a labeled document set. It has gained great popularity and importance in recent years since the amount of documents in electronic medium which necessitate organization and arrangement increased considerably. A large amount of statistical techniques and machine learning approaches have been used for this task as naïve bayes, linear regression, rocchio, neural network, k-nearest neighbor(KNN) and support vector machines[1].In text categorization, generally a document is represented as a set of words without regarding grammar and word order. This representation is called "bag of words" model. Since a document set may contain thousands of words, a bag of words representation of a document will probably have a very high dimensionality. This situation is a critical challenge for most learning algorithms. Therefore, feature selection is broadly used in text categorization systems for the purpose of reducing the dimensionality. Dimensionality reduction has many benefits such as improving the interpretability of data reducing the time and storage requirements and speeding up the learning process. Moreover it may improve the classification accuracy since it can prevent over fitting by eliminating the terms that are useless or misleading for the classifier [2].Latent Semantic Indexing (LSI) techniques compresses document vectors into vectors of a lower-dimensional space whose dimensions are obtained as combinations of the original dimensions by looking at their patterns of co-occurrence [3].We make a remark on the origin of the name "Principle Divisive Partitioning". The words "Principle Direction" are used because the algorithm is based on the computation of the leading principle direction at each stage in the partitioning. The key component in this algorithm that allows it to operate fast is a fast solver for this direction. This principal direction is used to cut a cluster of documents repeatedly. The use of a distance o similarity measure is limited to deciding which cluster should be split next, but the similarity measure is not used to do the actual splitting. We use the word "Partitioning" to reflect the fact that we place all the documents in one cluster, so that at every stage the clusters are disjoint and their equals the entire set of documents [4].This paper presents a simple KNN algorithm adapted to text categorization that does aggressive feature selection. This feature selection method allows the removal of feature that add no new information given that some other feature highly interacts with them, which weak prediction capability. Redundancy and irrelevancy could harm KNN learning by giving it some unwanted bias, and by additional complexity. By taking into account both the redundancy and relevancy of feature, we aim at providing solid ground for the use of KNN algorithms in text categorization where the document set is very large and the vocabulary diverse[5].Data sets used in the experiments are rarely same different studies usually use different portions of the test sets differently. More over classification will be performed using KNN. For the analysis and comparison of different results precision recall and F-measure are used.

II. FEATURE SELECTION

FS (Feature Selection) is an effective approach to reduce the size of feature space. Many researchers have proposed a lot such as DF (Document Frequency), IG(Information Gain), MI(Mutual Information) and combination of multiple

methods. In these methods, features are selected according to the feature function value. The features with to the higher function value will be selected firstly. In the selection process, the size of feature set plays an important role in text categorization. Experimental results show that the performance of text categorization does not always increases with the growth of features. When the features are too much the performance of those feature selection methods may be lower. Thus, a reasonable measure of feature set can not only reduce the great number of processing overhead but also improve the effectiveness of classifiers [6].

2.1 Latent semantic indexing:

Latent semantic Indexing technique compresses document vectors into vectors of a lower-dimensional space whose dimensions are obtained as combinations of the original dimensions by looking at their patterns of co-occurrence. Although it was originally applied in the context of information retrieval, since then it has been successfully applied to a wide variety of text-based tasks. These studies of LSI have mostly used standard text collections in information retrieval, some of them having simpler document models. The purpose of this paper is to investigate the use of LSI on some real-life text documents, namely patent documents. Our motivation came from the continuous growth of patent documents database in the recent years, an increase which requires the development of new and efficient methods for classification and retrieval of patent documents.

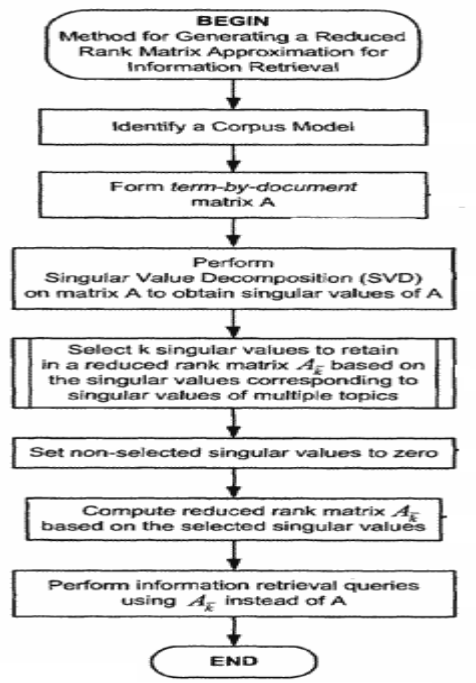

Fig .1

Latent Semantic Indexing is a statistical technique which tries to surpass some limitations imposed by the traditional Vector Space Model (VSM). In VSM, which uses the so-called bag-of-words representation of documents, the collection of text documents is represented by a terms-documents matrix

$$A = [aij] \varepsilon R^{txd}$$

where each entry aij corresponds to the number of times the term i appears in document j. Here t is the number of terms and d the number of documents in the collection. Therefore a document becomes a column vector and a query of a user can be represented as a vector of the same dimension. The similarity between the user's query vector and a document vector in the collection is measured as the cosine of the angle between the two vectors. LSI is a method for dimensionality reduction because it transforms the original terms-documents vector space into a new co-ordinate system of conceptual topics, a lower dimensional space that captures the implicit higher-order structure in the association of terms with documents . Both sets of documents and terms will be projected onto this new low-dimensional space spanned by the true factors or concepts, instead of representing documents as vectors of independent words. In order to obtain the space of concepts, i.e. the space of true representation of words and documents, LSI uses a truncated Singular Value Decomposition (SVD) applied to the terms-documents matrix A described above.

Given a t×d matrix A, where m = min(t; d), the singular value decomposition of A is defined as $A = USV^{T,}$

where U is a t_m orthonormal matrix ($U^TU = Im$), whose columns define the left singular vectors, V is a d × m orthonormal matrix ($V^TV = I_m$), whose columns define the right singular vectors and S is a m× m diagonal matrix containing the singular values of A decreasingly ordered along its diagonal: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_m = 0$, where r = rank(A). This decomposition is unique up to making the same permutations of columns of U, elements of S and columns of V(rows ofV^T).

To reduce the noise and redundancy, LSI, taking as input the terms-documents matrix described above, uses a truncation of SVD which consists in retaining only the largest k singular values and deleting the remaining ones which are smaller and thus considered unimportant. We remove also from U and V the columns corresponding to the small singular values and we get

$$A_k = U_k S_k V_k^T$$

where S_k is a k × k diagonal matrix containing the largest k singular values as entries, U_k is a t×k matrix of the corresponding left singular vectors as columns and V_k is a d × k matrix whose columns are the corresponding right singular vectors[7].

III.PRINCIPLE COMPONENT DIVISIVE PARTITIONING ALGORITHM

It operates on a sample space of m-samples in which each sample is an n-vector containing a numerical value. For clarity of presentation we will use the term "document" to refer to a sample, but in fact this algorithm can in principle be applied to samples in other domains for which the appropriate scaling may differ. Each document is represented by a column vector, of attributes values, which in the case of actual text documents are word counts. In our experiments using web documents we normalized each document vector to have a Euclidean length of one. For the purpose of our algorithm, the entire set of documents is represented by n x m matrix M=(d_1,….d_m)whose ith column, di, is the column vector representing the ith document.

The algorithm proceeds by separating the entire set of documents into two partitions by using principal directions in a way that we will describe. Each of the two partitions will be separated into two sub partitions using the same process recursively. The result is a hierarchical structure of partitions arranged into a binary tree(the "PDDP tree")_ in which each partition is either a leaf node (meaning it has not been separated) or has been separated into two sub partitions forming its two children in the PDDP tree. The details of the algorithm we must specify are (1) what method is used to split a partition into two sub partitions, and (2) in what order the partitions selected to be split. The rest of this section is devoted to filling in this details.

3.1. Splitting a Partition

A partition of p documents is represented by an n x p matrix $\mathbf{M}p = (\mathbf{d}_1,\ldots\mathbf{d}_p)$ where each d_i is an n-vector representing a document. The matrix $\mathbf{M}_p$ is a sub-matrix of the original matrix M consisting of some selection of p columns of **M**. The principle directions of the matrix M_p are the eigenvectors of its sample covariance matrix C. Let $\mathbf{W}=\mathbf{M}_p\ \mathbf{e}/p$ the sample mean of the documents $d_1,\ldots d_p$, and the covariance matrix is $c=(\mathbf{M}_p - \mathbf{we}^T)\ (\mathbf{M}_p - \mathbf{we}^T)^T$. The Karhunen-Loeve transformation consists of projecting the columns $\mathbf{M}_p$ onto the space spanned by the leading k eigenvectors. The result is a representation of the original data in k degrees of freedom instead of the original n. besides reducing the dimensionality, the transformation often has another beneficial effect of removing much noise present in the data, assuming an appropriate value of k can be chosen. In our case, we are interested in temporarily projecting each document onto the single leading eigenvector, which we will denote **u**. this leading eigenvector is chosen because it is the direction of maximum variance and hence is the direction in which the documents tend to be the most spread out.

The projection of the ith document $\mathbf{d}_i$ is given by the formula

$$\sigma\upsilon i = \mathbf{U}^T(\mathbf{d}_i - \mathbf{W})$$

where σ is a positive constant arising from the specific algorithm we use. In words, we translate all the documents so that their mean is at the origin, and then project the result onto the principal direction. The values $\upsilon_1\ldots\ldots\upsilon_k$ are used to determine the splitting for the clustering M_p. In the simplest version of the algorithm, we split the documents strictly according to the sign of the corresponding υ_i 's. All the documents $\mathbf{d}_i$ for which $\upsilon_i \leq 0$ are partitioned into the left child, and all the documents $\mathbf{d}_i$ for which $\upsilon_i > 0$ are put into the right child. A document coinciding exactly with the mean vector **w** is more or less in the middle of the entire cloud of documents. In this case, we make the arbitrary choice of documents into the left child.

3.2. Computational considerations

In the computation of the splitting of the partition, the single most expensive part of the computation is the computation of the eigen values and eigenvectors of the covariance matrix **C.** when **C** has large dimensionality this can be a significant expense. Standard off-the-shelf methods typically compute all the eigen values and eigenvectors. Even if special methods that compute only selected eigen vector are used, many methods still require the computation of all the eigen values. In addition, the covariance matrix is actually the product of a matrix and its transpose. This is exactly the situation where it is well known that accuracy can be improved by using the Singular Value Decomposition (SVD). The SVD of an n x m matrix A is the decomposition $\mathbf{U}\sum\mathbf{V}^T = \mathbf{A}$, where U,V are square orthogonal matrices of dimension nxn, mxm, respectively, and $\sum = \text{diag}\{\sigma_1,\ldots\ldots\ldots \sigma_{\min(m,n)}\}$, where $\sigma_1 \geq \sigma_2 \geq \ldots\ldots\ldots \geq \sigma_{\min\, m,n} \geq 0$. The σ,s are called the singular values, the columns of U are called the left singular vectors, and the columns of V are called the right singular vectors.

An orthogonal matrix U is a square matrix satisfying

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}$$

Setting $A=M_p - W_e^T$, we can show how to relate the SVD to the principle directions. We have

$$\mathbf{C} = \mathbf{AA}^T = (\mathbf{U}\sum\mathbf{V}^T).(\mathbf{U}\sum^T\mathbf{V}^T) = \mathbf{U}\sum^2\mathbf{V}^T,$$

Where we use $\sum^2$ as shorthand for the n × n diagonal matrix diag$\{\sigma_1^2, \sigma_2^2, \ldots\}$. From this formula, it is seen that the eigenvalues of **C** are the squares of the singular values of **A,** and the eigenvectors of C are the left singular vectors of **A**[8].

IV. K-NEAREST NEIGHBOR CLASSIFICATION APPROACH:

There are many approaches to assign category to incoming text. In our paper, we implemented text-to-text comparison (TTC), which is also known as the k-nearest neighbor (KNN) KNN is a statistical classification approach, which has been intensively studied in pattern recognition over four decades. KNN has been successfully applied to TC problem, The KNN algorithm is quite simple: Given training and test documents, the algorithm finds the k-nearest neighbors among the training documents, and uses the categories of the k-neighbors to weight the category of the test document. The similarity scores of each neighbor document to the test document are used as a weight of the categories of the neighbor n document. If several of the k-nearest-neighbors share a category, then the pre-neighbor weights of that category are added together, and the resulting weighted sum is used as the likelihood score of that category with respect to the test document. By sorting the scores of the candidates' categories, a ranked list is obtained for the test document [9].

Table-1

1	Acq	Mergers/Acquisitions
2	Grain	Grain(Commodity)
3	Ship	Shipping
4	Crude	Crude oil
5	Wheat	Wheat (Commodity)
6	Earn	Earning and earning for cast
7	Money-fx	Money/Foreign Exchange
8	Trade	Trade
9	Interest	Interest Rates
10	Corn	Com (Commodity)

V. Reuters 21578

The Reuters-21578 dataset and used the standard "modApté" train/test split. These documents appeared on the Reuters newswire in 1987 and were manually classified by personnel from Reuters Ltd. ModApt'e spli-9603 training and 3299 testing documents. Totoal-12902 documents. Out of 135 categories of documents only the top five were selected. Distinct words- 31715; Average number of words per document-126 words of which 70 were distinct.

VI. PERFORMANCE METRIC

The evaluation of a classifier is done using the precision and recall measures .To derive a robust measure of the effectiveness of the classifier It is able to calculate the breakeven point, the 11-point precision and "average precision" . to evaluate the classification for a threshold ranging from 0 (recall = 1) up to a value where the precision value equals 1 and the recall value equals 0, incrementing the threshold with a given threshold step size. The breakeven point is the point where recall meets precision and the eleven point precision is the averaged value for the precision at the points where recall equals the eleven values 0.0, 0.1, 0.2... 0.9, 1.0. "Average precision" refines the eleven point precision, as it approximates the area"below" the precision/recall curve.

Table-2

Category	**#Train Docs**	**# Test Docs**
Acq	1651	719
Grain	434	149
Ship	198	89
Crude	389	189
Wheat	212	71
Earn	2877	1087
Money-fx	538	179
Trade	369	117
Interest	348	131
Corn	181	56

Resultant Table:

Without Feature Selection, without clustering		
Accuracy	Error Rate	Speed
90.04	9.96	40.43
93.94	6.06	29.54
95.91	4.09	20.19
With Feature Selection, with clustering		
93.94	6.06	29.54
95.91	4.09	20.19
98.05	9.99	42.45

VII. CONCLUSION

Analyzed the text classification using the KNN with the Feature selection(LSI) and PDDP with KNN classification techniques. . The advantage of the proposed approach is, the classification algorithm learns importance of attributes and utilizes them in the similarity measure. In future the classification model can be build, which analyzes terms on the concept sentence in document.

REFERENCE

1. F.Sebastiani, "Machine learning in automated text categorization" ACM Computing Surveys, 34(1):1-47, 2002.
2. Serafettin Tascr, Tunga Gungor, "LDA-based keyword selection in Text Categorization", Bogazici University, Turkey.
3. Fabrizio sebastion "Machine learning in automated text categorization" ACM computing surveys Vol.34, No.1 March-2002 p.p1-47.
4. Koteeswaran.S, P.Visu and J.Janet."A review on clustering and outlier analysis Techniques in Data Mining. American Journal of Applied Sciences 9 (2):254-258-2012.ISSN:1546-9239©2012 Science Publications.
5. Pascal Soucy, Guy W.Mineau "A simple KNN Algorithm for text Categorization" Universite Laval,Quebec, Canada, 0-7695-1119-8/01©2001IEEE.
6. JinDai,Zhongshi He, and Feng Hu."A High Performance Algorithm for Text Feature Automatic Selection" Proceedings of International Symposium on Information Processing, August 21-23, 2009.
7. Andress Moldovan, Radu Ioan Bot, Gert wanka,."Latent Semantic Indexing for Patent Documents".Faculty of Mathematics, Chemnitz University of Technology, D-09107 Chemnitz,Germany.
8. Daniel Boley, "Principal Direction Divisive Partitioning", Department of Computer Science and Engineering,University of Minnesota, Minneapolis, USA.
9. Wael Musa Hadi, Fadi Thabtah, Hussein Abdel-jaber "A Comparative Study using VSM with KNN on text CategorizationData" Proceedings of the world congress on Engineering 2007 Vol-I WCE2007, July2-4-2007,London, U.K.

