Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 695720, 10 pages
http://dx.doi.org/10.1155/2015/695720

Research Article

Distance Variance Score: An Efficient Feature Selection Method in Text Classification

Heyong Wang and Ming Hong

Department of E-Business, South China University of Technology, Guangzhou 510006, China

Correspondence should be addressed to Heyong Wang; wanghey@scut.edu.cn

Received 21 January 2015; Revised 30 March 2015; Accepted 15 April 2015

Academic Editor: Alexander Klimenko

Copyright © 2015 H. Wang and M. Hong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of web applications such as social network, a large amount of electric text data is accumulated and available on the Internet, which causes increasing interests in text mining. Text classification is one of the most important subfields of text mining. In fact, text documents are often represented as a high-dimensional sparse document term matrix (DTM) before classification. Feature selection is essential and vital for text classification due to high dimensionality and sparsity of DTM. An efficient feature selection method is capable of both reducing dimensions of DTM and selecting discriminative features for text classification. Laplacian Score (LS) is one of the unsupervised feature selection methods and it has been successfully used in areas such as face recognition. However, LS is unable to select discriminative features for text classification and to effectively reduce the sparsity of DTM. To improve it, this paper proposes an unsupervised feature selection method named Distance Variance Score (DVS). DVS uses feature distance contribution (a ratio) to rank the importance of features for text documents so as to select discriminative features. Experimental results indicate that DVS is able to select discriminative features and reduce the sparsity of DTM. Thus, it is much more efficient than LS.

1. Introduction

Text classification is one of the important subfields of text mining, which recently gains increasing attention with the rapid development of web applications, such as social network. The problem of text classification can be described as follows. In practice, n text documents are often represented as a document term matrix DTM $= [D_1, D_2, \ldots, D_n]^T$ (shown in (1)). Row vector $D_i = [f_{i1}, f_{i2}, \ldots, f_{im}]$ $(i = 1, 2, \ldots, n)$ denotes the ith text document among the n text documents. Each feature f_j $(j = 1, 2, \ldots, m)$ corresponds to a term in the n text documents and f_{ij} is term frequency of f_j in D_i. Consider a simple example of two short text documents. The first document is "*You are beautiful!*" and the second is "*Good morning, you guys!*". DTM of two documents is shown in (2). $D_1 = [1, 1, 0, 0, 0, 1]$ and $D_1[1] = f_{11} = 1$ show that the first feature "*are*" exists once in the first text document ("*You are beautiful!*"). Moreover, D_i is labeled with a class c in which D_i should be. c is drawn from a predefined t-class label set $C = [c_1, c_2, \ldots, c_t]$. It is the major task of text classification to construct classification models using training text documents and then to predict a class label c $(c \in C)$ for other text documents whose class labels are unknown.

DTM of n Documents

$$\begin{array}{cccccc} & f_1 & f_2 & \cdots & f_m \\ D_1 & f_{11} & f_{12} & \cdots & f_{1m} \\ D_2 & f_{21} & f_{22} & \cdots & f_{2m} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ D_n & f_{n1} & f_{n2} & \cdots & f_{nm} \end{array} \tag{1}$$

DTM of Two Documents

$$\begin{array}{cccccc} & \text{are}\,(f_1) & \text{beautiful}\,(f_2) & \text{good}\,(f_3) & \text{guys}\,(f_4) & \text{morning}\,(f_5) & \text{you}\,(f_6) \\ D_1 & 1 & 1 & 0 & 0 & 0 & 1 \\ D_2 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \tag{2}$$

The most important characteristics of text data are its high dimensionality and sparsity [1]. A corpus of text documents often contains a large amount of words (high dimensionality) and a given text document may contain only a small number of words (sparsity). This idea is described in the aspect of DTM as follows. Consider DTM in (1), In most cases, m is very large, which makes DTM high-dimensional. For D_i, a large number of $f_{ji} = 0$ because D_i only contains a small number of the total m terms. This makes DTM high-sparse. Due to the high dimensionality and sparsity of text data, it is essential to find an efficient feature selection method to reduce feature space. The reduction of feature space causes the loss of information for text classification. A feature selection method is efficient if it keeps classification accuracy of classifiers when feature space is reduced.

Feature selection is one of the important tasks in text classification due to the high dimensionality of feature space and the existence of indiscriminative features [1]. Feature selection methods are able to reduce the high-dimensional indiscriminative feature space into low-dimensional discriminative feature subspace [2]. For example, in (2), feature "*beautiful*" and feature "*morning*" may be discriminative features and then a six-dimensional DTM can be reduced into a two-dimensional one. This paper focuses on filter feature selection and all the methods introduced in this paper are filter feature selection methods. Filter methods are chosen because they are more computationally efficient than wrapper methods [2]. A lot of feature selection methods have been proposed. Supervised feature selection methods include information gain (IG), mutual information, Gini index (GI), and expected cross-entropy. Unsupervised feature selection methods include document frequency (DF), variance score (VS) (or term variance), and LS.

LS is an unsupervised feature selection method. It is the key assumption that data from the same class label are often close to each other and the importance of a feature is evaluated by its power of locality preserving ability [3]. More details of LS are covered in Section 2.2.3. LS has been successfully used in areas such as face recognition. LS has not been used in feature selection for text classification currently. We are interested in the efficiency of this proved-discriminative feature selection method in text classification. However, both experiments and analysis indicate that LS is inefficient for text classification due to the sparsity of DTM. LS is unable to keep accuracy of classifiers when feature space is reduced and is unable to efficiently reduce the sparsity of DTM. To improve LS, this paper proposes an unsupervised feature selection method DVS. DVS uses feature distance contribution (a ratio) to replace similarity measure used in original LS to rank feature importance so as to select discriminative features. DVS is efficient in selecting discriminative features from DTM because feature distance contribution pays much attention to nonzero values and is able to avoid negative effects caused by 0 values. DVS is also able to reduce the sparcity of DTM more efficiently than LS. More details of DVS are covered in Section 3.

The rest of this paper is organized as follows. Section 2 introduces current feature selection methods and gives details of LS. Section 3 formally introduces DVS and analyzes details of DVS. In Section 4, experiments are performed using two DTM from UCI repository to validate efficiency of DVS on DTM. More discussion between efficiency of DVS and LS in text classification are in Section 4.2.3. Finally, Section 5 gives the conclusion.

2. Related Work

This section discusses several supervised and unsupervised feature selection methods proposed in current studies.

2.1. Supervised Feature Selection Methods. For DTM in (1), $P(c_k)$ denotes the prior probability that a text document D_i belongs to class c_k. $P(f_j)$ denotes the prior probability that $f_{ij} > 0$. $P(c_k \mid f_j)$ denotes the prior probability of c_k such that D_i belongs to c_k and in D_i, $f_{ij} > 0$ ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$).

2.1.1. Information Gain (IG). IG selects features according to amounts of information that a feature uses for classification [4–6]. IG of f_j in text classification is given by the following formula:

$$\begin{aligned} \mathrm{IG}\left(f_j\right) = &-\sum_{k=1}^{t} P\left(c_k\right) \log\left(P\left(c_k\right)\right) + P\left(f_j\right) \\ &\cdot \sum_{k=1}^{t} P\left(c_k \mid f_j\right) \log P\left(c_k \mid f_j\right) + \left(1 - P\left(f_j\right)\right) \\ &\cdot \sum_{k=1}^{t} \left(1 - P\left(c_k \mid f_j\right)\right) \log\left(1 - P\left(c_k \mid f_j\right)\right). \end{aligned} \tag{3}$$

Some studies have applied IG to feature selection for text classification [7–9], including improvements of IG [10, 11].

2.1.2. Mutual Information (MI). MI [12, 13] observes interesting relationship between two variables. MI of f_j in text classification is given by the following formula:

$$\mathrm{MI}\left(f_j\right) = \sum_{k=1}^{t} P\left(c_k\right) \log \frac{P\left(f_j \cap c_k\right)}{P\left(f_j\right) \times P\left(c_k\right)}. \tag{4}$$

MI has been used in feature selection for text classification in current studies [14, 15]. Improvements on MI are also covered in current studies [16, 17].

2.1.3. Gini Index (GI). GI is first introduced in CART (a typical decision tree algorithm) to evaluate impurity of f_j [6, 18]. GI of f_j in text classification is given by the following formula:

$$\mathrm{GI}\left(f_j\right) = 1 - \sum_{k=1}^{t} P^2\left(f_j \mid c_k\right). \tag{5}$$

GI has been used in feature selection for text classification [19]. Improvements of GI are also catch attentions of current studies [20].

2.1.4. Expected Cross Entropy (ECE). The same as IG and MI, ECE [21] is a measure from information theory. ECE of f_j in text classification is given by the following formula:

$$\mathrm{ECE}\left(f_j\right) = P\left(f_j\right) \sum_{k=1}^{t} P\left(c_k \mid f_j\right) \log \frac{P\left(c_k \mid f_j\right)}{P\left(f_j\right)}. \tag{6}$$

ECE is imposed to feature selection for text classification in some studies [22, 23], including improvements of ECE [24].

2.2. Unsupervised Feature Selection Methods. For DTM in (1), $\overline{f}_j$ is the mean of f_j and $\overline{f}_j = (1/n)(\sum_i f_{ij})$.

2.2.1. Document Frequency (DF). DF [7, 25] assumes that frequent terms are more informative than nonfrequent terms. DF is the total number of documents in which f_j appears. DF is usually used as a criterion to evaluate efficiency of other feature selection methods. DF of f_j is given by the following formula:

$$\mathrm{DF}\left(f_j\right) = n\left(f_{ij} > 0\right) \quad \left(i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m\right). \tag{7}$$

$n(f_{ij} > 0)$ denotes the total number of documents in which f_j appears. A discriminative feature gets high DF. Improvements on DF have been proposed in some studies [26, 27].

2.2.2. Variance Score (VS). VS [25] ranks features by calculating variance of each feature f_j in DTM. VS is given by the following formula:

$$\mathrm{VS}\left(f_j\right) = \mathrm{Var}\left(f_j\right) = \frac{1}{n} \sum_{i=1}^{n} \left(f_{ij} - \overline{f}_j\right)^2. \tag{8}$$

A discriminative feature gets high VS. VS is a simple feature selection method used in feature selection for text mining [28].

2.2.3. Laplacian Score (LS). It is the key assumption of LS that data from the same class label are often close to each other and the importance of a feature is evaluated by its power of locality preserving ability and LS is described as follows [3]:

$$\mathrm{LS}\left(f_j\right) = \frac{\sum_{pq} \left(f_{pj} - f_{qj}\right)^2 S_{pq}}{\mathrm{Var}\left(f_j\right)}. \tag{9}$$

S_{pq} evaluates the similarity between D_p and D_q. D_p and D_q are similar if D_p is among k nearest neighbors of D_q or D_q is among k nearest neighbors of D_p. S_{pq} is given by the following formula:

$$S_{pq} = \begin{cases} e^{-\|D_p - D_q\|^2/t}, & \text{if } D_p \text{ and } D_q \text{ are similar} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

t is a suitable constant. S_{pq} is the (p, q) element of weighted matrix S which evaluates similarities of samples of D.

Define vector $l = [1, 1, \ldots, 1]^T$, then $A = \mathrm{diag}(Sl)$. $\mathrm{Var}(f_j)$ in formula (4) is the weighted variance of f_j. $\mathrm{Var}(f_j)$ is given by the following formula:

$$\mathrm{Var}\left(f_j\right) = \sum_i \left(f_{ij} - \overline{f}_j\right)^2 A_{ii}. \tag{11}$$

A_{ii} is the (i, i) element of A. The weighted variance can be simply replaced by standard variance formulated as the following formula:

$$\mathrm{Var}\left(f_j\right) = \sum_i \left(f_{ij} - \overline{f}_j\right)^2. \tag{12}$$

For a discriminative feature, $(f_{pj} - f_{qj})$ is smaller whenever S_{pq} is bigger and, on the other hand, $\mathrm{Var}(f_j)$ is bigger. Thus, LS of a discriminative feature tends to be small. (more details of LS are in [3]).

LS is able to evaluate and rank importance of features and it has been used incurrent studies. He et al. [3] propose LS for feature selection and apply it and data variance (VS) to select features from CUM PIE face database for clustering. Experimental results show that LS performs better than data variance (VS) in several tests with different clusters ($k = 5, 10, 30, 68$). Hu and Man [29] impose LS to spare coding. They use it to evaluate the locality preserving ability of candidate atoms from data examples and select high ranked candidates. Experiments on UCI data demonstrate effectiveness, which also indicates efficiency of LS. Solorio-Fernández et al. [30] propose LS for supervised feature selection. It is imposed to rank features to reduce the search space. This method outperforms better than the other feature selection methods on UCI data sets. Good performance of this method also demonstrates robustness of LS in feature selection. Huang et al. [31] propose LS for face recognition. They impose LS to rank the regular and irregular features in order to achieve the best performances more quickly and it is demonstrated that it is better than other several methods. It is proved that it is capable of giving a better ranking of features in their experiments. Chunlu et al. [32]

study classification of sMRI data in Alzheimer's disease. LS is imposed to select more discriminative features by evaluating their locality preserving abilities. Experimental results prove robustness of LS in feature selection. Bo et al. [33] impose LS to gene feature selection and develop a new method called locality sensitive Laplacian Score (LSLS) by combining LS and locality sensitive discriminant analysis. Experimental results indicate that the method is effective and LS performs well.

Current studies major in the ideas of improving LS. Liu et al. [34] propose an improved method called LSE (Laplacian Score combined with distance-based entropy measure) for unsupervised feature selection. LSE uses entropy measure to replace k-means clustering used in LS to solve the drawbacks of LS. Experimental results of 6 UCI data sets demonstrate that LSE performs better than LS. Padungweang et al. [35] also propose an efficient method based on the strong constraint on the global topology of the data space. Experimental results indicate that compared with conventional LS, this method has better performances. Zhu et al. [36] propose an improved LS method called IterativeLS. It tires improving LS by gradually improving the nearest neighbor graph by discarding the least relevant features at each iteration. Experimental results on both UCI data sets and face databases demonstrate that IterativeLS performs better than LS. Benabdeslem and Hindawi [37] address the problem of semisupervised feature selection for high-dimensional data and propose Constrained Laplacian Score (CLS) to improve LS. Experimental results on 5 UCI data sets demonstrate that CLS is better than LS in the problem of semisupervised feature selection. Moreover, Wang et al. [38] also propose a new method termed Label Reconstruction based Laplacian Score (LRLS) to improve LS for semisupervised feature selection. Experimental results on three UCI data sets indicate that LRLS is clearly better than LS in most occasions.

It is shown that LS is efficient and widely used in feature selection of current studies. It is used as a benchmark technique in order to compare with a new technique in both unsupervised and supervised feature selections [35]. LS is successfully used in some areas such as sparse coding [29], face recognition [31], and gene feature selection [33]. This paper studies feature selection in text classification and argues that conventional LS is not appropriate for feature selection of DTM. This paper proposes an unsupervised method termed DVS to improve LS in feature selection of DTM. Section 3 introduces DVS and analyzes details of DVS.

Current feature selection methods for text classification have been studied a lot and a lot of improvements have been proposed. LS is proved discriminative in the application of several areas. However, LS has not been used in feature selection for text classification currently. We are interested in performances of this proved-discriminative method in text classification. LS performs poorly in experiments. Through further analysis, we found that LS is not appropriate for text classification because LS is not able to select discriminative features from a usually sparse DTM. A new method called DVS is proposed to improve LS in text classification. DVS uses difference measure to replace similarity measure of original LS. More discussions of DVS and LS are covered in Section 4.2.3.

3. Distance Variance Score (DVS)

3.1. Feature Distance Contributions (DC). It is the difference between DVS and LS that DVS takes difference measures among text documents in DTM instead of similarity measures. Let $\mathrm{DC}(f_j)$ denote feature distance contributions of f_j. $\mathrm{DC}(f_j)$ shows how much f_j contributes to differences among text documents in DTM so as to evaluate importance of f_j for text classification. $\mathrm{DC}(f_j)$ is given by the following formula:

$$\mathrm{DC}\left(f_j\right) = \sum_{pq} \frac{\mathrm{dist}\left(f_{pj}, f_{qj}\right)}{\mathrm{dist}\left(D_p, D_q\right)}. \tag{13}$$

dist() is a distance measure. $\mathrm{dist}(D_p, D_q)$ evaluates the difference between D_p and D_q and $\mathrm{dist}(f_{pj}, f_{qj})$ evaluates the difference between f_{pj} and f_{qj}. $\mathrm{DC}(f_r)$ is a ratio to show the importance of $\mathrm{dist}(f_{pj}, f_{qj})$ for $\mathrm{dist}(D_p, D_q)$. $\mathrm{dist}(f_{pj}, f_{qj})$ gives little information to understand how much it contributes to $\mathrm{dist}(D_p, D_q)$. However, if $\mathrm{dist}(D_p, D_q)$ is taken into account, it is easy to understand how much $\mathrm{dist}(f_{pj}, f_{qj})$ contributes to $\mathrm{dist}(D_p, D_q)$. City block distance is introduced to implement dist() because it is a well-know and computationally easy-calculated distance measure [6]. $\mathrm{dist}(D_p, D_q) = \sum_k |f_{pk} - f_{qk}|$ and $\mathrm{dist}(f_{pj}, f_{qj}) = |f_{pj} - f_{qj}|$.

Consider the DTM in (16):

$$\begin{aligned}
\mathrm{dist}\left(D_1, D_2\right) &= \sum_{k=1}^{9} \left|f_{1k} - f_{2k}\right| = 20, \\
\mathrm{dist}\left(f_{13}, f_{23}\right) &= \left|f_{13} - f_{23}\right| = 8, \\
\mathrm{dist}\left(D_3, D_4\right) &= \sum_{k=1}^{9} \left|f_{3k} - f_{4k}\right| = 10, \\
\mathrm{dist}\left(f_{33}, f_{34}\right) &= \left|f_{33} - f_{43}\right| = 5.
\end{aligned} \tag{14}$$

Obviously, $\mathrm{dist}(f_{13}, f_{23}) > \mathrm{dist}(f_{33}, f_{43})$ but it is hard to understand whether $\mathrm{dist}(f_{13}, f_{23})$ contributes more to $\mathrm{dist}(D_1, D_2)$ than $\mathrm{dist}(f_{33}, f_{43})$ contributes to $\mathrm{dist}(D_3, D_4)$. However, if $\mathrm{dist}(D_1, D_2)$ and $\mathrm{dist}(D_3, D_4)$ are taken into account,

$$\begin{aligned}
\mathrm{DC}\left(f_3\right)_1 &= \frac{\mathrm{dist}\left(f_{13}, f_{23}\right)}{\mathrm{dist}\left(D_1, D_2\right)} = 40\%, \\
\mathrm{DC}\left(f_3\right)_2 &= \frac{\mathrm{dist}\left(f_{33}, f_{43}\right)}{\mathrm{dist}\left(D_3, D_4\right)} = 50\%.
\end{aligned} \tag{15}$$

It is clear that $\mathrm{dist}(f_{13}, f_{23})$ contributes 40% to $\mathrm{dist}(D_1, D_2)$ and $\mathrm{dist}(f_{33}, f_{43})$ contributes 50% to $\mathrm{dist}(D_3, D_4)$, which indicates that $\mathrm{dist}(f_{33}, f_{43})$ is more important for $\mathrm{dist}(D_3, D_4)$ than $\mathrm{dist}(f_{13}, f_{23})$ is for $\mathrm{dist}(D_1, D_2)$.

$\mathrm{DC}(f_j)$ is efficient to evaluates importance of a given feature f_j for text documents.

A DTM

$$\begin{array}{cccccccccc} & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 \\ D_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 5 & 9 \\ D_2 & 0 & 0 & 8 & 3 & 0 & 0 & 0 & 6 & 0 \\ D_3 & 4 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ D_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ D_5 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \end{array} \tag{16}$$

3.2. Algorithm of DVS. Let $\mathrm{DTM} = [D_1, D_2, \ldots, D_n]$ denote a document term matrix of n text documents and $D_i = [f_{i1}, f_{i2}, \ldots, f_{im}]$ $(i = 1, 2, \ldots, n)$ denote the ith text document of D. Let f_j $(j = 1, 2, \ldots, m)$ denote the rth term of D and f_{ij} denote the ith term frequency of f_j. Let $\mathrm{DVS}(f_j)$ denote DVS of f_j. $\mathrm{DVS}(f_j)$ is formulated as:

$$\begin{aligned} \mathrm{DVS}\left(f_j\right) &= \mathrm{Var}\left(f_j\right) \times \mathrm{DC}\left(f_j\right) \\ &= \sum_i \left(f_{ij} - \overline{f}_j\right)^2 \sum_{pq} \frac{\left|f_{pj} - f_{qj}\right|}{\sum_k \left|f_{pk} - f_{qk}\right|}. \end{aligned} \tag{17}$$

$\overline{f}_j$ is the mean of f_j. DVS is constructed by two parts: standard variance $\mathrm{Var}(f_j)$ (shown in formula (8)) and feature distance contributions $\mathrm{DC}(f_j)$. $\mathrm{Var}(f_j)$ shows global structure of a given feature and $\mathrm{DC}(f_j)$ shows local structure of a given feature. The key idea of DVS in choosing discriminative features is that a feature f_j is discriminative if it contains much information ($\mathrm{Var}(f_j)$ is large) and it contributes much to differences among documents ($\mathrm{DC}(f_j)$ is large). DVS is able to show differences among documents so as to classify text documents from different classes. S_{pq} and $\|D_p - D_q\|^2$ in original LS (shown in Formula (6)) are replaced by $\mathrm{DC}(f_j)$ to evaluate differences instead of similarities. More details of $\mathrm{DC}(f_j)$ are in Section 3.1. For a discriminative feature f_j, if $\mathrm{DC}(f_j)$ and $\mathrm{Var}(f_j)$ are large, $\mathrm{DVS}(f_j)$ is large, too.

Algorithm of feature selection using DVS is shown in Algorithm 1. $\mathrm{DVS}(f_j)$ of each f_j is calculated using DVS() and all the $\mathrm{DVS}(f_j)$ are sorted in decreasing order using FSDVS() in Algorithm 1. FSDVS() also creates a matrix dmat such that $\mathrm{dmat}[p, q] = \mathrm{dist}(D_p, D_q)$. dmat is used to store city block distances among documents to avoid recalculating each $\mathrm{dist}(D_p, D_q)$ in procedure DVS() so as to improve efficiency of the algorithm. The N features whose DVS are bigger are chosen for text classification. N is determined manually.

4. Experiments

4.1. Data. DBWorld e-mails data set is taken from UCI ML repository and contains 64 e-mails of two classes manually collected from DBWorld mailing list. This data set is represented as DTM. Each attribute corresponds to a term in documents. This data set is donated on the 6th of November, 2011, and it is for classification task. Stop Words have been removed in the preprocessing task. The donator has used it to train different classification algorithms.

Table 1: Details of the two UCI data sets.

Data set	DBWorld data set	CNAE data set
Number of documents	64	1080
Number of terms	3721	856
Data preprocessing task	Stop word removal	—
Associated tasks	Classification	Classification
Date donated	2011-11-06	2012-08-03
Number of web hits	13438	17858

CNAE data set is also taken form UCI ML repository. It contains 1080 documents of free text business descriptions of Brazilian companies categorized into a subset of 9 categories. Each document of this data set is represented as a term vector. CNAE data set is then transformed into a DTM for experiments. The donator donated this data set on the 3rd of October, 2012.

Details about the two data sets are shown in Table 1.

4.2. Discriminative Features Experiments

4.2.1. Experiments of DVS. DVS is calculated using formula (17).

$\mathrm{DVS}(f_j)$ are calculated using DVS() in Algorithm 1 and all the $\mathrm{DVS}(f_j)$ are sorted in decreasing order using FSDVS() in Algorithm 1. For both data sets, the N features of total features are selected by LS and DVS from feature spaces to compare performances of LS and DVS. It shows performances of LS and DVS as N rapidly decreases, choosing N = 2721, 1721, 721, 521, 321, 121, and 21 for DBWorld data set and, moreover, choosing N = 656, 456, 256, 156, and 56 for CNAE data set. DF can be used as a criterion to evaluate performance of a feature selection method for text mining. DF is also applied to experiments to select features from the two data sets to evaluate performance of DVS.

In order to compare efficiency of LS and DVS for feature selection, features selected by LS and DVS are tested by classifiers. The commonly used classifiers for text classification include Naïve Bayes [39, 40], k-nearest neighbor [40, 41], neural network [42, 43], support vector machine (SVM, [40]), and decision tree (DT, [40, 43]). This paper chooses DT(C5.0) and SVM to test features selected by LS and DVS because DT has been used either as main classification tool or as baseline classifier and SVM offers two advantages for text classification, according to Fabrizio's paper [43].

Both data sets are separated into training sets and testing sets. A training set is 70% of the whole data set. DT(C5.0) and SVM classifiers are constructed to test the first N features selected by LS and DVS. Meanwhile, DT(C5.0) and SVM classifiers constructed by total features are used as baselines to compare efficiency of LS and DVS for feature selection.

For each N, 20 DT(C5.0) and 20 SVM classifiers are constructed using randomly separated training sets. "Two-tailed paired t test" is used to test whether feature selection using LS and DVS will affect accuracy of classification. If

```
Procedure FSDVS(DTM, N)
begin
  let mapf denotes the map storing f_j and DVS(f_j) and mapf = null
  #dmat is the matrix to store value of city block distance among documents and
  #dmat[p, q] = dist(D_p, D_q)
  #when p = q, dmat[p, q] = 0
  for each p in Id do
    for each q in Id do
      dmat[p, q] = dist(D_p, D_q);
    end
  end
  #calculate DVS(f_j) for each f_j
  for j in DTM do
    mapf.key = f_j;
    mapf[f_j] = DVS(f_j, dmat);
  end
  #sort mapf in decreasing order according to value
  mapf = sort(mapf);
  #select and return the first N features
  return mapf[1 : N];
end
Procedure DVS(f_j, dmat)
#dmat is the matrix to store value of city block distance among documents and
#dmat[p, q] = dist(D_p, D_q)
#when p = q, dmat[p, q] = 0
begin
  let Var(f_j) denotes the standard variance of f_j
  let DC(f_j) denotes feature distance contributions of f_j
  let Id denotes the index vector of samples in D and Id = [1, 2, ..., n]
  #calculate DC(f_j). See Section 3.1
  for each p in Id do
    for each q in Id and p ≠ q do
      #calculate city block distance between D_p and D_q. See Section 3.1
      distx = dmat[p, q];
      #calculate city block distance between f_pj and f_qj. See Section 3.1
      distf = dist(f_pj, f_qj);
      DC(f_j) += distf/distx;
    end
  end
  return var(f_j) * DC(f_j);
end
```

Algorithm 1: Algorithm of feature selection using DVS.

significances (Sig.'s) are lower than 0.05, then means of the baseline and means of classifiers using LS or DVS are considered not the same (to reject the null hypothesis); that is, LS or DVS will affect accuracy of classification negatively or positively. If Sig.'s are greater than 0.05, then mean of the pairs is considered the same (not to reject the null hypothesis); that is, LS or DVS has no effects on accuracy of classification.

4.2.2. Experimental Results. Figure 1 shows average accuracies of each N of both DT(C5.0) and SVM classifiers of the two data sets. Obviously, DVS outperforms LS and accuracy of classifiers using feature selected by DVS keeps in a high level as N reduces from 2721 to 21 and from 656 to 56. Features selected by DVS are more effective than those selected by LS. At a very small subfeature space, classifiers using LS gets low accuracies. High accuracies can be achieved by using features selected by DVS. Figure 1 also shows that DVS performs as well as DF. Furthermore, Table 2 shows average accuracies (mean) and standard deviations (std) of testing sets of classifiers and results of two-tailed paired t test. For DBWorld data set, most of the significances (Sig.'s) of DT(C5.0) or SVM using LS in Table 2(a) are much lower than 0.05, which means accuracies of classifier using features selected by LS are significantly different form baseline. On the contrary, in Table 2(a), all the significances of the two classifiers using DVS are much higher than 0.05, showing that DVS significantly gets the same good classification accuracies as baseline. For CNAE data set, most of the significances

Table 2: Accuracy of classifiers (unit for mean and std: %).

(a) DBWorld data set

N	LS				DVS			
	DT(C5.0)		SVM		DT(C5.0)		SVM	
	Mean ± std	Sig.	Mean ± std	Sig.	Mean ± std	Sig.	Mean ± std	Sig.
3721	91.30 ± 7.54	—	85.73 ± 6.73	—	91.30 ± 7.54	—	85.73 ± 6.73	—
2721	82.64 ± 8.17	.004	84.14 ± 7.06	.505	89.50 ± 6.59	.736	85.22 ± 6.10	.807
1721	70.27 ± 8.22	.000	80.73 ± 6.68	.023	89.61 ± 5.57	.779	87.02 ± 8.92	.291
721	58.53 ± 10.52	.000	63.97 ± 11.09	.000	91.74 ± 6.07	.433	84.15 ± 8.67	.129
521	56.49 ± 6.94	.000	62.66 ± 9.16	.000	89.07 ± 7.08	.185	85.10 ± 7.57	.584
321	53.58 ± 10.79	.000	57.75 ± 8.83	.000	92.16 ± 4.58	.454	83.55 ± 7.19	.192
121	52.44 ± 10.34	.000	51.92 ± 10.91	.000	92.74 ± 6.61	.179	90.56 ± 6.16	.098
21	58.03 ± 6.73	.000	57.38 ± 9.53	.000	88.26 ± 7.68	.332	91.40 ± 6.41	.722

(b) CNAE data set

N	LS				DVS			
	DT(C5.0)		SVM		DT(C5.0)		SVM	
	Mean ± std	Sig.	Mean ± std	Sig.	Mean ± std	Sig.	Mean ± std	Sig.
856	87.59 ± 2.25	—	92.93 ± 1.58	—	87.59 ± 2.25	—	92.93 ± 1.58	—
656	86.67 ± 2.21	.910	82.19 ± 4.52	.000	87.64 ± 1.77	.924	83.41 ± 2.02	.000
456	86.87 ± 2.62	.394	84.42 ± 2.26	.000	88.25 ± 1.86	.338	85.18 ± 2.73	.000
256	73.11 ± 2.77	.000	66.53 ± 2.09	.000	87.93 ± 2.31	.700	84.53 ± 2.12	.000
156	17.60 ± 1.48	.000	15.08 ± 2.48	.000	86.54 ± 2.46	.146	82.77 ± 2.33	.000
56	8.86 ± 0.930	.000	5.33 ± 0.95	.000	81.99 ± 1.90	.000	77.47 ± 2.33	.000

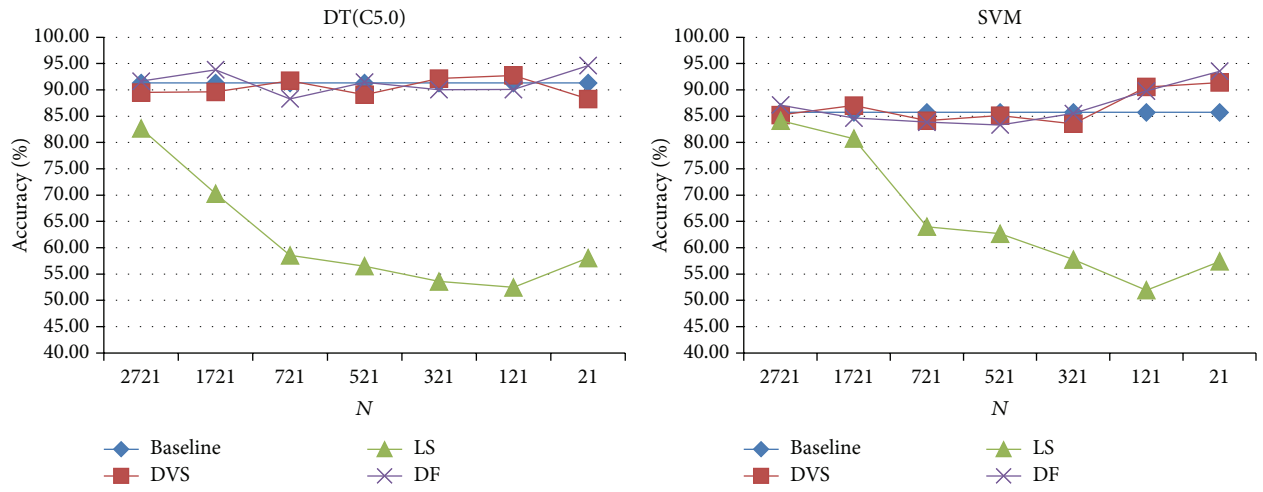
(Sig.'s) of DT(C5.0) or SVM using LS in Table 2(a) are much lower than 0.05, which means accuracies of classifier using features selected by LS are significantly different form baseline. Classifiers using LS gets a very low accuracies when $N = 156$ or 56. Table 2(b) shows that DT(C5.0) using DVS significantly gets the same good accuracies as baseline. However, in Table 2(b), SVM using DVS does not significantly gets the same accuracies as baseline still keeps relatively high accuracies. Figure 1 shows low accuracies of classifier using features selected by LS. Feature selected by LS have negative effects on text classification, especially when N is very small. Figure 1 shows high accuracies of classifiers using features selected by DVS. Feature selected by DVS have positive effects on text classification. Figure 1 also shows that DVS performs better in DT (C5.0) than in SVM.

Experimental results in this paper show that DVS is much better than LS in feature selection of text classification, especially as N is reduced to a small number. Features selected by DVS have positive effects on text classification and features selected by LS make negative effects. DVS is able to overcomes negative effects caused by sparsity of DTM.

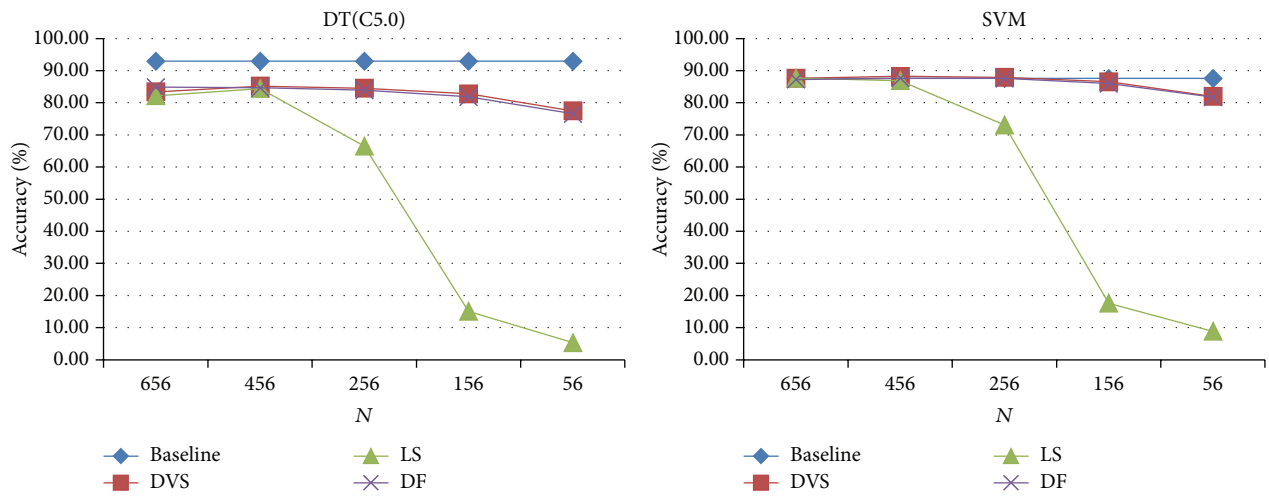
4.2.3. More Discussion about DVS and LS. As mentioned above, it is the key assumption of LS that documents with the same class are close. LS of a discriminative feature should be smaller. Consider formula (10) introduced in Section 2. LS uses S_{pq} to measure similarity between D_p and D_q and $(f_{pj} - f_{qj})^2$ to measure similarity between f_{pj} and f_{qj}. For a discriminative feature, $(f_{pj} - f_{qj})^2 S_{pq}$ tends to small, which indicates that each feature selected by LS contains many 0 values. Thus, LS is easily affected by 0 values in DTM and the sub-DTMs constructed by features selected by LS are also sparse. As N reduces, the sub-DTMs contain less and less information, resulting in pool performances of classifiers in text classification. On the contrary, the key idea of DVS is that text documents with different classes are not close. DVS tries to discover differences instead of similarities among documents and $\mathrm{DC}(f_j)$ are used to evaluate how much a feature contributes to differences among text documents. $\mathrm{DC}(f_j)$ pays much attention to nonzero value and is able to avoid negative effects caused by 0 values. Thus, DVS is capable of reducing the sparsity of DTM and select discriminative features capable of discovering differences among text documents. By discovering differences among text documents, DVS is much more efficient than LS. The sparsity of sub-DTMs of LS and DVS is shown in Table 3. Table 3 shows that as N reduces, the sparsity of sub-DTMs constructed by features selected by DVS also reduce but the sparsity of sub-DTMs constructed by features selected by LS still keep a high level.

5. Conclusion

This paper proposes an unsupervised feature selection method DVS for text classification. DVS aims to improve LS in feature selection of text classification. DVS uses feature distance distributions to discover importance of a feature for text classification and overcomes affects caused by the sparsity of DTM. DVS efficiently reduces the sparsity of DTM



(a) DBWorld data set



(b) CNAE data set

Figure 1: Accuracy of classifiers of two UCI data set.

Table 3: The sparsity of sub-DTMs constructed by features selected by LS and DVS (unit: %).

DBWorld data set			CNAE data set		
N	DVS	LS	N	DVS	LS
3721	95.01		856	99.20	
2721	93.76	97.28	656	99.00	99.01
1721	91.04	98.00	456	98.60	98.81
721	83.16	98.10	256	97.73	98.68
521	79.47	98.03	156	96.60	99.78
321	73.01	98.40	56	92.80	99.90
121	61.23	98.44			
21	49.70	98.44			

and features selected by DVS have positive effects on text classification. Experimental results on UCI DBWorld e-mails data set indicates that DVS performs much better than LS in feature selection of text classification.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by Project of National Social Sciences Foundation, Grant no. 13BTJ005.

References

[1] C. C. Aggarwal and C.-X. Zhai, "An introduction to text mining," in *Mining Text Data*, pp. 1–10, Springer, 2012.

[2] K. K. Bharti and P. K. Singh, "A survey on filter techniques for feature selection in text mining," in *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28–30, 2012*, vol. 236 of *Advances in Intelligent Systems and Computing*, pp. 1545–1559, Springer, New Delhi, India, 2014.

[3] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*, vol. 17, MIT Press, Cambridge, Mass, USA, 2005.

[4] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[5] I. Iguyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[6] J. Han, M. Kamber, and J. Pei, *Data Ming Concepts and Techniques*, 3rd edition, 2012.

[7] Y. M. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, pp. 412–420, 1997.

[8] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning (ECML '98)*, pp. 137–142, Chemnitz, Germany, 1998.

[9] K. Li, X. Diao, and J. Cao, "Improved algorithm of text feature weighting based on information gain," *Computer Engineering*, vol. 37, no. 1, pp. 16–21, 2011.

[10] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Information Processing and Management*, vol. 42, no. 1, pp. 155–165, 2006.

[11] Y. Guo and X. Liu, "Study on information gain-based feature selection in Chinese text categorization," *Computer Engineering and Applications*, vol. 48, no. 27, pp. 119–122, 2012.

[12] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 27, no. 1, pp. 22–29, 1990.

[13] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the International Conference on Information and Knowledge Management*, pp. 148–155, 1998.

[14] Z. Pei, Y. Zhou, L. Liu, L. Wang, Y. Lu, and Y. Kong, "A mutual information and information entropy pair based feature selection method in text classification," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10)*, pp. V6-258–V6-261, IEEE, Taiyuan, China, October 2010.

[15] Z. Ge, "Reverse text frequency based on mutual information on text categorization," *Application Research of Computers*, vol. 29, no. 2, pp. 487–489, 2012.

[16] Z. Lu, H. Shi, Q. Zhang, and C. Yuan, "Automatic Chinese text categorization system based on mutual information," in *Proceedings of the International Conference on Mechatronics and Automation (ICMA '09)*, pp. 4986–4990, IEEE, Changchun, China, August 2009.

[17] H. Liu, Z. Yao, and Z. Su, "Optimization mutual information text feature selection method based on word frequency," *Computer Engineering*, vol. 40, no. 7, pp. 179–182, 2014.

[18] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text clustering," *Expert Systems with Applications*, vol. 33, no. 1, pp. 1–5, 1997.

[19] C. C. Aggarwal and C. X. Zhai, "A survey of text classification algorithms," in *Mining Text Data*, pp. 163–222, Springer, New York, NY, USA, 2012.

[20] H. Park and H. C. Kwon, "Improved Gini-index algorithm to correct feature-selection bias in text classification," *IEICE Transactions on Information and Systems D*, vol. 94, no. 4, pp. 855–865, 2011.

[21] D. Mladenić and M. Grobelnik, "Feature selection on hierarchy of web documents," *Decision Support Systems*, vol. 35, no. 1, pp. 45–87, 2003.

[22] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, pp. 170–178, July 1997.

[23] W. Wang, Y. Kang, and X. Wu, "Study on feature selection in text categorization," *Information Technology*, no. 12, pp. 29–31, 2008.

[24] L.-L. Shan, B.-Q. Liu, and C.-J. Sun, "Comparison and improvement of feature selection method for text categorization," *Journal of Harbin Institute of Technology*, vol. 43, no. 1, pp. 319–324, 2011.

[25] L. Liu, J. Kang, J. Yu, and Z. Wang, "A comparative study on unsupervised feature selection methods for text clustering," in *Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE '05)*, pp. 597–601, November 2005.

[26] Y. Xu, B. Wang, J. T. Li, and H. Jing, "An extended document frequency metric for feature selection in text categorization," in *Information Retrieval Technology*, vol. 4993 of *Lecture Notes in Computer Science*, pp. 71–82, Springer, Berlin, Germany, 2008.

[27] F. Xia, T. Jicun, and L. Zhihui, "A text categorization method based on local document frequency," in *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09)*, vol. 7, pp. 468–471, August 2009.

[28] D. Zhang, S. Chen, and Z.-H. Zhou, "Constraint score: a new filter method for feature selection with pairwise constraints," *Pattern Recognition*, vol. 41, no. 5, pp. 1440–1451, 2008.

[29] J. Hu and H. Man, "Dictionary learning based on laplacian score in sparse coding," in *Machine Learning and Data Mining in Pattern Recognition*, vol. 6871 of *Lecture Notes in Computer Science*, pp. 253–264, Springer, 2011.

[30] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "Hybrid feature selection method for supervised classification based on Laplacian score ranking," in *Advances in Pattern Recognition*, vol. 6256 of *Lecture Notes in Computer Science*, pp. 260–269, Springer, Berlin, Germany, 2010.

[31] H. Huang, H. L. Feng, and C. Y. Peng, "Complete local Fisher discriminant analysis with Laplacian score ranking for face recognition," *Neurocomputing*, vol. 89, pp. 64–77, 2012.

[32] L. Chunlu, L. Ju, and W. Qiang, "Classification of sMRI data in Alzheimer's disease based on UMPCA and Laplacian Score," in *Proceedings of the 9th International Conference on Information, Communications and Signal Processing (ICICS '13)*, pp. 1–4, December 2013.

[33] L. Bo, J. Yan, L. Wei, Z. Wen, C. Lijun, and C. Zhi, "Gene selection using locality sensitive Laplacian score," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 6, pp. 1146–1156, 2014.

[34] R. Liu, N. Yang, X. Ding, and L. Ma, "An unsupervised feature selection algorithm: laplacian score combined with distance-based entropy measure," in *Proceedings of the 3rd International Symposium on Intelligent Information Technology Application (IITA '09)*, pp. 65–68, IEEE, Nanchang, China, November 2009.

[35] P. Padungweang, C. Lursinsap, and K. Sunat, "Univariate filter technique for unsupervised feature selection using a new laplacian score based local nearest neighbors," in *Proceedings of the Asia-Pacific Conference on Information Processing (APCIP '09)*, vol. 2, pp. 196–200, July 2009.

[36] L. Zhu, L. Miao, and D. Zhang, "Iterative laplacian score for feature selection," in *Pattern Recognition*, vol. 321 of *Communications in Computer and Information Science*, pp. 80–87, 2012.

[37] K. Benabdeslem and M. Hindawi, "Constrained Laplacian score for semi-supervised feature selection," in *Machine Learning and Knowledge Discovery in Databases*, vol. 6911 of *Lecture Notes in Computer Science*, pp. 204–218, Springer, Berlin, Germany, 2011.

[38] J. Wang, Y. Li, and J. Chen, "Label Reconstruction based Laplacian Score for semi-supervised feature selection," in *Proceedings of the International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC '13)*, pp. 1112–1115, IEEE, Shengyang, China, December 2013.

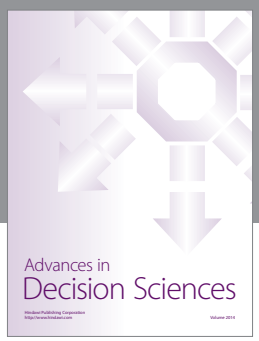
[39] D. D. Lewis, "Naïve (Bayes) at forty: the independence assumption in information retrieval," in *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings*, vol. 1398 of *Lecture Notes in Computer Science*, pp. 4–15, Springer, Berlin, Germany, 1998.

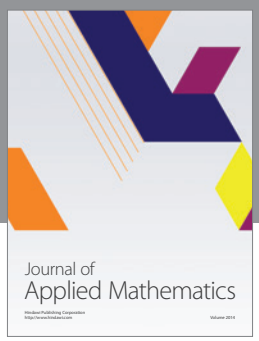
[40] J. Bakus and M. S. Kamel, "Higher order feature selection for text classification," *Knowledge and Information Systems*, vol. 9, no. 4, pp. 468–491, 2006.

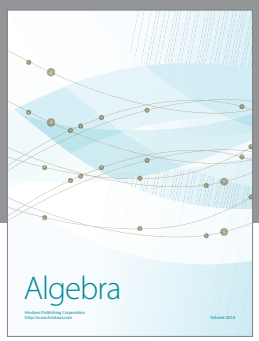
[41] S. Manne, S. K. Kotha, and S. S. Fatima, "Text categorization with K-nearest neighbor approach," in *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012*, vol. 132 of *Advances in Intelligent and Soft Computing*, pp. 413–420, Springer, Berlin, Germany, 2012.

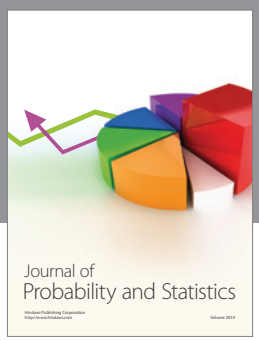
[42] E. Wiener, J. Pedersen, and A. Weigend, "A neural network approach to topic spotting," in *Proceedings of the Symposium on Document Analysis and Information Retrieval*, San Jose, Calif, USA, February 1995.

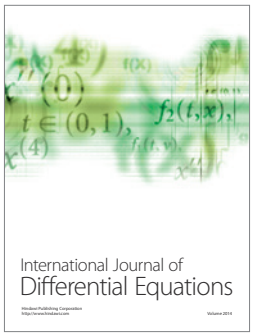
[43] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

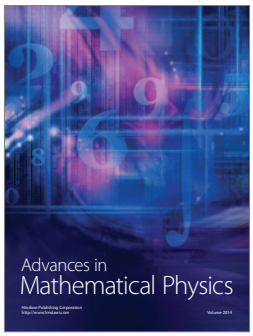
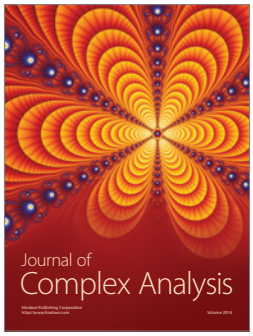
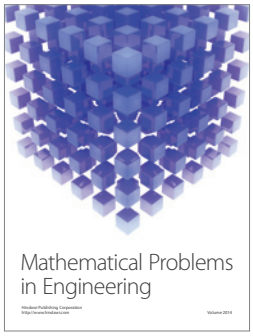
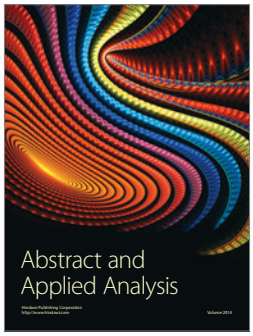
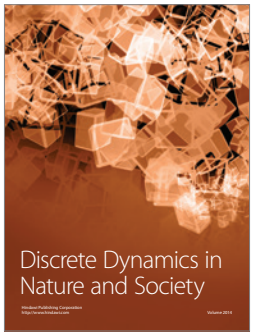
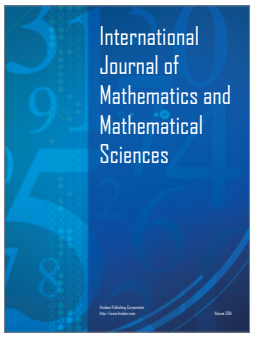
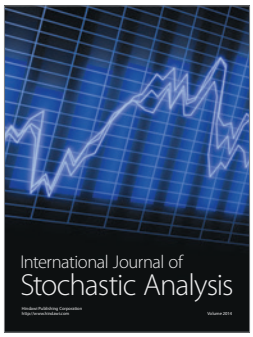










Hindawi

Submit your manuscripts at
http://www.hindawi.com