Research Article

A New Feature Selection Method for Text Classification Based on Independent Feature Space Search

Yong Liu,1,2 Shenggen Ju,1 Junfeng Wang,1 and Chong Su3

1Department of Computer, University of Sichuan, Chengdu, Sichuan Province 610065, China
2Information Center, Nanjing Jiangbei People’s Hospital, Nanjing, Jiangsu Province 210048, China
3Engineering Research Center of Medicine Information, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu Province 210003, China

Correspondence should be addressed to Shenggen Ju; jsg@scu.edu.cn

Received 3 November 2019; Accepted 11 March 2020; Published 12 May 2020

Academic Editor: Sajad Azizi

Copyright © 2020 Yong Liu et al. 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.

Feature selection method is designed to select the representative feature subsets from the original feature set by different evaluation of feature relevance, which focuses on reducing the dimension of the features while maintaining the predictive accuracy of a classifier. In this study, we propose a feature selection method for text classification based on independent feature space search. Firstly, a relative document-term frequency difference (RDTFD) method is proposed to divide the features in all text documents into two independent feature sets according to the features’ ability to discriminate the positive and negative samples, which has two important functions: one is to improve the high class correlation of the features and reduce the correlation between the features and the other is to reduce the search range of feature space and maintain appropriate feature redundancy. Secondly, the feature search strategy is used to search the optimal feature subset in independent feature space, which can improve the performance of text classification. Finally, we evaluate several experiments conducted on six benchmark corpora, the experimental results show the RDTFD method based on independent feature space search is more robust than the other feature selection methods.

1. Introduction

In the task of text classification, how to remove the noise in a large number of text documents, such as the irrelevant and redundant features, is a challenging topic. Therefore, the dimension reduction methods have been proposed to solve this problem, including feature extraction and feature selection. Feature extraction methods extract features from the new and the low-dimensional feature space that is transformed from original feature space, such as principal component analysis (PCA) [1], Linear Discriminant Analysis (LDA) [2], and so on. Feature selection methods that select a small subset from the original feature set by the different evaluation on feature relevance, there are two main models that deal with feature selection, namely, wrapper models and filter models [3]. Wrapper models apply the specific classifier to evaluate and select features, which can generate different feature sets [4], although wrapper models can find a better and nonredundant feature set by the classifier with cross validation [5], and nevertheless, in the case of the high-dimensional space, the wrapper models will consume more feature search time than the filter models [6]. In contrast to wrapper models, filter models only rely on various evaluation algorithms rather than classifiers [7], and moreover, in many cases, filter models can be applied to scale large datasets because of its high-efficiency processing speed [8]. Within filter models, feature ranking methods attempt to rank features according to feature relevance based on different evaluation [7], many related methods have been widely used in feature selection for text classification [9, 10], which can be categorized into three groups: document frequency (DF), term frequency (TF), and document-term frequency (DTF), among which many theories have been proposed, such as Chi-square (CHI) [10], information gain (IG) [10], term frequency based on information gain (TFIG) [11], Gini index [12], improved Gini index (IMGI) [13],
normal term frequency based Gini index (GININTF) [14],
discriminative power measure (DPM) [15], odds ratio (OR)
[16], $t$-test based feature selection (TFFS) [17], document
frequency (DF) [10], term frequency-inverse document
frequency (TFIDF) [18], and improved TFIDF (IMTFIDF)
[19]. CHI, IG, GINI, IMGI, DF, DPM, and OR belong to DF,
and TFFS and TFIG belong to TF, and moreover, TFIDF
and IMTFIDF belong to DTF. Note that the feature/features
is equivalent to the term/terms in this study, such as features
number is equivalent to terms number.

The motivation behind this work is the following. We
know that the feature ranking methods are often used for
feature selection, which assign the feature weightings that
are often normalized in the range (0, 1) to each feature [20],
then rank the feature weightings in descending order, and
finally, select the top $N$ features by the feature weightings.
Although the feature weightings reflect the importance of
the features in feature set, they cannot guarantee to construct
a better feature set for classification, and the main reason is
the higher correlation between the features, which leads to
the redundancy of the features [21], therefore, the feature
ranking methods are difficult to obtain much gain [22]. If a
feature is highly correlated not only to the class but also to
other features, it is redundant and should be removed [8],
and therefore, Minimum Redundancy-Maximum Relevance
(mRmR) was proposed to minimize the correlation between
features and maximize the relevance of the features with
class, and it has significantly improved classification
performance on five gene expressions data sets [21]. However,
some learning algorithms use the redundant but correlated
features, which obtain the better performance on some
datasets [4]. Overall, we known that the relationship between
the features is extraordinarily complex. Hence, if we ex-
cessively seek features of the lower redundancy or better
purity can lead to some good features to be abandoned,
which can cause the negative effect on classification.

To sum up, we know that a good feature set should
contain two important characteristics, one is the high class
Correlation of the features and the lower correlation between
the features, the other is the moderate redundancy of the
features. Therefore, our research focuses on finding a feature
selection method that can meet both characteristics
simultaneously.

In this paper, we propose a novel and effective idea of
feature selection and use the diagrams to illustrate the
difference between this method and the general feature
selection method. Note that we call others the feature selec-
tion method as the general selection method, except the
new feature selection method proposed in this paper.

Figure 1 shows the process diagram of the general feature
selection methods, which consists of four steps: step ① shows
all features are added to the original features set, step ②
represents the calculating of all features weighting, step ③
represents that all features are ranked in descending order
according to the weighting of the features, and step ④
represents that top $N$ features are selected from the feature set.

Figure 2 shows the process diagram of the new feature
selection method, namely, the RDTFD method, step ①
represents all features are added to the original features set.

Step ② consists of two substeps. The first substep represents
that all features are divided into the positive or negative
feature subsets according to the positive or negative values of
the weighting, which could improve the high class relevance
of the features and reduce the correlation between the
features, this step corresponds to equation (8) in Section 3.1.
The second substep is designed to reduce the search range of
feature space and maintain appropriate feature redundancy,
which corresponds to equation (9) in Section 3.1. Step ③
represents that all features of the positive feature subset are
ranked in descending order by weighting, and all features of
the negative feature subset are ranked in ascending order by
weighting. Step ④ represents that the candidate feature
subsets are selected from two independent feature subsets
with some search strategies, as detailed in Section 3.2.

Comparing with the general feature selection method,
the RDTFD method has two major differences. Firstly, the
RDTFD method divides the original feature set into two
independent feature subsets, and however, the general selec-
tion method generates a feature set. Secondly, the RDTFD
method selects feature subsets by some search strategies
from two independent feature subsets, which are flexible and
scalable, but the general selection method is only selects top
$N$ features from a feature set.

In this paper, the application scenario of spam filtering is
given to illustrate the advantages of the proposed feature
selection method over the general feature selection method.
As we known, the content-based filtering methods can be
used to improve the accuracy of e-mail classification by
machine learning, such as Naive Bayesian classifiers [23, 24],
K-nearest neighbor [25], neural networks [26, 27], Support
Vector Machines [28, 29], Boosting [30, 31], Three-way
decisions [32, 33], and so on. The Naive Bayesian method is
often used, because of its high efficiency and accuracy.
Moreover, the support vector machine attempts to reduce
the generalization error by using the constraints of the
decision boundary and achieve the better performance [34].

The remaining paper is divided into following main
sections: In Section 2, we review related work on feature
selection methods. In Section 3, we proposed a relative
document-term frequency difference (RDTFD) method and
the particle swarm optimization (PSO) algorithm based on
independent feature space search to achieve high perfor-
mane for text classification. In Section 4, we describe the
experimental results. In Section 5, we conclude our work
with some possible extensions.

2. Related Work

2.1. Chi-Square (CHI). The chi-square test is one of the most
useful statistical methods, which not only provides the in-
formation with respect to the significance of any observed
differences, but also provides the information of categories
difference [35]. Compared to the $t$-test, the chi-square test
does not assume the data to meet the normal distribution,
and the null hypothesis of the independence will be rejected
when there is a significant difference between observed
frequency and expected frequency [36]. The feature mea-
surement is defined to be
\[ \chi^2(t_i) = \max_{c_k \in \{s, h\}} \chi^2(t_i, c_k) \],
\[ \chi^2(t_i, c_k) = \frac{N (a_{ik}d_{ik} - c_{ik}b_{ik})^2}{(a_{ik} + c_{ik})(b_{ik} + d_{ik})(a_{ik} + b_{ik})(c_{ik} + d_{ik})}. \]

where \( N \) is the total number of documents in the corpus, \( a_{ik} \) is the number of documents in class \( c_k \) that contains the feature \( t_i \), \( b_{ik} \) is the number of documents in class \( c_k \) that contains the feature \( t_i \) and does not belong to class \( c_k \), \( c_{ik} \) is the number of documents in class \( c_k \) that does not contain the feature \( t_i \), and \( d_{ik} \) is the number of the documents that do not contain the feature \( t_i \) and do not belong to class \( c_k \). When feature \( t_i \) is independent of class \( c_k \), the chi-square will obtain 0 value. In addition, the chi-square value is normalized in the calculation process, which exaggerates the role of the low-frequency feature [10].

2.2 Information Gain (IG). The information gain can be used to measure capability of information acquisition for class prediction by the presence or absence of a term in a document. If most of the terms do not appear in most class, the information gain value will incline to the case of term
probability that a document contains or does not contain spam class, the distributions of the terms in relevant class and corpus. Based on the f\textsubscript{ñ}ñ he algorithm is follows \[10\]:

\[
IG(t_i) = - \sum_{c_k \in \{s,h\}} p(c_k) \log(p(c_k)) + p(t_i) \sum_{c_k \in \{s,h\}} p \\
\quad \cdot (c_k \mid t_i) \log p(c_k \mid t_i) + p(T_i) \sum_{c_k \in \{s,h\}} p(c_k \mid T_i) \log p \\
\quad \cdot (c_k \mid T_i),
\]

\[(2)\]

where \( s \) denotes spam class, \( h \) denotes ham class, \( p(c_k) \) is the probability of the documents in \( c_k \), \( p(t_i) \) and \( p(T_i) \) are the probability that a document contains or does not contain \( t_i \), and \( p(c_k \mid t_i) \) and \( p(c_k \mid T_i) \) are the conditional probability that a document in \( c_k \) contains and does not contain \( t_i \).

2.3. Term Frequency-Based Information Gain (TFIG). TFIG is an information gain method based on the term frequency, which can measure the amount of information obtained for class prediction by knowing one appearance, multiple appearances, or absence of a term in a document. The TFIG can be defined as follows \[11\]:

\[
TFIG(t_i) = - \sum_{c_k \in \{s,h\}} p(c_k) \log(p(c_k)) + p(T_i) \sum_{c_k \in \{s,h\}} p \\
\quad \cdot (c_k \mid t_i) \log p(c_k \mid t_i) + p(t_i) \sum_{c_k \in \{s,h\}} p \\
\quad \cdot (c_k \mid t_1) \log p(c_k \mid t_1) + \theta \times p(t_2) \sum_{c_k \in \{s,h\}} p \\
\quad \cdot (c_k \mid t_2) \log p(c_k \mid t_2),
\]

\[(3)\]

where \( t_1 \) denotes the one time appearance of the term \( t_i \), \( t_2 \) denotes the multiple appearances of the term \( t_i \), and \( \theta \) is constant parameter (\( \theta \geq 1 \)).

2.4. Improved Gini Index (IMGI). The Gini index is a measurement of the purity of the term, the greater purity, the better classification performance. Gini Index can be demonstrated as follows:

\[
GINI(t_i) = p(t_i \mid s)^2 \times p(s \mid t_1)^2 + p(t_i \mid h)^2 \times p(h \mid t_1)^2,
\]

\[(4)\]

where \( p(t_i \mid s) \) is the probability that the term \( t_i \) occurs in spam class, \( p(s \mid t_1) \) is the conditional probability that an e-mail belongs to spam class when \( t_1 \) occurs, \( p(t_i \mid h) \) is the probability that the term \( t_i \) occurs in ham class, \( p(h \mid t_1) \) is the conditional probability that an e-mail belongs to ham class when the term \( t_1 \) occurs.

2.5. T-Test-Based Feature Selection (TTFS). The algorithm is based on the T-test, which can measure the different distributions of the terms in relevant class and corpus. T-test can be defined as follows:

\[
TTFS(t_i, c_k) = \frac{|tf_{ki} - tf_{ji}|}{m_k \times s_i},
\]

\[(5)\]

where \( c_k \) denotes spam or ham class, \( N_k \) is the number of e-mails in \( c_k \), \( k \) is the number of categories, \( tf_{ki} \) is the average term frequency of the term \( t_i \) in \( c_k \), \( tf_{ji} \) is the average term frequency of the term \( t_i \) in all e-mails, \( N \) denotes the number of all e-mails, and \( s_i \) denotes standard deviation within a class. If there is threshold \( \theta \), when TTFS \( (t_i, c_k) < \theta \), which shows that \( tf_{ji} \) has the same or similar mean value in the whole e-mails, in this case, the term has less discrimination capability for class \( c_k \), otherwise, it shows that the term has more discrimination capability for class \( c_k \).

2.6. Improved Term Frequency-Inverse Document Frequency (IMTFIDF). According to the TFIG, the term with smaller frequency in e-mails is a good distinguishing capability, otherwise, the term has a poor distinguishing capability, and however, this theory cannot effectively reflect importance of all the terms in practice. The improved TFIDF (IMTFIDF) was proposed \[19\], which is defined as follows:

\[
IMTFIDF(t_i, d_j, c_k) = tf_{ij} \times \log \left( \frac{N}{K_i} \right), \quad \text{if} \quad \frac{M_j}{M_j + K_i} > 70\%,
\]

\[(6)\]

where \( N \) denotes the number of document frequency in all e-mails, \( c_k \) denotes spam or ham class, \( K_i \) is the number of e-mails which contain term \( t_i \) but do not belong to \( c_k \), \( M_j \) is the number of e-mails which contain term \( t_j \) and belong to \( c_k \), and \( tf_{ij} \) is the term frequency of the term \( t_i \) in e-mail \( d_j \) of \( c_k \). If \( \frac{M_j}{M_j + K_i} \) is bigger than 70\%, then the term \( t_i \) can well represent the text features of this class of e-mail documents, and otherwise, the \( t_i \) will be abandoned.

2.7. Normal Term Frequency-Based Gini Index (GININTF). Considering some drawbacks of using document frequency, GININTF was proposed for feature selection by term frequency \[14\], which is defined as follows:

\[
GININTF(t_i) = \sum_{c_k \in \{s,h\}} \left( \frac{A(t_i, c_k)}{M_k} \right)^2 \left( \frac{A(t_i, c_k)}{A(t_i, c_k) + B(t_i, c_k)} \right)^2,
\]

\[(7)\]
where \( A(t_i, c_k) \) is the total normalized term frequency of the term \( t_i \) of e-mails that belongs to class \( c_k \), \( B(t_i, c_k) \) is the total normalized term frequency of the term \( t_i \) of e-mails that does not belong to class \( c_k \), and \( M_{c_k} \) is the total number of e-mails belongs to class \( c_k \).

### 3. Feature Selection Method Based on Independent Feature Space Search

#### 3.1. Relative Document-Terms Frequency Difference (RDTFD)

In order to describe the RDTFD method clearly, we list the definition of the concept related to the frequency in Table 1, including document frequency, term frequency, average term frequency, and term frequency distribution.

Table 2 shows the number of documents and the total number of terms in ham and spam datasets, it is worth noting that Dataset1 and Dataset2 including spam and ham datasets, respectively. Table 3 shows some samples and the corresponding document frequency, term frequency, average term frequency, and term frequency distribution in ham and spam datasets, respectively.

As shown in Table 3, when DF methods (CHI, IG, and IMGI) are applied, the term “mailings” can obtain the highest value of DF methods, which shows that “mailings” has a discriminative capability compared with others terms. Moreover, the document frequency of “mailings” in spam class is higher than that in ham class, which means that an e-mail document tends to spam class if it contains “mailings.” Further, although “marketing” and “linguistic” have the same document frequency, however, “marketing” has a higher average frequency in ham class, which means that it is more likely to appear in ham class than in spam class.

Since “debian” and “workshop” have the same document frequency in spam and ham class, if DF methods such as IG is used, “debian” and “workshop” will be abandoned. In addition, according to TTFS method GININTF, “debian” and “workshop” will get the same GININTF value, which shows that “debian” and “workshop” cannot be identified as spam or ham class. However, the term frequency distribution of “workshop” is more variable in ham class than in spam class, and therefore, an e-mail document containing “workshop” will tend to spam class by using TTFS method, although TTFS method is interpretable and easy to implement. Nevertheless, the method is based on \( t \)-test, namely, the prior distribution of the data must conform to normal distribution, this assumption is not consistent with distribution of the real data [37], and hence, it cannot obtain the better performance for feature selection.

In this section, we propose a relative document-terms frequency difference (RDTFD) method, and it considers not only the document frequency and term frequency, but also the number of documents and the total number of terms, which can construct a better model to measure the ability to identify spam or ham class for each term and it can be demonstrated as follows:

\[
\text{RDTFD} = \frac{\log(d_{f,t_i,\text{ham}})}{\log(D_{\text{NH}})} \times \frac{\log(t_{f,t_i,\text{ham}})}{\log(T_{\text{NH}})} - \frac{\log(d_{f,t_i,\text{spam}})}{\log(D_{\text{NS}})} \times \frac{\log(t_{f,t_i,\text{spam}})}{\log(T_{\text{NS}})}
\]

(8)

if \( \text{MAX} \left( \frac{d_{f,t_i,\text{ham}}}{d_{f,t_i,\text{ham}} + d_{f,t_i,\text{spam}}} - \frac{d_{f,t_i,\text{spam}}}{d_{f,t_i,\text{ham}} + d_{f,t_i,\text{spam}}} \right) > 0.7 \),

(9)

where \( d_{f,t_i,\text{ham}} \) denotes the document frequency of term \( t_i \) in ham documents, \( d_{f,t_i,\text{spam}} \) denotes the document frequency of term \( t_i \) in spam documents, \( t_{f,t_i,\text{ham}} \) denotes the term frequency of term \( t_i \) in ham documents, \( t_{f,t_i,\text{spam}} \) denotes that term frequency of term \( t_i \) in spam documents, \( D_{\text{NH}} \) is the number of ham documents, \( D_{\text{NS}} \) is the number of spam documents, \( T_{\text{NH}} \) is the total number of terms in ham documents, and \( T_{\text{NS}} \) is the total number of terms in spam documents.

Equation (8) is intended to address the content mentioned in the first substep of step 2 in Figure 2, which shows that when document and term frequency of term \( t_i \) in spam class are very close to ham class, RDTFD value is approximately equals to zero, which means that term \( t_i \) has a poor discrimination capability in spam or ham class. Therefore, the bigger positive RDTFD weighting represents a document that containing the term \( t_i \) tends to ham class and the smaller negative RDTFD weighting represents a document that containing the term \( t_i \) tends to spam class, which can construct two independent term subsets, namely, the terms of the positive weighting will be put into the subset of ham class and the terms of the negative weighting will be put into the subset of spam class. For instance, in Table 2, since term “debian” has same document frequency, term frequency and distribution of term frequency, it has no discriminating capability in spam or ham class according to the typical DF and TF methods. However, the total terms number of documents contain “debian” in spam class more than in ham class, we know \((\log(10)/\log(10)) \times (\log(20) /\log(2000)) - (\log(10)/\log(10)) \times (\log(20)/\log(5000)) > 0\), by which, a document contains “debian” will tend to ham class.

Since the distribution of each term in the text is very complicated and cannot be accurately measured by a constant formula, we can delineate a general scope for the distribution of these terms, as long as the terms in this scope are better for text classification. Furthermore, to limit the search range of features can not only reduce the time cost of feature search, but also maintain appropriate feature redundancy. Thus, we propose equation (9) that makes the term \( t_i \) be selected as the candidate term belonging ham or spam class; otherwise, the term \( t_i \) will be abandoned, which is similar to that mentioned in [19]. In the experiment of this paper, when this equation (9) is satisfied, the text classification performance would be better.
3.2. Feature Search Strategy. From the step 4 in Figure 2, we know that the candidate feature subsets are selected from two independent term spaces by some search strategies. Compared with other optimization algorithms, the parameter setting of particle swarm optimization (PSO) algorithm is convenient and the population is rich. In addition, the convergence speed of PSO algorithm is fast and it is suitable to search the high-dimensional feature spaces. Therefore, in this study, we apply the PSO algorithm as the search strategy for the feature selection method.

The PSO was proposed by Kennedy and Eberhart in 1995, the original inspiration was from the behavior of birds flocking and fish schooling [38]. In our experiment, the PSO algorithm that can search the optimal term set by multi-iteration and is given as follows [39]:

\[
\begin{align*}
    v_{id}^{t+1} &= w \times v_{id}^t + c_1 \times r_1 \times (p_{id}^t - x_{id}^t) + c_2 \times r_2 \times (p_{gd}^t - x_{id}^t), \\
    x_{id}^{t+1} &= x_{id}^t + v_{id}^t, \quad d = 1, 2, \ldots, D.
\end{align*}
\]

(10)

Figure 3 shows the process of feature search strategy implementation, the dotted line rectangle can be viewed as a feature selection window, the red part is from \(S_1\) to \(S_m\), which shows that the negative term weighting of spam is sorted in the ascending order, and the green part is from \(H_1\) to \(H_m\), which shows that the positive term weighting of ham is sorted in the descending order. In addition, the movement of the feature selection window is controlled by two parameters, the first parameter is the width of the feature selection window that determines the number of terms for classification, the second parameter is the ratio of spam terms number to the terms number, which is in the interval (0,1) and generated by the PSO algorithm, note that the terms number includes the spam terms number and the ham terms number. In Figure 3, \(R_1 \rightarrow R_m\) represents each ratio, and \(m\) denotes the particle number. The feature selection window will move back and forth between ham and spam term set by the ratio value, which can generate \(m\) candidate term subsets, and they can be fed to a specific classifier such as NB or SVM to select the optimal term set with the highest F1 values. For instance, when the width of feature selection window is set to 10, namely, \(N = 10\), it denotes 10 terms will be selected between spam and ham term subsets for spam filtering, and moreover, the ratio is set to 0.7, which denotes feature selection window will select terms in terms of spam and 3 terms in terms of ham. In extreme cases, the ratio approaches to 0 or 1, which means that all terms of ham or spam will be selected as a candidate term subset, respectively. Therefore, the term search space is \((2 \times 10 = 20)\), namely, \(O(2N)\).

Suppose there are \(N\) terms in term set, exhaustive search space is \(O(2^N)\), and greedy search sequential search space is \(O(N^3)\) [12], which are impractical unless \(N\) is a smaller value, it is unpractical and unnecessary for the RDTFD method to search all possible candidate terms from ham and spam term set. Moreover, because Chi-square and Information gain methods can reach the peaked value in 2000 terms [10], we set maximum number of candidate terms to \(n = 2000\) in this experiment, and in general, \(n\) is the term number of term subset and far less than \(N\).
We can view each feature in the feature space as each particle in the search space of PSO. In equation (10), \( D \) denotes dimensions of term space, \( v_{id} \) denotes the speed of particle \( i \) on the \( d \)th dimension at the iteration \( t \), which represents the ratio of spam terms number to the terms number. \( g_{bd} \) denotes the best position of individual particle \( i \) comparing with its current best fitness at iteration \( t \). \( p_{gd} \) denotes the globally best position of all particles comparing with its current globally best fitness at iteration \( t \). \( x_{id}^{t+1} \) denotes the position of particle \( i \) on the \( d \)th dimension at the iteration \( t \). \( x_{id}^{t} \) denotes the new position of particle \( i \) on the \( d \)th dimension at the iteration \( t+1 \). \( W \) denotes inertia weighting which controls the velocity in the previous iteration on the current velocity, \( W \) is nonnegative value which can adjust the scope of space search, and we initialize \( W_{min} = 0 \) and \( W_{max} = 1 \), \( c_1 \) and \( c_2 \) are acceleration constants initialized to 2, which can adjust maximum step size, \( r_1 \) and \( r_2 \) are random numbers, in the interval (0,1), which can be used to increase the randomness. Moreover, the maximum and minimum \( v_{id} \) of particle are \( V_{max} = 0.01 \) and \( V_{min} = 0.001 \), respectively. The maximum and minimum \( x_{id} \) of particle are \( X_{max} = 0.999 \) and \( X_{min} = 0.001 \), respectively. The maximum iteration is set to \( N_i = 30 \), for each iteration \( t \), \( W \) is given as follows:

\[
W^{t+1} = W_{max} - \left( \frac{W_{max} - W_{min}}{N_i} \right) t.
\]

### 3.3. Correlation Algorithm

Algorithms 1 and 2 outline how the RDTTFD method obtains the optimal term set by using PSO algorithm.

Note that the \( F1 \) value in step 3.3 of Algorithm 1 represents the mean value of \( F1 \) measure, which is defined as follows:

\[
F1 = \frac{2 \times r \times p}{r + p}.
\]

In equation (12), \( r = (n_{p} / (n_{p} + n_{h})) \), \( p = (n_{h} / (n_{p} + n_{h})) \), \( n_{p} \) represents the number of spams that are correctly identified as spams, \( n_{h} \) represents the number of spams that are identified as hams, \( n_{hp} \) represents the number of hams that are identified as spams.

In the step 2.18 of Algorithm 2, we apply evaluation function on each iteration to obtain the optimal candidate term set, which is defined as follows:

\[
F = EVAL(Ft). \tag{13}
\]

In equation (13), \( Ft \) denotes the candidate term subset, note that the candidate term subsets range from 200 terms to \( n \) terms with intervals of 200 terms, and \( n \) denotes the number of terms for text classification. The function \( EVAL(Ft) \) evaluates the performance on NB or SVM classifier with 10-fold cross validation, which can obtain the mean value of \( F1 \) measure.

In order to improve the search efficiency of particles and obtain optimal term set, we consider to select the particle of minimum \( F1 \) value and remove it at each iteration and further to construct a new mutation particle into particle swarm for new iteration.

### 4. Experimental and Results

#### 4.1. Experimental Environment Configuration

In this experiment, we use Intel(TM)-i5 Processor with a CPU clock rate of 3.2GHZ and 8GB main memory. The feature selection methods run on the platform of windows7 Ultimate and python3.5.
4.2. Algorithms Selection for the Experiment. For the convenience of description, the RDTFD method and PSO algorithm based on independent feature space search are called RDTFD. In order to evaluate the performance of RDTFD, we selected six feature selection methods, including CHI, IG, TFIG, TTFS, IMTFIDF, and GININTF.

4.3. Datasets. In this experiment, we chose six spam datasets from a wide variety of applications, such as PU123A [40], CSDMC2010 (CS), and Enron-spam3 (ES) [41], which are shown in Table 4.

4.4. Stopwords. The stopwords are usually removed to improve performance for classification in spam filtering, however, we do not apply it in our experiment in the light of two main reasons. Firstly, since stopwords are language-specific and domain-specific, removing these words can lead to the negative result [42]. Secondly, in order to protect the privacy personal e-mails, PU1, PU2, PU3, and PUA are encrypted with a set of digits, and therefore, it is impossible to identify which of the encrypted contents are belong to stopwords. Moreover, CS, and ES datasets consist of many nonnumerical characters, therefore, in order to unify the experimental standards, we do not simply remove the terms from these datasets.

4.5. Classifiers and Evaluation Measure. Because this paper focuses on feature selection algorithm, rather than discussing the advantages and disadvantages of classifiers, we apply the common classifiers to meet the requirement of this paper, namely, the NB [43] and SVM [44] are used to evaluate the performance of feature selection methods, respectively. Because multinomial NB is the better model in NB classifier, which can achieve the highest classification performance on multiple datasets [45], it will be used as a classifier in this experiment. In addition, because the popular sequential minimal optimization (SMO) classifier is the linear support vector machine, which can handle very large training set and has a higher performance in sparse datasets [46], we utilize it as another classifier in this experiment. Finally, the experiment was conducted on six corpus by utilizing tenfold cross validation [47], by which we apply F1 measure to evaluate classification performance on each dataset.

4.6. Performance Comparison of Different Feature Selection Methods. Figure 4 shows the performance of seven feature selection methods on PU1, PU2, PU3, PUA, CS, and ES by using F1 measure when NB and SVM are used, respectively. Further, we design Tables 5 and 6 to show the relation about the feature selection methods, terms number (column terms), and classifiers on six datasets, respectively.

Tables 5 and 6 include four column parts: the first column part is Datasets, which represents each dataset for experiment, the second column part is Method/Terms Number/Highest F1, which means the highest F1 value based on N terms and the feature selection methods, the third column part is Method/Terms Number/Lowest F1, which denotes the lowest F1 value based on N terms and the feature selection methods, and the fourth column is Method/Lowest terms Number/Highest F1, which represents the highest F1 value based on the lowest N terms and the feature selection methods. Note that terms range from 200 to 1000 with intervals of 200 terms. In Tables 5 and 6, we know that RDTFD method outperforms the other methods in text classification. Moreover, since F1 value of SVM based on SMO almost completely surpasses NB on seven datasets, the SVM is more suitable than NB in text classification.

```
Input: the e-mail dataset
Output: Gs, Gt and Gr
Step 1: parameters initialization
(1.1) set Gs = null/initialize variable Gs to preserve the optimal F1 value
(1.2) set Gt = null/initialize variable Gt to preserve the optimal terms
(1.3) set Gr = null/initialize variable Gr to preserve the optimal ratio
(1.4) set N = 10000/initialize maximal candidate number of terms for feature selection
Step 2: generating training term set
(2.1) Calculate term weighting of all terms according to equations (8) and (9)/since the log function is used in both the numerator and denominator of equation (8), when the parameter values of the log function is 0 or 1, the value of the numerator divided by the denominator will be set to the constant 0
(2.2) The terms of positive weighting are put into Fh
(2.3) The terms of negative weighting are put into Fs
(2.4) Ranking all the terms of Fh by the weighting in descending order, ranking all the terms of Fs by the weighting in ascending order
Step 3: seeking the optimal candidate term subset by using PSO algorithm
(3.1) Transmitting Fh, Fs, and N into PSO algorithm
(3.2) Run PSO algorithm (Algorithm 2)
(3.3) Return the optimal F1 value and put into Gs
(3.4) Return the optimal term subset and put into Gt
(3.5) Return the optimal ratio and put into Gr

Algorithm 1: RDTFD method (terms grouping and ranking, and to call PSO algorithm).
```
4.7. Distribution of Particles in Different Term Spaces.

In this experiment, the PSO algorithm is used to find the optimal term set for spam filtering by searching the best particle that represents the ratio of spam terms number to terms number. Tables 7 and 8 include three column parts. The first column part has Datasets, which represent each dataset for experiment. The second column part has Terms Number/Highest $F_1$/Ratio, which means the ratio of spam terms number to the terms number when the RDTFD method obtains the highest $F_1$ values. The third column part has Terms Number/Lowest $F_1$/Ratio, which means the ratio of spam terms number to the terms number when the RDTFD method obtains the lowest $F_1$ values.

4.8. Analysis of Time Complexity. Compared with the general feature selection, although the RDTFD method takes much...
Figure 4: Continued.
Table 5: Performance of feature selection methods in NB classifier.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Method</th>
<th>Terms number</th>
<th>Highest $F_1$</th>
<th>Method</th>
<th>Terms number</th>
<th>Lowest $F_1$</th>
<th>Method</th>
<th>Terms number</th>
<th>Lowest $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU1</td>
<td>RDTFD</td>
<td>600</td>
<td>0.971</td>
<td>GININTF</td>
<td>200</td>
<td>0.865</td>
<td>RDTFD</td>
<td>200</td>
<td>0.960</td>
</tr>
<tr>
<td>PU2</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.939</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.706</td>
<td>RDTFD</td>
<td>200</td>
<td>0.904</td>
</tr>
<tr>
<td>PU3</td>
<td>RDTFD/IMTFIDF</td>
<td>1000</td>
<td>0.967</td>
<td>GININTF</td>
<td>200</td>
<td>0.788</td>
<td>RDTFD</td>
<td>200</td>
<td>0.953</td>
</tr>
<tr>
<td>PUA</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.973</td>
<td>GININTF</td>
<td>200</td>
<td>0.875</td>
<td>RDTFD</td>
<td>200</td>
<td>0.954</td>
</tr>
<tr>
<td>CS</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.954</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.713</td>
<td>TFIG</td>
<td>200</td>
<td>0.906</td>
</tr>
<tr>
<td>ES</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.967</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.754</td>
<td>CHI</td>
<td>200</td>
<td>0.938</td>
</tr>
</tbody>
</table>

Table 6: Performance of feature selection method in SVM classifier.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Method</th>
<th>Terms number</th>
<th>Highest $F_1$</th>
<th>Method</th>
<th>Terms number</th>
<th>Lowest $F_1$</th>
<th>Method</th>
<th>Terms number</th>
<th>Lowest $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU1</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.977</td>
<td>TTFS</td>
<td>200</td>
<td>0.925</td>
<td>RDTFD</td>
<td>200</td>
<td>0.966</td>
</tr>
<tr>
<td>PU2</td>
<td>RDTFD</td>
<td>400</td>
<td>0.938</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.792</td>
<td>RDTFD</td>
<td>200</td>
<td>0.913</td>
</tr>
<tr>
<td>PU3</td>
<td>RDTFD/IMTFIDF</td>
<td>1000</td>
<td>0.970</td>
<td>TTFS</td>
<td>200</td>
<td>0.943</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.955</td>
</tr>
<tr>
<td>PUA</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.979</td>
<td>GININTF</td>
<td>200</td>
<td>0.915</td>
<td>RDTFD</td>
<td>200</td>
<td>0.963</td>
</tr>
<tr>
<td>CS</td>
<td>RDTFD</td>
<td>800</td>
<td>0.983</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.926</td>
<td>TFIG</td>
<td>200</td>
<td>0.951</td>
</tr>
<tr>
<td>ES</td>
<td>RDTFD</td>
<td>1000</td>
<td>0.971</td>
<td>IMTFIDF</td>
<td>200</td>
<td>0.890</td>
<td>TFIG</td>
<td>200</td>
<td>0.947</td>
</tr>
</tbody>
</table>

Table 7: The distribution of particles corresponding to the ratio in different terms number by using NB classifier.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Terms number</th>
<th>Highest $F_1$</th>
<th>Ratio</th>
<th>Terms number</th>
<th>Lowest $F_1$</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU1</td>
<td>600</td>
<td>0.971</td>
<td>0.216</td>
<td>200</td>
<td>0.960</td>
<td>0.236</td>
</tr>
<tr>
<td>PU2</td>
<td>1000</td>
<td>0.939</td>
<td>0.376</td>
<td>200</td>
<td>0.907</td>
<td>0.447</td>
</tr>
<tr>
<td>PU3</td>
<td>1000</td>
<td>0.967</td>
<td>0.472</td>
<td>200</td>
<td>0.953</td>
<td>0.504</td>
</tr>
<tr>
<td>PUA</td>
<td>1000</td>
<td>0.973</td>
<td>0.633</td>
<td>200</td>
<td>0.954</td>
<td>0.624</td>
</tr>
<tr>
<td>CS</td>
<td>1000</td>
<td>0.954</td>
<td>0.780</td>
<td>200</td>
<td>0.903</td>
<td>0.728</td>
</tr>
<tr>
<td>ES</td>
<td>1000</td>
<td>0.967</td>
<td>0.952</td>
<td>200</td>
<td>0.897</td>
<td>0.985</td>
</tr>
</tbody>
</table>
less than some methods based on exhaustive search and greedy search, it is still more time consuming than the methods based on feature weighting ranking.

As mentioned Section 3.2, the feature selection window will select \( n \) terms in \((2 \times n)\) terms as terms for spam filtering. Therefore, in this experiment, we select term subsets ranging from 200 to \( n \) terms with the step of 200 terms between \((2 \times 200)\) and \((2 \times n)\) terms, which finally generate \( Ft = (n/200) \) candidate term sets. Moreover, the particle number and iteration times in PSO algorithm (Algorithm 2) will affect the running time of RDTFD method, for instance, the iteration times will decrease as the terms number decreases, and thus, we can consider adjusting these parameters. In addition, for a unifying experiment standard, the particle number and iteration times are set to 30, which means the \((30 \times 30 \times Ft)\) candidate term subsets will be generated. For instance, when the term subsets range from 200 to 1000 terms with the step of 200 terms, we will select \( Ft = (1000/200) = 5 \) candidate term subsets between \((2 \times 200)\) and \((2 \times 1000)\) terms, and thus, altogether \((30 \times 30 \times 5 = 4500)\) candidate term subsets are generated and the size of each candidate subsets will increase as the terms number increases.

Consequently, the running time of RDTFD method is affected by the number of the selected terms on the specific samples. Table 9 shows that the running time of RDTFD method will increase linearly with the increase of the samples and the number of the selected terms when particle number and iteration times are constant.


When the two sets of paired data approximately obey the normal distribution, the paired \( t \)-test can be used. Otherwise, we can use Wilcoxon signed ranks test (Wilcoxon, 1945) to replace the paired \( t \)-test. Moreover, Wilcoxon signed ranks test method not only considers the positive and the negative differences, but also ranks the differences in performances of two classifiers for each dataset, and thus, it is more sensible than the \( t \)-test [48]. In this experiment, NB and SVM classifiers are applied to evaluate the performance of RDTFD method on six datasets, such as PU1, PU2, PU3, PUA, CS, and ES, and Wilcoxon signed ranks test is applied on the paired \( F1 \) values by the RDTFD method and the other six feature selection methods, such as CHI, IG, TFIG, TTFS, IMTFIDF, and GININTF. In addition, the terms range

<table>
<thead>
<tr>
<th>Method</th>
<th>NB</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHI</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>IG</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>TFIG</td>
<td>0.085</td>
<td>0.005</td>
</tr>
<tr>
<td>TTFS</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>IMTFIDF</td>
<td>0.412</td>
<td>0.005</td>
</tr>
<tr>
<td>GININTF</td>
<td>0.005</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Consequently, the running time of RDTFD method is affected by the number of the selected terms on the specific samples. Table 9 shows that the running time of RDTFD method will increase linearly with the increase of the samples and the number of the selected terms when particle number and iteration times are constant.
Acknowledgments

This work was supported by the Science Foundation of Jiangsu Province, China (BK20170566) and the National Natural Science Foundation of China (11703009). The authors also would like to thank the developers and express their gratitude to the donors and maintainers of the different datasets.

References

Information and Knowledge Management, pp. 1482–1486, Maui, HI, USA, October 2012.


