Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6076272, 14 pages
https://doi.org/10.1155/2020/6076272

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 Su[3]

[1]*Department of Computer, University of Sichuan, Chengdu, Sichuan Province 610065, China*
[2]*Information Center, Nanjing Jiangbei People's Hospital, Nanjing, Jiangsu Province 210048, China*
[3]*Engineering 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

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 conduced 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 (TTFS) [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 TTFS 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 excessively 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 selection 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 selection 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 performance 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 information 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 measurement is defined to be

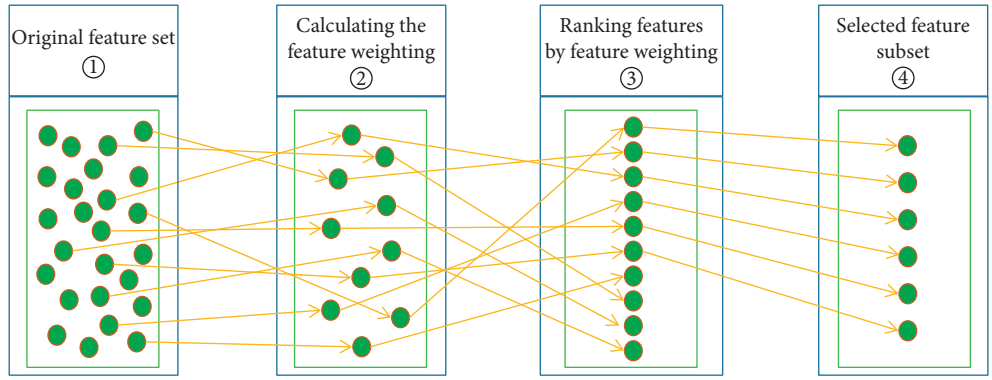

Figure 1: Process diagram of the general feature selection methods. Note that the green circle represents each feature in feature set.

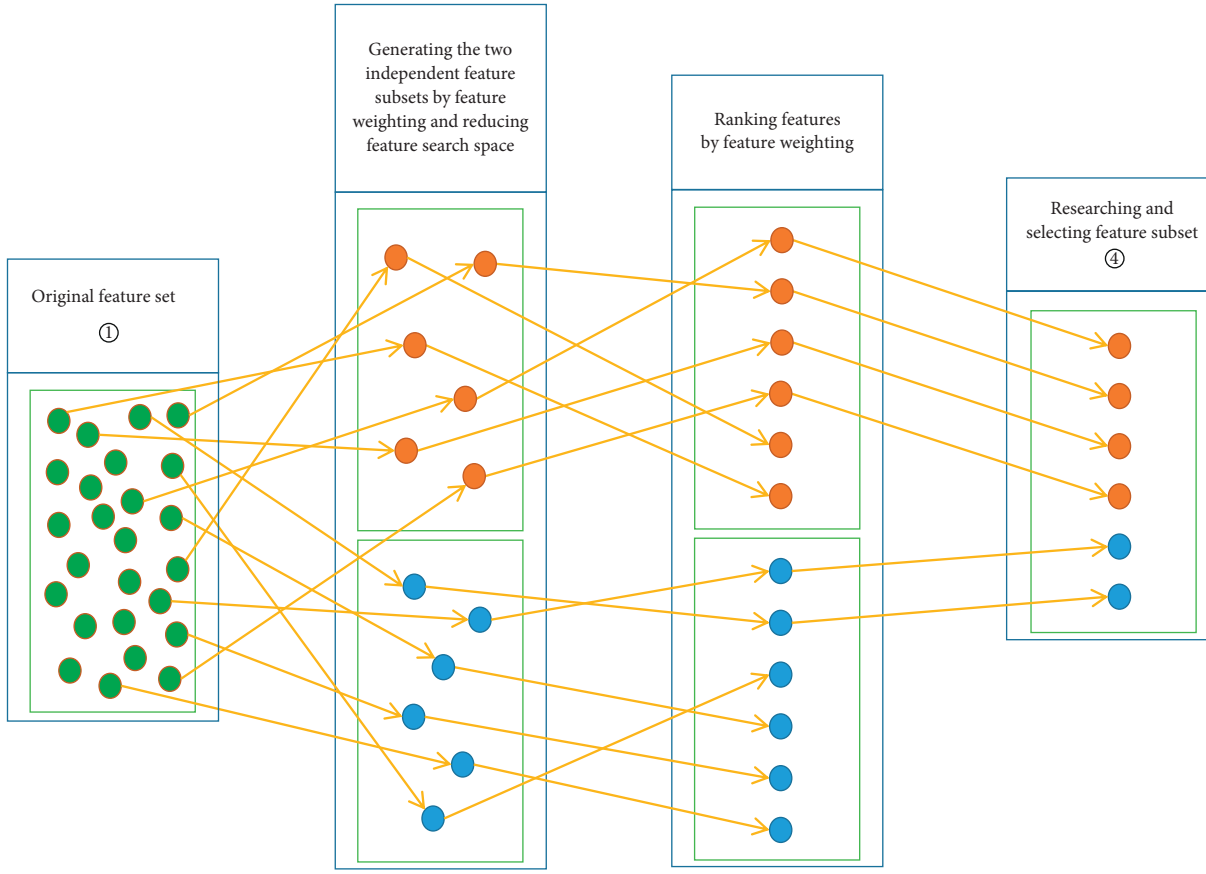

Figure 2: Process diagram of the RDTFD method. Note that the green circle represents each features in features set, the orange circle represents each feature of the positive features, and the blue circle represents each feature of the negative features.

$$\chi^2(t_i) = \max_{c_k \in \{s,h\}} \left\{ \chi^2(t_i, c_k) \right\},$$

$$\chi^2(t_i, c_k) = \frac{N(a_{ik}d_{ik} - c_{ik}b_{ik})^2}{(a_{ik} + c_{ik})(b_{ik} + d_{ik})(a_{ik} + b_{ik})(c_{ik} + d_{ik})}, \tag{1}$$

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 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

absence. The information gain of the term t_i is defined as follows [10]:

$$\begin{aligned}\mathrm{IG}(t_i) = &-\sum_{c_k\in\{s,h\}} p(c_k)\log(p(c_k)) + p(t_i)\sum_{c_k\in\{s,h\}} p\\ &\cdot\left(c_k\mid t_i\right)\log p\left(c_k\mid t_i\right) + p(\overline{t_i})\sum_{c_k\in\{s,h\}} p\left(c_k\mid \overline{t_i}\right)\log p\\ &\cdot\left(c_k\mid \overline{t_i}\right),\end{aligned} \tag{2}$$

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 $p(\overline{t_i})$ are the probability that a document contains or does not contain t_i, and $p(c_k\mid t_i)$ and $p(c_k\mid \overline{t_i})$ 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]:

$$\begin{aligned}\mathrm{TFIG}(t_i) = &-\sum_{c_k\in\{s,h\}} p(c_k)\log(p(c_k)) + p(\overline{t_i})\sum_{c_k\in\{s,h\}} p\\ &\cdot\left(c_k\mid \overline{t_i}\right)\log p\left(c_k\mid \overline{t_i}\right) + p(t_1)\sum_{c_k\in\{s,h\}} p\\ &\cdot\left(c_k\mid t_1\right)\log p\left(c_k\mid t_1\right) + c\times p(t_2)\sum_{c_k\in\{s,h\}} p\\ &\cdot\left(c_k\mid t_2\right)\log p\left(c_k\mid t_2\right),\end{aligned} \tag{3}$$

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

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:

$$\mathrm{GINI}(t_i) = p\left(t_i\mid s\right)^2\times p(s\mid t_i)^2 + p\left(t_i\mid h\right)^2\times p(h\mid t_i)^2, \tag{4}$$

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

$$\mathrm{TTFS}(t_i, c_k) = \frac{\left|\overline{tf_{ki}} - \overline{tf_i}\right|}{m_k\times s_i},$$

$$S_i^2 = \frac{1}{N-K}\sum_{k=1}^{k}\sum_{j\in c_k}\left(tf_{ij} - \overline{tf_{ki}}\right)^2, \tag{5}$$

$$m_k = \sqrt{\frac{1}{N_k} - \frac{1}{N}},$$

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, $\overline{tf_{ki}}$ is the average term frequency of the term t_i in c_k, $\overline{tf_i}$ 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 θ, when $\mathrm{TTFS}(t_i, c_k) < \theta$, which shows that $\overline{tf_i}$ 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 TFIDF, 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:

$$\mathrm{IMTFIDF}\left(t_i, d_j, c_k\right) = tf_{ij}\times\log\left(\frac{N}{K_i}\right),\quad \text{if } \frac{M_i}{M_i + K_i} > 70\%, \tag{6}$$

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_i is the number of e-mails which contain term t_i 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 $(M_i/(M_i + K_i))$ 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:

$$\mathrm{GINI}_{\mathrm{NTF}}(t_i) = \sum_{c_k\in\{s,h\}}\left(\frac{A(t_i, c_k)}{M_{c_k}}\right)^2\left(\frac{A(t_i, c_k)}{A(t_i, c_k) + B(t_i, c_k)}\right)^2, \tag{7}$$

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 TFFS 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:

$$\mathrm{RTDFD} = \frac{\log\left(df_{t_i \in \mathrm{ham}}\right)}{\log(\mathrm{DNH})} \times \frac{\log\left(tf_{t_i \in \mathrm{ham}}\right)}{\log(\mathrm{TNH})} - \frac{\log\left(df_{t_i \in \mathrm{spam}}\right)}{\log(\mathrm{DNS})} \times \frac{\log\left(tf_{t_i \in \mathrm{spam}}\right)}{\log(\mathrm{TNS})}, \tag{8}$$

$$\text{if } \mathrm{MAX}\left(\frac{df_{t_i \in \mathrm{ham}}}{df_{t_i \in \mathrm{ham}} + df_{t_i \in \mathrm{spam}}}, \frac{df_{t_i \in \mathrm{spam}}}{df_{t_i \in \mathrm{ham}} + df_{t_i \in \mathrm{spam}}}\right) > 0.7, \tag{9}$$

where $df_{t_i \in \mathrm{ham}}$ denotes the document frequency of term t_i in ham documents, $df_{t_i \in \mathrm{spam}}$ denotes the document frequency of term t_i in spam documents, $tf_{t_i \in \mathrm{ham}}$ denotes the term frequency of term t_i in ham documents, $tf_{t_i \in \mathrm{spam}}$ denotes that term frequency of term t_i in spam documents, DNH is the number of ham documents, DNS is the number of spam documents, TNH is the total number of terms in ham documents, and TNS 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 to 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.

Table 1: Definition related to frequency terms.

Keyword	Definition
Document frequency	The number of documents that contains the term in the dataset
Term frequency	The number of the term in all documents of the dataset
Average term frequency	The average number of the term in all documents of the dataset
Term frequency distribution	The number of the term in each document of the dataset

Table 2: Datasets examples of spam and ham.

Datasets	Spam		Ham	
	Document number	Total terms number	Document number	Total terms number
Dataset 1	5	1000	5	500
Dataset 2	10	5000	10	2000

Table 3: Document frequency and terms frequency of term examples in dataset 1 and dataset 2.

Datasets	Term	Document frequency		Term frequency		Average term frequency		Term frequency distribution	
		Spam	Ham	Spam	Ham	Spam	Ham	Spam	Ham
Dataset 1	Investment	5	3	5	3	1	0.6	1,1,1,1,1	0,1,1,0,1
	Linguistic	5	5	10	5	2	1	2,2,2,2,2	1,1,1,1,1
	Marketing	5	5	10	10	2	2	2,2,2,2,2	2,3,2,1,2
Dataset 2	Debian	10	10	20	20	2	2	2,2,2,2,2,2,2,2,2,2	2,2,2,2,2,2,2,2,2,2
	Mailings	10	1	20	10	2	1	2,2,2,2,2,2,2,2,2,2	10,0,0,0,0,0,0,0,0,0
	Workshop	10	10	20	20	2	2	2,2,2,2,2,2,2,2,2,2	1,3,1,2,3,1,3,2,1,3

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]:

$$v_{id}^{t+1} = w \times v_{id}^{t} + c_1 \times r_1 \times \left(p_{id}^{t} - x_{id}^{t}\right) + c_2 \times r_2 \times \left(p_{gd}^{t} - x_{id}^{t}\right),$$
$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t}, \quad d = 1, 2, \ldots, D. \tag{10}$$

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_n, 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_n, 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, $R_1 \longrightarrow R_m$ 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 7 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 $O(N^2)$ [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.

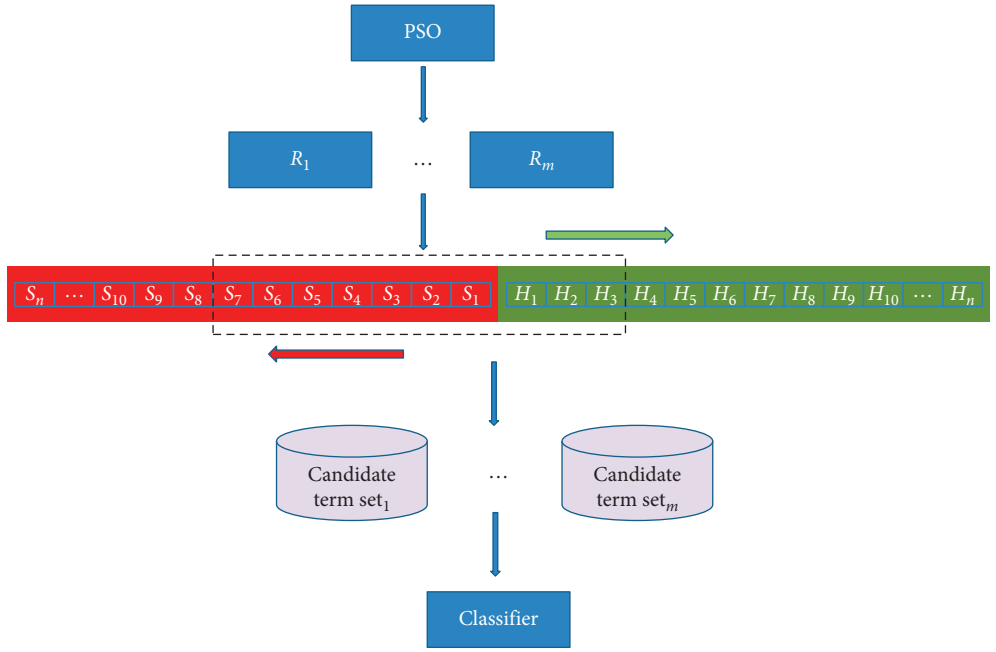

Figure 3: A feature selection window can select the optimal term subset by adjusting the ratio of spam terms number to terms number. Terms number includes terms number of spam and ham.

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}^t denotes the speed of particle i on the dth dimension at the iteration t, which represents the ratio of spam terms number to the terms number. p_{id}^t denotes the best position of individual particle i comparing with its current best fitness at iteration t. p_{gd}^t denotes the globally best position of all particles comparing with its current globally best fitness at iteration t. x_{id}^t denotes the position of particle i on the dth dimension at the iteration t. x_{id}^{t+1} denotes the new position of particle i on the dth 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}^t of particle are $V_{\max} = 0.01$ and $V_{\min} = 0.001$, respectively. The maximum and minimum x_{id}^t of particle are $X_{\max} = 0.999$ and $X_{\min} = 0.001$, respectively. The maximum iteration is set to $N_t = 30$, for each iteration t, W is given as follows:

$$w^{t+1} = w_{\max} - \frac{(w_{\max} - w_{\min})t}{N_t}. \tag{11}$$

3.3. Correlation Algorithm. Algorithms 1 and 2 outline how the RDTFD 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:

$$F1 = \frac{2 \times r \times p}{r + p}. \tag{12}$$

In equation (12), $r = (n_{ss}/(n_{ss} + n_{sh}))$, $p = (n_{ss}/(n_{ss} + n_{hs}))$, n_{ss} represents the number of spams that are correctly identified as spams, n_{sh} represents the number of spams that are identified as hams, n_{hs} 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:

$$F = \mathrm{EVAL}(Ft). \tag{13}$$

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 8 GB main memory. The feature selection methods run on the platform of windows7 Ultimate and python3.5.

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 *F*1 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* = 1000//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 *F*1 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).

4.2. 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 *F*1 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 *F*1 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 *F*1, which means the highest *F*1 value based on *N* terms and the feature selection methods, the third column part is Method/Terms Number/Lowest *F*1, which denotes the lowest *F*1 value based on *N* terms and the feature selection methods, and the fourth column is Method/Lowest terms Number/Highest *F*1, which represents the highest *F*1 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 *F*1 value of SVM based on SMO almost completely surpasses NB on seven datasets, the SVM is more suitable than NB in text classification.

Input: Fh and Fs denote term set of ham and spam, respectively. N denotes the number of candidate terms.
Output: Gs, Gt, Gr
Step 1: parameters initialization
(1.1) set $N_p = 30$//initialize the number of particles
(1.2) set $N_t = 30$//initialize the number of iteration times
(1.3) **for** $i = 1$ to N_p
(1.4) set $x_i^1 = \text{random}[x\min, x\max]$//initialize the position of each particle
(1.5) set $v_i^1 = \text{random}[v\min, v\max]$//initialize the velocity of each particle
(1.6) **end for**
(1.7) Set $r = v_i^1$//r is the ratios of spam terms number to the terms number
Step 2: main procedure
(2.1) set $t = 1$//initialize iterations
(2.2) **while** $t \le N_t$ **do**
(2.3) **for** $i = 1$ to N_p
(2.4) set $r = v_i^t$//update the ratios at t iterations
(2.5) $j = 0$//initialize the terms number for training
(2.6) set Fa = NULL//initialize terms Fa
(2.7) **while** $j < N$
(2.8) $j = j + 200$;
(2.9) $Ns = r \times j$//terms number of spam
(2.10) $Nh = j - Ns$//terms number of ham
(2.11) **for** $k = 0$ to $Nh - 1$
(2.12) put $Fh[k]$ into Fa
(2.13) **end for**
(2.14) **for** $k = 0$ to $Ns - 1$
(2.15) put $Fs[k]$ into Fa
(2.16) **end for**
(2.17) **end while**
(2.18) $F = \text{EVAL}(Fa)$//calculate the local $F1$ value according to equation (12)
(2.19) **end for**
(2.20) **if** $F > Gs$ **then**
(2.21) $Gs = F$//update Gs by the maximal $F1$ value
(2.22) $Gt = Fa$//update Gr by Fa in terms of the maximal $F1$ value
(2.23) $Gr = r$//update Gt by using r according to the maximal $F1$ value
(2.24) remove the particle of the lowest $F1$ value from particle swarm
(2.25) set $x_i^t = \text{random}[x\min, x\max]$
(2.26) set $v_i^t = \text{random}[v\min, v\max]$
(2.27) construct new particle and insert into the particle swarm
(2.28) **for** $i = 1$ to N_p
(2.29) calculate x_i^{t+1}//update position of particle according to equation (11)
(2.30) calculate v_i^{t+1}//update velocity of particle according to equation (11)
(2.31) **end for**
(2.32) $t = t + 1$//update iterations
(2.33) **end while**
(2.34) **return** Gs, Gt, Gr

Algorithm 2: PSO (to search the optimal term set).

Table 4: Mail datasets consist of PU1, PU2, PU3, PUA, CS, and ES.

Datasets	Ham	Spam
PU1	610	480
PU2	570	140
PU3	2310	1820
PUA	570	570
CS	2949	1378
ES	4012	1500

4.7. The 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 represents each dataset for experiment. The second column part has Terms Number/Highest $F1$/Ratio, which means the ratio of spam terms number to the terms number when the RDTFD method obtains the highest $F1$ values. The third column part has Terms Number/Lowest $F1$/Ratio, which means the ratio of spam terms number to the terms number when the RDTFD method obtains the lowest $F1$ values.

4.8. Analysis of Time Complexity. Compared with the general feature selection, although the RDTFD method takes much

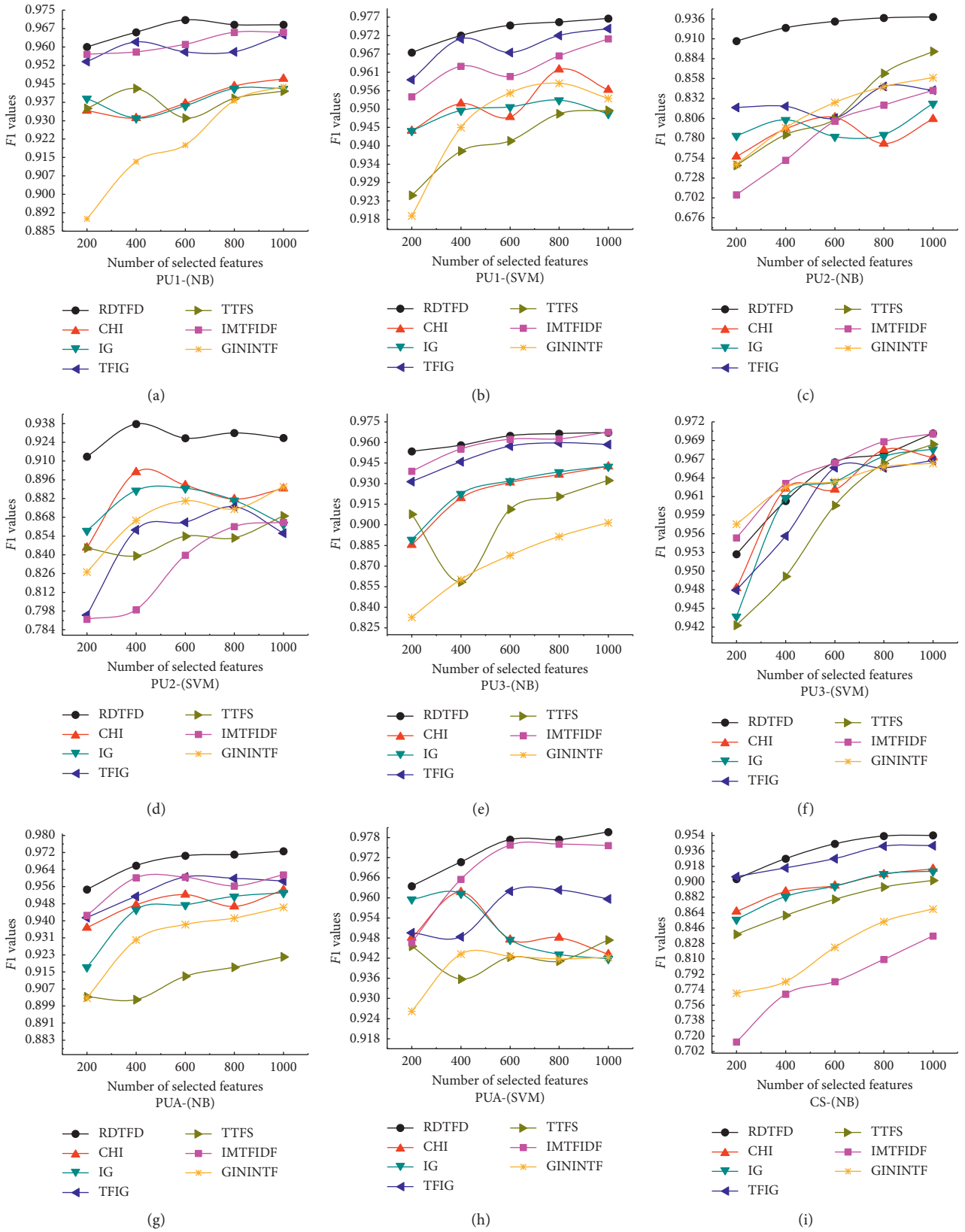

Figure 4: Continued.

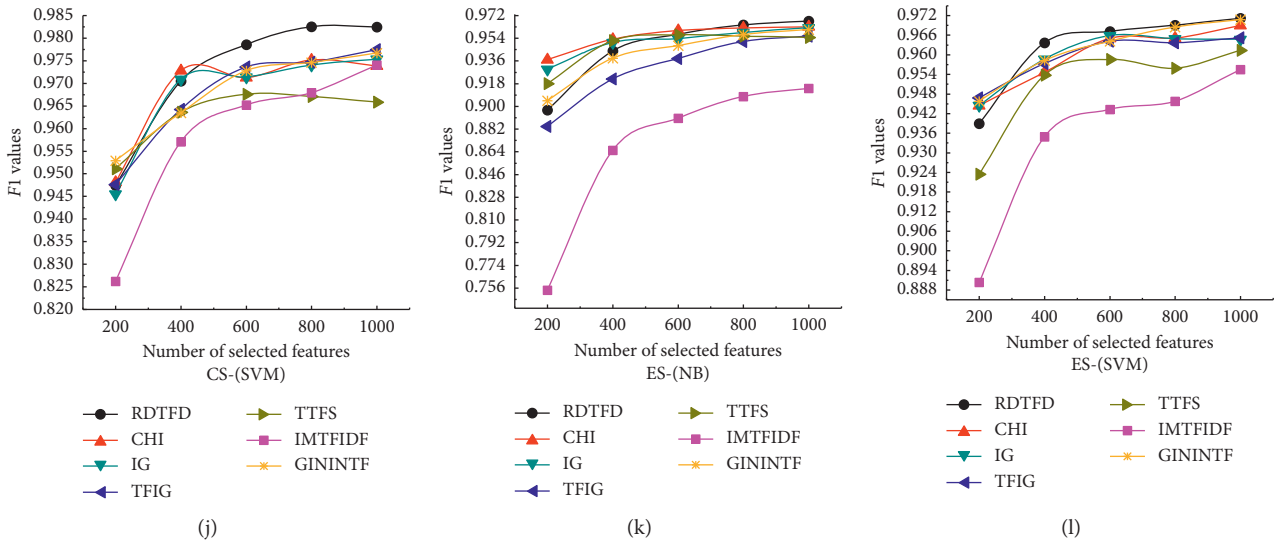

Figure 4: $F1$ values on six spam datasets.

Table 5: Performance of feature selection methods in NB classifier.

Datasets	Method	Terms number	Highest $F1$	Method	Terms number	Lowest $F1$	Method	Lowest terms number	Highest $F1$
PU1	RDTFD	600	0.971	GININTF	200	0.865	RDTFD	200	0.960
PU2	RDTFD	1000	0.939	IMTFIDF	200	0.706	RDTFD	200	0.904
PU3	RDTFD/ IMTFIDF	1000	0.967	GININTF	200	0.788	RDTFD	200	0.953
PUA	RDTFD	1000	0.973	GININTF	200	0.875	RDTFD	200	0.954
CS	RDTFD	1000	0.954	IMTFIDF	200	0.713	TFIG	200	0.906
ES	RDTFD	1000	0.967	IMTFIDF	200	0.754	CHI	200	0.938

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

Datasets	Method	Terms number	Highest $F1$	Method	Terms number	Lowest $F1$	Method	Lowest terms number	Highest $F1$
PU1	RDTFD	1000	0.977	TTFS	200	0.925	RDTFD	200	0.966
PU2	RDTFD	400	0.938	IMTFIDF	200	0.792	RDTFD	200	0.913
PU3	RDTFD/ IMTFID	1000	0.970	TTFS	200	0.943	IMTFIDF	200	0.955
PUA	RDTFD	1000	0.979	GININTF	200	0.915	RDTFD	200	0.963
CS	RDTFD	800	0.983	IMTFIDF	200	0.926	TFIG	200	0.951
ES	RDTFD	1000	0.971	IMTFIDF	200	0.890	TFIG	200	0.947

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

Datasets	Terms number	Highest $F1$	Ratio	Terms number	Lowest $F1$	Ratio
PU1	600	0.971	0.216	200	0.960	0.236
PU2	1000	0.939	0.376	200	0.907	0.447
PU3	1000	0.967	0.472	200	0.953	0.504
PUA	1000	0.973	0.633	200	0.954	0.624
CS	1000	0.954	0.780	200	0.903	0.728
ES	1000	0.967	0.952	200	0.897	0.985

Table 8: The distribution of particles corresponding to the ratio in different terms number by using SVM classifier.

Datasets	Terms number	Highest $F1$	Ratio	Terms number	Lowest $F1$	Ratio
PU1	1000	0.977	0.454	200	0.967	0.432
PU2	400	0.938	0.425	200	0.913	0.439
PU3	1000	0.970	0.595	200	0.953	0.615
PUA	1000	0.980	0.532	200	0.963	0.415
CS	800	0.983	0.565	200	0.947	0.248
ES	1000	0.971	0.724	200	0.939	0.377

Table 9: Running time of the RDTFD method on specific samples and terms number in NB and SVM.

Datasets	Samples	Terms number					Terms number				
		200	400	600	800	1000	200	400	600	800	1000
		Running time of NB (sec)					Running time of SVM (sec)				
PU1	1090	128	230	324	416	506	127	223	318	409	499
PU2	710	86	147	206	264	319	88	156	216	275	333
PU3	4130	464	843	1270	1645	2005	441	805	1164	1508	1844
PUA	1140	130	235	333	428	519	135	235	334	425	520
CS	4327	486	904	1330	1720	2109	479	852	1219	1615	1959
ES	5512	603	1132	1673	2184	2672	571	1051	1569	2086	2575

Table 10: Two-tailed Wilcoxon signed ranks test results by using NB and SVM.

Method	NB						SVM					
	PU1	PU2	PU3	PUA	CS	ES	PU1	PU2	PU3	PUA	CS	ES
CHI	0.005	0.005	0.005	0.005	0.005	**0.508**	0.005	0.005	0.017	0.005	0.017	0.028
IG	0.005	0.005	0.005	0.005	0.005	**0.185**	0.005	0.005	0.011	0.005	0.012	0.027
TFIG	**0.085**	0.005	0.005	0.005	0.007	0.005	0.005	0.005	0.011	0.005	0.008	**0.074**
TTFS	0.005	0.005	0.005	0.005	**0.113**	0.005	0.005	0.005	0.005	0.005	0.005	0.005
IMTFIDF	**0.412**	0.005	0.012	0.005	0.007	0.005	0.005	0.011	0.023	0.021	0.005	0.005
GININTF	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.012	0.005

less time 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 range from 200 to n terms with the step of 200 terms between (2×200) 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 (2×200) and (2×1000) 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.

4.9. Statistical Analysis. 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

from 200 to 2000 with the step of 200. Table 10 shows the p values when the significance level is at 5%, we apply IBM SPSS Statistics 23 software to observe the data, and experimental result shows that the performance of RDTFD method is obviously better than other feature selection methods in 66 cases ($p < 0.05$), but in 6 cases ($p > 0.05$), the RDTFD method has no significant difference than the other methods. Therefore, the RDTFD method based on independent feature space search is more robust than the other feature selection methods in spam filtering.

5. Conclusions and Future Work

In this paper, we propose a new feature selection method based on independent feature space search for text classification, which can be divided into two steps. Firstly, a relative document-term frequency difference (RDTFD) method is used to divide the features of text into two independent features subsets according to the ability to discriminate the ham and spam samples, which can improve the high correlation of feature class and reduce the correlation between features. Furthermore, the RDTFD method also reduces the search range of feature space and maintains appropriate feature redundancy. Secondly, the feature search strategy based on particle swarm optimization algorithm (PSO) is used to find the optimal feature space, which can improve the performance of text classification. Finally, we apply NB and SVM to evaluate RDTFD model method by using the $F1$ measure on six spam datasets such as PU123A, CS, and ES, respectively. Experiment result shows that, in most cases, the RDTFD method based on independent feature space search outperforms the others general feature selection methods.

In future work, we will try to use other feature search strategies to improve RDTFD method. In addition, we are expanding this study to improve the running speed of RDTFD, especially the parallel operation of large-scale PSO on terms of the high-dimensional feature space.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

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

[1] J. Shlens, "A tutorial on principal component analysis," 2014, https://arxiv.org/abs/1404.1100.

[2] A. J. Izenman, "Linear discriminant analysis," in *Modern Multivariate Statistical Techniques*, pp. 237–280, Publishing House, New York, NY, USA, 2013.

[3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[4] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proceedings of the 18th International Conference on Machine Learning*, vol. 1, pp. 74–81, Williamstown, MA, USA, June 2001.

[5] I. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza, "Filter versus wrapper gene selection approaches in DNA microarray domains," *Artificial Intelligence in Medicine*, vol. 31, no. 2, pp. 91–103, 2004.

[6] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Speeding up incremental wrapper feature subset selection with Naive Bayes classifier," *Knowledge-Based Systems*, vol. 55, pp. 140–147, 2014.

[7] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, pp. 1205–1224, 2004.

[8] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," in *Proceedings of the 20th International Conference on Machine Learning, ICML-2003*, pp. 856–863, Washington DC, USA, 2003.

[9] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in *Proceedings of the 11th International Conference on Information and Knowledge Management*, pp. 659–661, McLean, VA, USA, November 2002.

[10] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning*, pp. 412–420, Nashville, TN, USA, July 1997.

[11] Y. Xu and L. Chen, "Term-frequency based feature selection methods for text categorization," in *Proceedings of the 2010 4th International Conference on Genetic and Evolutionary Computing (ICGEC)*, IEEE, Shenzhen, China, pp. 280–283, December 2010.

[12] S. Shankar and G. Karypis, "A feature weighting adjustment algorithm for document categorization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000)*, Boston, MA, USA, Auguest, 2000.

[13] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Systems with Applications*, vol. 33, no. 1, pp. 1–5, 2007.

[14] N. Azam and J. Yao, "Comparison of term frequency and document frequency based feature selection metrics in text categorization," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4760–4768, 2012.

[15] C.-M. Chen, H.-M. Lee, and Y.-J. Chang, "Two novel feature selection approaches for web page classification," *Expert Systems with Applications*, vol. 36, no. 1, pp. 260–272, 2009.

[16] S. S. R. Mengle and N. Goharian, "Ambiguity measure feature-selection algorithm," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 5, pp. 1037–1050, 2009.

[17] D. Wang, H. Zhang, R. Liu et al., "Feature selection based on term frequency and T-test for text categorization," in *Proceedings of the 21st ACM International Conference on*

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

[18] G. Domeniconi, G. Moro, R. Pasolini et al., "A study on term weightinging for text categorization: a novel supervised variant of tf.idf," in *Proceedings of 4th International Conference on Data Management Technologies and Applications*, pp. 26–37, Colmar, Alsace, France, July, 2015.

[19] X. Wang, J. Cao, Y. Liu et al., "Text clustering based on the improved TFIDF by the iterative algorithm," in *Proceedings of 2012 IEEE Symposium on Electrical & Electronics Engineering (EEESYM)*, pp. 140–143, IEEE, Kuala Lumpur, Malaysia, June 2012.

[20] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: a review," *Data Class: Algorithms and Applications*, pp. 37–64, CRC Press, Boca Raton, FL, USA, 2014.

[21] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 185–205, 2015.

[22] M. Xiong, X. Fang, and J. Zhao, "Biomarker identification by feature wrappers," *Genome Research*, vol. 11, no. 11, pp. 1878–1887, 2001.

[23] K. Wrtniak and M. Woniak, "Combined bayesian classifiers applied to spam filtering problem," in *Proceedings of 2012 International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions*, Springer, Ostrava, Czech Republic, pp. 253–260, September 2012.

[24] D. D. Arifin and M. A. Shaufiah, "Enhancing spam detection on mobile phone short message service (SMS) performance using FP-growth and naive Bayes classifier," in *Proceedings of 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 80–84, IEEE, Bandung, Indonesia, September 2016.

[25] A. Sharma and A. Suryawanshi, "A novel method for detecting spam email using KNN classification with spearman correlation as distance measure," *International Journal of Computer Applications*, vol. 136, no. 6, pp. 28–35, 2016.

[26] A. Nosseir, K. Nagati, and I. Taj-Eddin, "Intelligent word-based spam filter detection using multi-neural networks," *International Journal of Computer Science Issues*, vol. 10, pp. 17–21, 2013.

[27] A. Barushka and P. Hajek, "Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks," *Applied Intelligence*, vol. 48, no. 10, pp. 3538–3556, 2018.

[28] G. Caruana, M. Li, and Y. Liu, "An ontology enhanced parallel SVM for scalable spam filter training," *Neurocomputing*, vol. 108, pp. 45–57, 2013.

[29] M. Diale, D. W. C. Van, T. Celik, and A. Modupe, "Feature selection and support vector machine hyper-parameter optimisation for spam detection," in *Proceedings of 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, November 2016.

[30] A. Ali and Y. Xiang, "Spam classification using adaptive boosting algorithm," in *Proceedings of 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, July 2007.

[31] F. Akbari and H. Sajedi, "SMS spam detection using selected text features and boosting classifiers," in *Proceedings of 2015 7th Conference on Information and Knowledge Technology (IKT)*, May 2015.

[32] B. Zhou, Y. Yao, and J. Luo, "Cost-sensitive three-way email spam filtering," *Journal of Intelligent Information Systems*, vol. 42, no. 1, pp. 19–45, 2014.

[33] V. Basto-Fernandes, I. Yevseyeva, J. R. Méndez et al., "A spam filtering multi-objective optimization study covering parsimony maximization and three-way classification," *Applied Soft Computing*, vol. 48, pp. 111–123, 2016.

[34] K. Jain and S. Agrawal, "A hybrid approach for spam filtering using support vector machine and artificial immune system," in *Proceedings of 2014 First International Conference on Networks & Soft Computing*, Auguest 2014.

[35] M. L. McHugh, "The chi-square test of independence," *Biochemia Medica*, vol. 23, pp. 143–149, 2013.

[36] C. D. Manning and H. Schtze, "Pearson's chi-square test," in *Foundations of Statistical Natural Language Processing*, pp. 170-171, Publishing House, Boston, MA, USA, 1999.

[37] K. W. Church and R. L. Mercer, "Introduction to the special issue on computational linguistics using large corpora," *Computational Linguistics*, vol. 19, pp. 1–24, 1993.

[38] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Publishing House, Boston, MA, USA, 2011.

[39] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[40] Y.. Kong and J. Zhao, "Learning to filter unsolicited commercial e-mail," in *Proceedings of International Conference on Industrial Technology and Management (ICITM 2012)*, pp. 267–272, Singapore, 2012.

[41] R. Shams and R. E. Mercer, "Classifying spam emails using text and readability features," in *Proceedings of 2013 IEEE 13th international conference on data mining*, December 2013.

[42] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.

[43] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, pp. 41–48, Madison, WI, USA, 1998.

[44] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, vol. 29, no. 1, pp. 63–92, 2008.

[45] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive Bayes-which naive Bayes?" in *Proceedings of 3rd Conference on Email and Anti-Spam*, vol. 17, pp. 28–69, Mountain View, CA, USA, July 2006.

[46] J. Platt, "Sequential minimal optimization: a fast algorithm for training support vector machines," Microsoft Research Technical Report, Microsoft Research, Redmond, WA, USA, 1998.

[47] Y. Zhu and Y. Tan, "A local-concentration-based feature extraction approach for spam filtering," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 486–497, 2011.

[48] J. Demar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2007.