A Framework of Centroid-Based Methods for Text Categorization

Dandan WANG†, Member, Qingcai CHEN†(a), and Xiaolong WANG†, Nonmembers

SUMMARY Text Categorization (TC) is a task of classifying a set of documents into one or more predefined categories. Centroid-based method, a very popular TC method, aims to make classifiers simple and efficient by constructing one prototype vector for each class. It classifies a document into the class that owns the prototype vector nearest to the document. Many studies have been done on constructing prototype vectors. However, the basic philosophies of these methods are quite different from each other. It makes the comparison and selection of centroid-based TC methods very difficult. It also makes the further development of centroid-based TC methods more challenging. In this paper, based on the observation of its general procedure, the centroid-based text classification is treated as a kind of ranking task, and a unified framework for centroid-based TC methods is proposed. The goal of this unified framework is to classify a text via ranking all possible classes by document-class similarities. Prototype vectors are constructed based on various loss functions for ranking classes. Under this framework, three popular centroid-based methods: Rocchio, Hypothesis Margin Centroid and DragPushing are unified and their details are discussed. A novel centroid-based TC method called SLRCM that uses a smoothing ranking loss function is further proposed. Experiments conducted on several standard databases show that the proposed SLRCM method outperforms the compared centroid-based methods and reaches the same performance as the state-of-the-art TC methods.

key words: text categorization, centroid-based methods, smoothing listwise ranking centroid method, a unified framework

1. Introduction

Text categorization (TC) is a critical task in text mining and information retrieval. It has been widely used in many applications such as text indexing, document sorting, text filtering, and web page categorization[1]. In recent years, TC has received more attention due to a large number of texts available on the Internet. TC is the task of classifying documents to one or more predefined categories based on contents [2], [3]. The general architecture of text categorization is usually divided into five parts: pre-processing, feature selection, feature weighting, classification and evaluation [4]. A growing number of machine learning techniques have been applied to TC and had been proved to be successful [5], which include Centroid-based methods [6]–[8], Decision Trees [9], [10], Neural Network [11], Naive Bayes [12], K-Nearest Neighbor (KNN) [13], [14], and Support Vector Machines (SVM) [15]–[17] etc. Among them, SVM has been shown superior than other methods by Sebastiani [5], Kotsiantis [18] and Wang [4].

Among above TC methods, centroid-based methods have attracted more and more attentions for their two advantages. One is that centroid-based methods are usually more efficient than other methods since their computational complexity is linear in the training phase [8]. The other advantage is that, compared with other models, the classification model of centroid-based method is relatively simpler while achieving acceptable accuracy [19]. The basic idea of centroid-based methods is to construct one prototype vector (centroid) for each class during the training phase and then classify a document into the class that owns the nearest prototype vector [20]. The popular centroid-based methods include Rocchio formula, average formula, sum formula and normalized sum formula [21]. However, the traditional centroid-based methods often suffer from model misfitting, i.e., a document belonging to A class gets higher similarity value with class B [22], [23], which causes the worse quality of initial prototype vectors [6]. To overcome misfitting, numerous iterative approaches had been proposed to optimize the prototype vectors, which include DragPushing [8], Hypothesis Margin [24], and pairwise optimized Rocchio algorithm [7]. These approaches have demonstrated that centroid-based methods can achieve good performance that is comparable to the performance of more complex approaches, such as SVM.

Though numerous centroid-based TC methods have been proposed, the basic philosophies of them are usually quite different from each other, which makes the comparison and selection of these methods very difficult. It also makes the further development of centroid-based TC methods more challenging. The most important issue is that the performance of exiting centroid-based methods is still inferior to the state-of-the-art methods.

To address above issues, this paper firstly proposes a unified framework for centroid-based TC methods. This framework provides them a clear view by treating centroid-based text categorization as a ranking task. The basic idea is to construct prototype vectors based on various ranking loss functions. The merit of this framework is that we can make full use of numerous ranking techniques developed for information retrieval to refine prototype vectors for centroid-based TC method. In this paper, we illustrate the representation of three popular centroid-based methods by using the unified framework. Under this unified framework, a new centroid-based method called SLRCM (Smoothing Listwise Ranking Centroid Method) is proposed. The experiments conducted on several standard corpora show that
SLR CM outperforms all compared centroid-based methods and reaches the performance of the state-of-the-art TC techniques.

The remainder of this paper is organized as follows. Section 2 introduces related work on centroid-based techniques. Section 3 proposes a unified framework of centroid-based methods. Three popular centroid-based methods are unified under our framework and a novel method called SLR CM is set forth. Experiment settings and experimental results are presented and discussed in Sect. 4. Section 5 draws our conclusion and outlines the future work.

2. Related Work

Text categorization is to classify each document into a predefined category. The study of text categorization should be traced back to the early 1960s [25]. It has been successfully used in many applications such as information filtering, text indexing and document sorting [1].

In the past decade, due to the simplicity and efficiency, centroid-based methods have become more and more popular. Tan [8] presented DragPushing method to iteratively refine the prototype vectors. It takes advantage of misclassified samples to drag the centroid of correct class and push the centroid of incorrect class. Tan [26] also used Error-Correcting Output Codes (ECOC) to improve prototype vectors and applied Model-Refinement to reduce inductive bias gradually. Tan [24] took into account the training-set errors and margins in the training phase so as to obtain an appropriate centroid classification model. Guan [6] designed a Class-Feature-Centroid classifier which combines inter-class feature with inner-class feature to represent prototype vectors. Miao [7] introduced a pairwise optimized Rocchio algorithm, which adjusts the prototype position between pairs of classes. Lertnattee [27] explored the effect of term distributions and clusters within a negative class to improve the performance of a centroid-based binary classifier. Lertnattee [28] investigated the effectiveness of several commonly used normalization functions and proposed a new type of class normalization, called term-length normalization, in centroid-based text categorization. Eui-Hong [29] presented a simple linear-time centroid-based document classification algorithm that uses a similarity function to account for the term similarity between the test document and the documents in the class, as well as for the dependencies between the terms in these documents.

Recently, researchers have speculated that information retrieval and classification technique are more inseparable in practice. Class information can improve relevance ranking problems and vice versa. For example, Bennett [30] showed that classifying web pages can improve the precision of a ranking returned by the search engine. Gabrilovich [31] built a system to classify search queries for commercial web search engine. Elissieff [32] proposed a ranking based SVM to deal with multi-label issues and reached positive results on Yeast gene functional classification problem.

3. The Framework of Centroid-Based Methods

Consider an \( m \) class text classification problem where \( m \geq 2 \). There are \( n \) training samples \((x_1^d, y_1), (x_2^d, y_2), \ldots, (x_n^d, y_n)\) with \( n \) corresponding labels \((y_1, y_2, y_3, \ldots, y_n)\). Each sample \( x_j^d \) has a \( d \) dimensional feature vector \((x_{j1}, x_{j2}, \ldots, x_{jd})\) and corresponding labeled class \( j \in \{1, 2, 3, \ldots, m\} \). Each class \( j \) has a centroid \( c_j \) with \( d \) dimensional feature vector \((c_{j1}, c_{j2}, c_{j3}, \ldots, c_{jd})\). This classification task can be converted to a ranking task in web search, which regards each sample as the query and all centroids as the corresponding hit documents. Take the sample \( x_k^d \) as an example, the goal of classifying \( x_k^d \) correctly can be regarded as training to put \( c_i \) at the first rank and other centroids rather than \( c_j \) unordered below.

Like the web search, each centroid has a relevance score used for ranking with the sample \( x_k^d \). The score is calculated by a relevance model with unknown parameters. For a general centroid-based classifier, the score is the inner-product similarity between the sample and centroid. For example, the similarity between \( x_k^d \) and \( c_i \) denoted by \( s_{kj}^i \), is calculated by formula (1) as follows:

\[
 s_{kj}^i = \langle x_k^d, c_i \rangle = \frac{(x_k^d)_n \cdot (c_i)_n}{\|x_k^d\| \cdot \|c_i\|}
\]  

(1)

Where \((x_k^d)_n\) and \((c_i)_n\) represent the normalized feature vector of \( x_k^d \) and \( c_i \) respectively. Here,

\[
(x_k^d)_n = \left( \frac{x_{k1}^d}{\sqrt{\sum_{m=1}^{d} (x_{km}^d)^2}}, \frac{x_{k2}^d}{\sqrt{\sum_{m=1}^{d} (x_{km}^d)^2}}, \ldots, \frac{x_{kd}^d}{\sqrt{\sum_{m=1}^{d} (x_{km}^d)^2}} \right)
\]  

(2)

\[
(c_i)_n = \left( \frac{c_{i1}}{\sqrt{\sum_{m=1}^{d} (c_{im})^2}}, \frac{c_{i2}}{\sqrt{\sum_{m=1}^{d} (c_{im})^2}}, \ldots, \frac{c_{id}}{\sqrt{\sum_{m=1}^{d} (c_{im})^2}} \right)
\]  

(3)

The unknown parameters of the relevance model are features that composed of the centroid vectors. Thus to get a better group of centroids, we should optimize model parameters by choosing proper ranking loss function that is used as the criterion for measuring the closeness between predicted and target ranking list. In following parts of this section, we analyze three popular centroid-based classifiers and show that they can be strongly connected by the ranking loss function.

3.1 The Rocchio Method

The Rocchio method [5] gets the centroid of each class by formula (4) as follows:

\[
 w_j' = \alpha w_j + \beta \frac{\sum_{j \neq j} x_j^d}{n_j} - \gamma \frac{\sum_{j \neq j} x_j^d}{n - n_j}
\]  

(4)

\( w_j' \) is the new weight of the class \( j \). In this formula, \( \alpha \) controls the variance of weight, \( \beta \) controls the balance between the weight of the other classes and the weight of the class \( j \), and \( \gamma \) controls the inductive bias of the model.
Where \( w_j \): The new iteration-step vector of centroid \( j \).
\( w'_j \): The current iteration-step vector of centroid \( j \).
\( n_j \): The number of positive samples belonging to class \( j \).

\( n \): The total number of samples.

Let’s consider about the ranking loss function given by formula (5) as follows:

\[
\text{loss}(x^i_j) = - \left( s^i_j - \sum_{j'} s^i_{k,j'} \right)
\]  

(5)

Where loss(\( x^i_j \)): The loss of document \( x^i_j \).

\( s^i_{k,j} \): The similarity score between document \( x^i_j \) and class centroid \( c_j \).

\( s^i_{k,j} \): The similarity score between document \( x^i_k \) and class centroid \( c_i \).

Calculating the gradient of the loss function, we can get formula (6) and (7) as follows:

\[
\frac{\partial \text{loss}(x^i_j)}{\partial c_j} = -s^i_k
\]  

(6)

\[
\frac{\partial \text{loss}(x^i_j)}{\partial c_i} = s^i_j
\]  

(7)

From formula (6) and (7), we can see that document \( x^i_k \) makes a gradient of \( -x^i_k \) and \( x^i_j \) contribution to class \( j \) and other classes respectively. By the gradient descent method and batch-mode update of training, we can make a conclusion that the Rocchio method is based on the ranking loss function described by formula (5). This loss function just roughly expresses that each document should be closed to the target class and far away from other classes. It puts little emphasis on the ranking position of the target class. For instance, assume class number \( m = 5 \), two samples \( x^i_1 \) and \( x^i_2 \) have the initial predicted ranking list for classes. Their similarity scores with respective classes are shown by Table 1.

| Similarity | \( x^i_1 \) | \( x^i_2 \) |
|------------|-------------|-------------|
| \( c_1 \)  | 0.6         | 0.6         |
| \( c_2 \)  | 0.5         | 0.5         |
| \( c_3 \)  | 0.4         | 0.1         |
| \( c_4 \)  | 0.3         | 0.1         |
| \( c_5 \)  | 0.2         | 0.1         |

Table 1: Similarities of \( x^i_1 \), \( x^i_2 \) with each centroid.

By formula (5), the loss of \( x^i_1 \) in Table 1 is equal to \(- (s^i_{11} - \sum_{j \neq 1} s^i_{k,j}) = - (0.6 - (0.5 + 0.4 + 0.3 + 0.2)) = 0.8,\) which is greater than the loss of \( x^i_2 \) equal to \(- (s^i_{22} - \sum_{j \neq 2} s^i_{k,j}) = - (0.5 - (0.6 + 0.1 + 0.1 + 0.1)) = 0.4.\) However, the class ranking list of \( x^i_2 \) is better than which of \( x^i_1 \).

### 3.2 Hypothesis Margin Centroid Method

Centroid classifier often suffers from the inductive bias or model misfit incurred by its assumption. Based on two training criterions, i.e., training-set errors and training-set margins, Hypothesis Margin Centroid method [24] is proposed to generate a refined centroid classification model. It uses the pairwise ranking loss function to get the best centroid vectors by formula (8) as follows:

\[
\text{loss}(x^i_k) = -(s^i_j - s^i_{k,j})
\]  

(8)

Where \( i \) represents the index of the centroid that is the most similar centroid to \( x^i_k \) with different label. Loss function of formula (8) is a simplified version of traditional pairwise ranking loss function. The loss function is not complete, i.e., it does not take advantage of other centroids to construct the pair; moreover, it is sample biased and is strongly affected by some samples, \( x^i_k \), which has a big distance between \( s^i_j \) and \( s^i_{k,j} \). For instance, assume that class number \( m = 5 \), samples \( x^i_{k1}, x^i_{k2}, x^i_{k3}, x^i_{k4} \) have the initial predicted classes ranking list and corresponding similarity scores shown by Table 2. By formula (8), the total loss of the five samples is 0.05 with four samples classified correctly. Just switching the first and second centroid position for each sample, we can see that the total loss becomes 0.05 with only one sample classified correctly.

### 3.3 DragPushing Method

DragPushing method [8] is an effective and efficient refinement strategy for rectifying the biases of centroid classifier. It refines the centroids with misclassified examples by dragging the centroid of a correct class towards a misclassified example and pushing the centroid of an incorrect class away from the misclassified example. Consider the following ranking loss function given by formula (9):

\[
\text{loss}(x^i_k) = -(s^i_j - \sum_{j \neq i} s^i_{k,j}) \text{ if } x^i_k \text{ is misclassified}
\]  

(9)

This loss function is similar to the loss function used by Rocchio method. The only difference is that it added a condition: only if the sample \( x^i_k \) is misclassified. It can make contributions to the update of centroids.

### 3.4 Smoothing Listwise Ranking Centroid Method

Unlike traditional ranking loss function, the ranking loss function for the classification task should put more emphasis on the position of the target class. So far as in this section,
we have shown the ranking loss functions used by three popular centroid-based classifiers. However, none of them pays enough attention to the position of the target class. Therefore, in this paper, we propose a centroid method called Smoothing Listwise Ranking Centroid Method (SLRCM) that uses a new smoothing ranking loss function given by formula (10) as follows:

\[
loss(x_k) = -\frac{1}{r^2}
\]  

(10)

Where \( \theta \) is a positive value. \( r \) represents the ranking position of the class which document \( x_k \) belongs to. It can be approximated by formula (11) as follows:

\[
r \approx 1 + \sum_{j \neq i} f(s_{kj}^j - s_{kj}^i)
\]  

(11)

Where \( f(\cdot) \) is the step function,

\[
f(x) = \begin{cases} 
0 & x > 0 \\
0.5 & x = 0 \\
1 & x < 0 
\end{cases}
\]  

(12)

The sigmoid function \( g(\cdot) \) can be adopted as a smooth approximation of the function \( f(\cdot) \):

\[
g(x) = \frac{1}{1 + \exp(\sigma \times x)} \quad \sigma > 0
\]  

(13)

Calculating the gradient of the ranking loss function given by formula (10), we can get formula (14) and (15) as follows:

\[
\frac{\partial loss(x_k^i)}{\partial c_{ji}} = \theta \times \frac{1}{r^{\rho+1}} \times \sigma \times x_j^i \\
\times \left( -\frac{\exp(\sigma \times (s_{kj}^j - s_{kj}^i))}{(1 + \exp(\sigma \times (s_{kj}^j - s_{kj}^i)))^2} \right)
\]  

(14)

\[
\frac{\partial loss(x_k^i)}{\partial c_{ji}} = -\theta \times \frac{1}{r^{\rho+1}} \times \sigma \times x_k^i \\
\times \left( -\frac{\exp(\sigma \times (s_{kj}^j - s_{kj}^i))}{(1 + \exp(\sigma \times (s_{kj}^j - s_{kj}^i)))^2} \right)
\]  

(15)

By gradient descent method, we can iteratively update the centroid feature vectors stochastically to get the best centroid by formula (16) and (17) as follows:

\[
c_{ji} = c_{ji} - \eta \times \frac{\partial loss(x_k^i)}{\partial c_{ji}}
\]  

(16)

\[
c_{ij} = c_{ij} - \eta \times \frac{\partial loss(x_k^i)}{\partial c_{ij}} \quad i \neq j
\]  

(17)

Where \( \eta \) represents the learning rate that controls the amount of update to the classifier. The training algorithm is given by Algorithm 1 as follows.

---

**Algorithm 1: SLRCM (Smoothing Listwise Ranking Centroid Method)**

**Input:** The number of training samples: \( N \);  
Training sample: \( x^1_i, x^2_i, \ldots, x^m_i \);  
Class number: \( m \);  
Class labels: \( c_1^e, c_2^e, \ldots, c_m^e \).

**Output:** Centroid lists of \( m \) classes.

**Procedure**

Calculate the initial centroid vectors for each class by averaging the samples’ vectors of each class;

for \( i = 1 \) to \( N \) do

for \( k = 1 \) to \( m \) do

Calculate the similarity between \( x^i_j \) and each class.

end

for \( k = 1 \) to \( m \) do

Calculate the ranking position of \( x^i_j \) by formula (9).

Calculate the common part of gradient from formula (14) and (15).

end

for \( p = 1 \) to \( d \) do

Update each dimensional feature of each centroid by formula (16) or (17).

end

end

---

4. Experiment

4.1 Datasets

**Reuters-21578**: The Reuters-21578 dataset is based on the Trinity College Dublin version. Trinity College Dublin changed the original SGML text documents into XML format text documents. Altogether there are 21578 documents in this 52-category corpus after removing all unlabeled documents and documents with more than one class labels. Since the distribution of documents over the 52 categories is highly unbalanced, we use the most populous 10 categories in our experiment [6]. It creates a dataset that contains 7289 documents.

**20-Newsgroup**: The 20-Newsgroup dataset is composed of 19997 articles from 20 Usenet discussion groups. This corpus is highly balanced (about one thousand articles per category). It has already become a very popular dataset in the research field of text classification and text clustering.

**OHSUMED**: OHSUMED corpus is a subset of the MEDLINE database. There are 23 cardiovascular diseases categories, which contain 348,566 medical abstracts from MeSH (Medical Subject Headings) categories during the years 1987 to 1991. We use a subset of OHSUMED dataset,
which is in total 23 categories and contains 34389 documents out of 50216 medical abstracts in the year 1991.

**Dataset Split:** In our experiments, three-fold cross validation method is used to evaluate the effectiveness of SLRCM. Each dataset is randomly split into three near equal size subsets. Two parts of them are for training and the remaining part for testing. We perform the training-test process three times and use the average of their performances as the final result. In this way, for Reuters-21578, the training set contains 5230 documents and the testing set contains 2059 documents. For 20-Newsgroup, 13296 documents are used for training and remaining 6667 documents for testing. For OHSUMED, 22937 documents are used for training and the remaining 11452 documents for testing.

4.2 Experimental Setting

**Preprocessing:** For Reuters-21578, 20-Newsgroup and OHSUMED, Lemur is used for etyma extraction. TF (term frequency), IDF (inverse document frequency), ICF (inverse category frequency) scores for feature weighting are extracted from the whole corpus. Stemming and stopword removal are applied.

**Feature Selection:** By contradistinguishing several feature selection methods, which include document frequency (DF), Information Gain (IG), Expected Cross Entropy (ECE) [33], Mutual Information (MI), $\chi^2$ (CHI) etc. ECE is finally selected as the feature selection method in our experiment and 4000 dimensional features are selected via this method.

**Feature Weighting:** By comparing the main existing term weighting methods such as TF-IDF [5], TF-ICF [34], [36] and TF-IDF-ICF [35], [36], TF-IDF-ICF is chosen as the term weighting method of this paper.

**Classification:** For the comparing Rocchio algorithm, the parameters are set as follows: $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.2$. For experiments involving the state-of-the-art text classification algorithm SVM, LibSVM† is used as the tool of SVM classification and the linear kernel and the default settings are applied. For Rocchio, DR, HMCM and SLRCM, the training set is split into four parts. Three parts are used for training and the remaining part is used for validation. In the process of training, the momentum is used for smoothing the updating of weights and suppressing cross-stitching by making non-radical revisions through the combination of gradient decreasing term with a fraction of previous weight change. Two heuristic rules are adopted to adjust the learning rate and momentum according to the cost function. One is reducing the learning rate and setting the momentum to zero while current loss of validation set has a certain percentage higher than the previous loss of validation set; the other is increasing the learning rate while current loss of validation set is not higher than the previous loss of validation set.

**Evaluation:** To evaluate the final performance, we use the most popular measures in text categorization which include F1, micro-average F1 and macro-average F1. F1 is defined as follows by formula (18).

$$F1 = \frac{2pr}{r+p} \tag{18}$$

where $p$ and $r$ represent precision and recall respectively. Here the Macro-F1 is computed by averaging the F1 measures of all categories, which puts emphasis on rare categories. The Micro-F1 is computed by the precision and recall of all documents, which puts emphasis on common categories.

4.3 Experimental Results

Several experiments are conducted with the proposed SLRCM algorithm. To provide a baseline for comparison, experimental results for Rocchio, HMCM, DR and SVM are also presented. The best performance comparison of different methods on Reuters-21578, 20-Newsgroup and OHSUMED are listed in Table 3 and Table 4.

| Dataset   | Reuters-21578 | 20-Newsgroup | OHSUMED |
|-----------|---------------|---------------|---------|
| Rocchio   | 0.8277        | 0.7202        | 0.4764  |
| HMCM      | 0.8733        | 0.8023        | 0.6202  |
| DR        | 0.9039        | 0.7565        | 0.4480  |
| SLRCM     | 0.9125        | 0.8259        | 0.6578  |
| SVM       | 0.9046        | 0.8272        | 0.6683  |
| **Improvement (%)** | 0.873 | -0.169 | -0.076 |

| Dataset   | Reuters-21578 | 20-Newsgroup | OHSUMED |
|-----------|---------------|---------------|---------|
| Rocchio   | 0.7956        | 0.7197        | 0.4363  |
| HMCM      | 0.8079        | 0.8021        | 0.5339  |
| DR        | 0.8409        | 0.7563        | 0.4130  |
| SLRCM     | 0.8643        | 0.8257        | 0.5761  |
| SVM       | 0.8549        | 0.8272        | 0.5754  |
| **Improvement (%)** | 1.100 | -0.182 | 0.122 |

†Liblinear:http://www.csie.ntu.edu.tw/~cjlin/liblinear/
initial learning rate and momentum are set to 200, 0.1 and 0.2 respectively.

On all the three datasets, SLRCM gets great improvements by comparing with all the other three centroid-based methods. On dataset Reuters-21578, the Micro-F1 of SLRCM is 10.25%, 4.49% and 0.95% higher than Rocchio, HMCM and DR with p-value equal to 0.025763, 0.019341 and 0.000417 respectively; The Macro-F1 of SLRCM is 8.63%, 6.98% and 2.78% higher than Rocchio, HMCM and DR with p-value equal to 0.042252, 0.008226 and 0.006815 respectively. On dataset 20-Newsgroup, the Micro-F1 for SLRCM is 14.68%, 2.94% and 9.17% better than Rocchio, HMCM and DR with p-value equal to 0.037728, 0.013934 and 0.033653 respectively; moreover, the Macro-F1 for SLRCM is 14.73%, 2.94% and 9.18% better than Rocchio, HMCM and DR with p-value equal to 0.037619, 0.013927 and 0.033602 respectively. On dataset OHSUMED, the Micro-F1 of SLRCM beats Rocchio by 38.08%, HMCM by 6.06% and DR by 46.83% with p-value equal to 0.042664, 0.005779 and 0.028566 respectively; meanwhile, the Macro-F1 beats Rocchio by 32.04%, HMCM by 7.90% and DR by 39.49% with p-value equal to 0.048771, 0.000322 and 0.010991 respectively.

Meanwhile, SLRCM outperforms SVM on Reuters-21578. The Micro-F1 and Macro-F1 for SLRCM is approximately 0.87% and 1.10% better than SVM with p-value equal to 0.002011 and 0.000112 respectively. On 20-Newsgroup and OHSUMED, SLRCM gets very similar performance with SVM. The Micro-F1 and Macro-F1 change within the scope of 0.169% and 0.182% respectively.

In addition, for proving the stability of SLRCM, three experiments were conducted.

1. Performance stability for different categories: the detailed Micro-F1 and Macro-F1 of all categories on three datasets are shown in Fig. 1, Fig. 2 and Fig. 3. SLRCM outperforms Rocchio, HMCM and DR in most categories on all datasets. Meanwhile, SLRCM achieves similar performance as the SVM in most categories on all datasets.

2. Performance stability for different feature selection methods: the detailed Micro-F1 and Macro-F1 of Rocchio, DR, HMCM, SVM and SLRCM for six feature selection methods on all three datasets are shown in Table 5. From this table, we can see that the performance of SLRCM is obvious better than Rocchio, HMCM and DR for most of the feature selection methods on three datasets. Meanwhile, it gets similar performance with SVM for all the six feature selection methods on three datasets.

3. Performance stability for different feature weighting methods: the detailed Micro-F1 and Macro-F1 of Rocchio, DR, HMCM, SVM and SLRCM for three feature weighting methods on all of the three datasets are shown in Table 6. The results show that SLRCM makes significant improvements by comparing to Rocchio, DR and HMCM for most of the feature weighting methods on test corpora and reaches nearly the same performance as SVM for all the three feature weighting methods on three datasets.

The parameters $\sigma$ and $\theta$ in SLRCM reflect the loss function sensitivity to the step function and the ranking position of the target centroid of samples respectively. Figures 4, 5 and 6 illustrate the categorization results with different parameters $\sigma$ and $\theta$ on Reuters-21578, 20-Newsgroup and OHSUMED. We can see that it performs differently on the three datasets as parameter $\sigma$ varies from 0.8 to 4.0 and $\theta$ varies from 0.2 to 2.8, both with the step of 0.2. For $\sigma$, the overall trend of Micro-F1 and Macro-F1 are both decreasing for 20-Newsgroup and OHSUMED, whereas the Micro-
Fig. 2  F1 measures of each category using different methods on 20-Newsgroup.

Fig. 3  F1 measures of each category using different methods on OHSUMED.

F1 increases and Macro-F1 decreases as $\sigma$ varies from 0.8 to 4.0 for Reuters-21578. For $\theta$, the overall trend of the Micro-F1 and Macro-F1 are increasing, decreasing and decreasing on Reuters-21578, 20-Newsgroup and OHSUMED respectively. The main reason is that different datasets have different sample distributions, which lead to different loss sensitivity to the step function and ranking position of the target centroid, i.e., in Fig. 4, the Micro-F1 and Macro-F1 on Reuters-21578 are almost continuously rising as $\theta$ varies from 0.2 to 2.8, which shows that it is positive for Reuters-21578 to widen the position distance temperately; moreover, in Fig. 6, the Micro-F1 and Macro-F1 on OHSUMED are both in a sharp decline when $\theta$ gets to a critical value of 2.2. It shows that the performance became very poor for OHSUMED as $\theta$ reaches a critical point, at which the position distances are excessively emphasized. Thus adjusting of the parameters $\sigma$ and $\theta$ for different datasets is necessary, which provides some benefit to the overall performance.

5. Conclusion and Future Work

In this paper, a unified framework of constructing prototype vectors through designing various ranking loss functions is proposed. This framework provides a clear view by treating centroid-based text categorization as a ranking task. Under this framework, three popular weight adjustment schemes for centroid-based classifier are discussed and a novel method named SLRCM is further proposed. Experi-
Table 5  Performance comparison of Rocchio, HMC, DR, SLRC, SVM for six feature selection methods on three datasets.

|               | Reuters-21578 |               | 20-Newsgroup |               | OHSUMED       |
|---------------|---------------|---------------|--------------|---------------|--------------|
|               | Rocchio DR HMC SVM SLRC | Rocchio DR HMC SVM SLRC | Rocchio DR HMC SVM SLRC | Rocchio DR HMC SVM SLRC |
| **DF**        |               |               |              |               |              |
| Micro-F1      | 0.8328 0.9080 0.8758 0.9113 0.9131 | 0.8664 0.7134 0.7597 0.8087 0.8039 | 0.6635 0.7132 0.7594 0.8085 0.8037 | 0.4580 0.2594 0.5304 0.5765 0.5912 |
| Macro-F1      | 0.7973 0.8329 0.8080 0.8591 0.8689 |               |              |               |              |
| **IG**        |               |               |              |               |              |
| Micro-F1      | 0.8047 0.8688 0.8590 0.8827 0.8718 | 0.6354 0.6366 0.6447 0.6435 0.6404 | 0.3952 0.3843 0.4292 0.4291 0.4252 | 0.3446 0.3358 0.3789 0.3809 0.3716 |
| Macro-F1      | 0.7560 0.7848 0.8102 0.8340 0.8308 |               |              |               |              |
| **ECE**       |               |               |              |               |              |
| Micro-F1      | 0.8277 0.9039 0.8733 0.9046 0.9125 | 0.7202 0.7565 0.8023 0.8273 0.8259 | 0.4764 0.4480 0.6202 0.6583 0.6578 | 0.4363 0.4130 0.5339 0.5754 0.5761 |
| Macro-F1      | 0.7956 0.8409 0.8079 0.8549 0.8643 |               |              |               |              |
| **MI**        |               |               |              |               |              |
| Micro-F1      | 0.7290 0.8212 0.7968 0.8607 0.8530 | 0.6318 0.602 0.6473 0.6440 0.6453 | 0.3043 0.2968 0.3218 0.3270 0.3265 | 0.2889 0.2892 0.3019 0.3022 0.3040 |
| Macro-F1      | 0.6404 0.7303 0.7442 0.7831 0.7850 |               |              |               |              |
| **CHI**       |               |               |              |               |              |
| Micro-F1      | 0.8265 0.8944 0.8772 0.9126 0.9146 | 0.7106 0.7451 0.7900 0.8198 0.8172 | 0.4869 0.3664 0.5878 0.6249 0.6295 | 0.4116 0.3217 0.4731 0.5175 0.5187 |
| Macro-F1      | 0.7798 0.8408 0.8130 0.8618 0.8644 |               |              |               |              |

Table 6  Performance comparison of Rocchio, HMC, DR, SLRC, SVM for three feature weighting methods on three datasets.

|               | Reuters-21578 |               | 20-Newsgroup |               | OHSUMED       |
|---------------|---------------|---------------|--------------|---------------|--------------|
|               | Rocchio DR HMC SVM SLRC | Rocchio DR HMC SVM SLRC | Rocchio DR HMC SVM SLRC | Rocchio DR HMC SVM SLRC |
| **TF-IDF**    |               |               |              |               |              |
| Micro-F1      | 0.7483 0.8991 0.8784 0.9166 0.9115 | 0.6779 0.7560 0.7823 0.8182 0.8128 | 0.4734 0.4425 0.6096 0.6543 0.6566 | 0.4219 0.4201 0.5157 0.5720 0.5667 |
| Macro-F1      | 0.7553 0.8669 0.8033 0.8672 0.8727 |               |              |               |              |
| **TF-ICF**    |               |               |              |               |              |
| Micro-F1      | 0.7282 0.8911 0.8990 0.9124 0.9077 | 0.7163 0.7473 0.7960 0.8195 0.8158 | 0.4713 0.4449 0.6171 0.6515 0.6546 | 0.4095 0.4130 0.5212 0.5676 0.5659 |
| Macro-F1      | 0.7236 0.8451 0.8289 0.8559 0.8582 |               |              |               |              |
| **TF-IDF-ICF**|               |               |              |               |              |
| Micro-F1      | 0.8277 0.9039 0.8733 0.9046 0.9125 | 0.7202 0.7565 0.8023 0.8273 0.8259 | 0.4764 0.4480 0.6202 0.6583 0.6578 | 0.4363 0.4130 0.5339 0.5754 0.5761 |
| Macro-F1      | 0.7956 0.8409 0.8079 0.8549 0.8643 |               |              |               |              |

Fig. 4  F1 measures with different $\sigma$ and $\theta$ on Reuters-21578.

Fig. 5  F1 measures with different $\sigma$ and $\theta$ on 20-Newsgroup.

Fig. 6  F1 measures with different $\sigma$ and $\theta$ on OHSUMED.

ments conducted on three benchmark evaluation collections show that SLRCM achieves great improvements compared with other centroid-based methods and reaches nearly the same performance as the state-of-the-art TC methods.

There are numerous information retrieval techniques which may be very helpful for text categorization and we just try to use one of them in this paper. Although SLRCM has reached good performance, it still has great room for further research to enhance the accuracy and efficiency. In addition, SLRCM is only tested on three TC datasets and there are still more classification datasets that can be used in the future research.
Acknowledgment

This work is supported in part by National Natural Science Foundation of China (No.61173075 and No.61272383).

References

[1] R. Feldman and J. Sanger, The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data, Cambridge University Press, Cambridge, 2007.
[2] K. Aas and L. Eikvil, Text Categorization: A Survey, Technical Report 941, Norwegian Computing Center, 1999.
[3] Y. Yang and J.O. Pedersen, “A comparative study on feature selection in text categorization,” International Conference on Machine Learning, 1997.
[4] D. Wang, Q. Chen, X. Wang, and B. Tang, “Macro features based text categorization,” 18th International Conference on Neural Information Processing, pp.211–219, 2011.
[5] F. Sebastiani and C.N.D. Ricerche, Machine Learning in Automated Text Categorization, 34: 1–47, ACM Computing Surveys, 2002.
[6] H. Guan, J. Zhou, and M. Guo, “A class-feature-centroid classifier for text categorization,” World Wide Web Conference, pp.201–210, 2009.
[7] Y.Q. Miao and M. Kamel, “Pairwise optimized Rocchio algorithm for text categorization,” Pattern Recognit. Lett., vol.33, pp.375–382, 2010.
[8] S. Tan, X. Cheng, B. Wang, H. Xu, M.M. Ghanem, and Y. Guo, “Using DragPushing to refine centroid text classifiers,” ACM Special Inspector General for Iraq Reconstruction Conference on Research and Development in Information Retrieval, pp.653–654, 2005.
[9] H. Chen and T.K. Ho, “Evaluation of decision forests on text categorization,” 7th SPIE Conference on Document Recognition and Retrieval, pp.191–199, 2000.
[10] H. Li and K. Yamanishi, “Text classification using ESC-based stochastic decision lists,” Information Processing and Management, vol.38, pp.343–361, 2002.
[11] B. Liu, Y. Xiao, L. Cao, and P.S. Yu, “Orientation distance-based discriminative feature extraction for multi-class classification,” Conference on Information and Knowledge Management, pp.909–918, 2010.
[12] A. McCallum and K. Nigam, “A comparison of event models for naive bayes text classification,” American Association for Artificial Intelligence National Conference, 1998.
[13] I. Batal and M. Hauskrecht, “Boosting KNN text classification accuracy by using supervised term weighting schemes,” Conference on Information and Knowledge Management, pp.2041–2044, 2009.
[14] G. Guo, H. Wang, and D. Dell, “Using KNN model for automatic text categorization,” Soft Computing, vol.10, pp.423–430, 2006.
[15] G. Forman, M. Scholz, and S. Rajaram, “Feature shaping for linear SVM classifiers,” Conference on Knowledge Discovery and Data Mining, pp.299–307, 2009.
[16] S.S.R. Mengle and N. Goharian, “Using ambiguity measure feature selection algorithm for support vector machine classifier,” ACM Symposium on Applied Computing, pp.916–920, 2008.
[17] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” Journal of Machine Learning Research, pp.45–66, 2001.
[18] S.B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” Informatica, vol.31, pp.249–268, 2007.
[19] V. Lertnattee and T. Theeramunkong, “Effect of term distributions on centroid-based text categorization,” Information Sciences, vol.158, pp.89–115, 2004.
[20] Z. Cataltepe and E. Aygun, “An improvement of centroid-based classification algorithm for text classification,” IEEE 23rd International Conference on Data Engineering Workshop, pp.952–956, 2007.
[21] A. CardosoCachopo and A.L. Oliveira, “Semi-supervised single-label text categorization using centroid-based classifiers,” ACM Symposium on Applied Computing, pp.844–851, 2007.
[22] S. Tan, “An improved centroid classifier for text categorization,” Expert Systems with Applications, vol.6, pp.1–7, 2007.
[23] H. Yu, T.H. Phang, B. Liu, and X. Li, “A refinement approach to handling model misfit in text categorization,” Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.207–216, 2002.
[24] S. Tan, “Using hypothesis margin to boost centroid text classifier,” ACM Symposium on Applied Computing, pp.398–403, 2007.
[25] M.E. Maron, “Automatic indexing: An experimental inquiry,” J. Association for Computing Machinery, vol.8, pp.404–417, 1961.
[26] S. Tan, “Using error-correcting output codes with model-refinement to boost centroid text classifier,” Proc. 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp.81–84, 2007.
[27] V. Lertnattee and T. Theeramunkong, “Effects of term distributions on binary classification,” IEICE Trans. Inf. & Syst., vol.E90-D, no.10, pp.1592–1600, 2007.
[28] V. Lertnattee and T. Theeramunkong, “Class normalization in centroid-based text categorization,” Information Sciences, vol.176, no.12, pp.1721–1738, 2006.
[29] E.-H. Han and G. Karypis, “Centroid-based document classification: analysis and experimental results,” Proc. 4th European Conference on Principles of Data Mining and Knowledge Discovery, pp.424–431, 2000.
[30] P.N. Bennett, K. Svore, and S.T. Dumais, “Classification-enhanced ranking,” World Wide Web Conference, pp.111–120, 2010.
[31] E. Gabriovich, A. Broder, M. Fontoura, A. Joshi, and V. Josifovski, “Classifying search queries using the Web as a source of knowledge,” ACM Trans. Web, vol.3, pp.1–28, 2009.
[32] A. Eliseev and J. Weston, “A kernel method for multi-labelled classification,” Advances in Neural Information Processing Systems, 2001.
[33] D. Mladenic and M. Grobelnik, “Feature selection on hierarchy of web documents,” Decision Support Systems, vol.35, no.1, pp.45–87, 2003.
[34] J.W. Reed, Y. Jiao, T.E. Potok, B.A. Klump, M.T. Elmore, and A.R. Hurson, “TF-ICF: A new term weighting scheme for clustering dynamic data streams,” Proc. 5th International Conference on Machine Learning and Applications, pp.258–263, 2006.
[35] F. Ren and M.G. Sohrab, “Class-indexing-based term weighting for automatic text classification,” Information Science, vol.236, no.1, pp.109–125, 2013.
[36] V. Lertnattee and T. Theeramunkong, “Analysis of inverse class frequency in centroid-based text classification,” IEEE International Symposium on Communications and Information Technology, vol.2, pp.1171–1176, 2004.
Qingcai Chen was born in 1975. He received the B.E. degree in Engineering Mechanics, the M.E. degree in Solid Mechanics, and the Ph.D. degree in Computer Application from Harbin Institute of Technology in 1996, 1998, and 2003, respectively. He served as a Senior Software Engineer in Intel (China) Co., Ltd from 2003 to 2004 and joined Harbin Institute of Technology Shenzhen Graduate School as an Associate Professor in 2004. Currently, he is a Professor of Computer Science at Harbin Institute of Technology Shenzhen Graduate School. His research interests include text/audio retrieval, voice processing, natural language processing, handwriting recognition and machine learning.

Xiaolong Wang was born in 1955. He received the B.E. degree in Computer Science from the Harbin Institute of Electrical Technology, China, the M.E. degree in Computer Architecture from Tianjin University, China, and the Ph.D. degree in Computer Science and Engineering from Harbin Institute of Technology in 1982, 1984, and 1989, respectively. He joined Harbin Institute of Technology as an Assistant Lecturer in 1984 and became an Associate Professor in 1990. He was a Senior Research Fellow in the Department of Computing, Hong Kong Polytechnic University from 1998 to 2000. Currently, he is a Professor of Computer Science at Harbin Institute of Technology. His research interests include artificial intelligence, machine learning, computational linguistics, and Chinese information processing.