Expert Systems with Applications 36 (2009) 9168–9174

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Rough set based hybrid algorithm for text classification

Duoqian Miao [*], Qiguo Duan, Hongyun Zhang, Na Jiao

Department of Computer Science and Technology, Tongji University, Caoan Street 4800, Shanghai 201804, China

ARTICLE INFO

Keywords:
Text classification
Variable precision rough set (VPRS)
k-nearest neighbor (kNN)
Rocchio algorithm

ABSTRACT

Automatic classification of text documents, one of essential techniques for Web mining, has always been a hot topic due to the explosive growth of digital documents available on-line. In text classification community, *k*-nearest neighbor (kNN) is a simple and yet effective classifier. However, as being a lazy learning method without premodelling, kNN has a high cost to classify new documents when training set is large. Rocchio algorithm is another well-known and widely used technique for text classification. One drawback of the Rocchio classifier is that it restricts the hypothesis space to the set of linear separable hyperplane regions. When the data does not fit its underlying assumption well, Rocchio classifier suffers. In this paper, a hybrid algorithm based on variable precision rough set is proposed to combine the strength of both kNN and Rocchio techniques and overcome their weaknesses. An experimental evaluation of different methods is carried out on two common text corpora, i.e., the Reuters-21578 collection and the 20-newsgroup collection. The experimental results indicate that the novel algorithm achieves significant performance improvement.

1. Introduction

Text classification or categorization is the task of automatically assigning unseen documents to suitable pre-defined categories. As one of essential techniques for Web mining, it has always been a hot topic due to the explosive growth of digital documents available on-line. A number of well-known algorithms have been introduced to deal with text classification, such as k-nearest neighbor (kNN) (Cover & Hart, 1967; Yang & Liu, 1999), Naïve Bayesian (NB) (Lewis, 1998), centroid-based classifier (Han & Karypis, 2000; Tang, 2007), Support Vector Machine (SVM) (Joachims, 1998), decision tree (Quinlan, 1986) and Rocchio classifier (Joachims, 1997).

The kNN is an instance-based learning algorithm, which is simple, intuitive but very effective for a variety of problem domains including text classification. It has been applied to text classification since the early days of its research, and is known to be one of the most effective methods on the Reuters corpus of newswire stories – one of the benchmark corpora used in text classification. However, kNN has a high cost of classifying new patterns. Its training phase just stores all training patterns as classifier, thus it has often been called as lazy learner since it defers the decision on how to generalize beyond the training data until each new query pattern is encountered (Sebastiani, 2002). The efficiency of kNN prohibits it from being applied to areas where efficiency is particularly required for text classification, such as dynamically mining large scale collection.

Rocchio algorithm, an early text classification technique from information retrieval, has also been widely used for document classification. Most classification techniques are based on some underlying models. When the data fits the model well, the classification accuracy can be very high, and vice versa. The Rocchio classification model is based on the assumption that a given document should be assigned to a particular class if the similarity between this document vector and the prototype vector of the class is the largest. Thus, Rocchio algorithm restricts the hypothesis space to the set of linear separable hyperplane regions, which has less expressive power than that of kNN. When the data does not fit the Rocchio classification model well, the Rocchio classifier suffers.

In this paper, a hybrid algorithm based on variable precision rough set (VPRS) is proposed to combine the strength of both kNN and Rocchio techniques and overcome their weaknesses. Firstly, feature space of training data is partitioned by using VPRS, and lower and upper approximations of each category are defined. Then kNN and two Rocchio classifiers are built on these new subspaces respectively. The two Rocchio classifiers are used to classify most of new documents effectively and efficiently. The kNN classifier is only required to classify new document which lies in the boundary region where Rocchio classifier suffers. And it is just required to find nearest neighbors of new document in the subset of training dataset, which can save time obviously compared with finding nearest neighbors in the whole training dataset. Experiments are carried out on two public benchmarks, namely, the ModApte version of the Reuters-21578 collection of news stories and the 20-newsgroup collection. The experimental results indicate that the proposed hybrid algorithm achieves significant performance improvement.

[*] Corresponding author.
E-mail addresses: miaoduoqian@163.com (D. Miao), dqgcn@126.com (Q. Duan).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.12.026

The remainder of the paper is organized as follows: Section 2 gives an overview of related work. Section 3 introduces the basic background ideas about VPRS, kNN and Rocchio algorithms for the sake of further discussion. Section 4 describes the proposed hybrid algorithm. Section 5 discusses experimental results. Finally, Section 6 presents concluding remarks and directions of our future work.

2. Related work

Improving prediction accuracy of text classifiers has been an important issue and many studies have been conducted in this area.

Rocchio is a linear classifier. When the decision boundary is non-linear, the classification accuracy of Rocchio classifier is low. Lam and Ho (1998) proposed a generalized pattern set algorithm to overcome the weakness of Rocchio algorithm. The main idea for this method is to construct more than one prototype vector for a category, in contrast to only one prototype vector for a category in the Rocchio algorithm. The drawback of this method is the difficulty to choose an appropriate k and the order in which positive patterns are chosen to construct each local prototype vector as the performance of the method depends on both of them.

Tang and Gao (2007) combined kNN and SVM to construct a classifier for improvement of classification accuracy. However, SVM is very sensitive to noise and the application of overlapped patterns to train SVM could make the classification performance poor.

Sarkar (2007) introduced fuzzy-rough uncertainty to enhance classification performance of the kNN algorithm. Some drawbacks still exist for this method. Firstly, it need to store all training data and hence for a large training set it may take large space. Secondly, for every new pattern, the distance should be computed between the new pattern and all training data. Thus, the efficiency of this method may be low.

Ensemble methods such as bagging and boosting have been studied extensively to improve the predictive accuracy of classification algorithms, which train a set of component classifiers and then combine their predictions to classify new patterns (Dietterich, 2000). Compared to ensemble methods, the proposed algorithm has two particularities. Unlike ensemble methods, the proposed algorithm does not need to retrain the classifier multiple times on the different versions of the entire training set. Consequently the proposed algorithm consumes much less training time than the ensemble methods. Otherwise, unlike ensemble methods, no voting is involved in the proposed algorithm. Hence the prediction is much faster than ensemble methods.

3. Background

3.1. Variable precision rough set

The rough set theory, introduced by Pawlak in the early 1980s, is a formal mathematical tool to deal with incomplete or imprecise information (Pawlak, 1982). As a generalized version of rough sets, VPRS allows objects to be classified with an error smaller than a certain pre-defined level (Ziarko, 1993). In this section, the brief introduction to rough set and VPRS is given.

3.1.1. Rough set

Definition 1. Information system.

In rough set theory, an information system is defined as a 4-tuple $S = \langle U, Q, V, f \rangle$, where U is a non-empty finite set of objects, Q is a non-empty finite set of attributes, V is a set of values of attributes in Q and $f : U \times Q \rightarrow V$ a description function.

For any $P \subseteq Q$, the indiscernibility relation, denoted by $IND(P)$, is defined as Definition 2.

Definition 2. Indiscernibility relation.

$$IND(P) = \{(x, y) \in U \times U : \quad \forall a \in p, a(x) = a(y)\} \tag{1}$$

where $a(x)$ denotes the value of feature a of object x. If $(x, y) \in IND(P)$, x and y are said to be indiscernible with respect to P. The equivalence classes of the P-indiscernibility relation are denoted by $[x]_p$.

For any concept $X \subseteq U$ and attribute set $P \subseteq Q$, X could be approximated by the lower approximation and upper approximation.

Definition 3. Lower approximation and upper approximation.

The lower approximation of X is the set of objects of U that are surely in X, defined as:

$$\underline{P}(X) = \{x \in U : [x]_p \subseteq X\} \tag{2}$$

The upper approximation of X is the set of objects of U that are possibly in X, defined as:

$$\overline{P}(X) = \{x \in U : [x]_p \cap X \neq \emptyset\} \tag{3}$$

3.1.2. Variable precision rough set

As a generalization of the standard inclusion relation, majority inclusion relation introduced by the VPRS is defined as Definition 4.

Definition 4. Majority inclusion relation

$$c(X, Y) = \begin{cases} 1 - \frac{\text{card}\,(X \cap Y)}{\text{card}\,(X)} & \text{if card}\,(X) \geqslant 0 \\ 0 & \text{if card}\,(X) = 0 \end{cases} \tag{4}$$

where X and Y are subsets of the universe U. The majority inclusion relation denotes the relative degree of misclassification of the set X with respect to set Y. Based on this measure, one can define the standard set inclusion relation between X and Y as: $X \subseteq Y$ if and only if $c(X, Y) = 0$.

Let $X \subseteq U$, R an equivalence relation on U, the β-lower approximation and β-upper approximation of X can be defined as Definition 5.

Definition 5. β-Lower approximation and β-upper approximation

$$\underline{R}_\beta X = \{x \in U : c([x]_R, X) \leqslant \beta\} \tag{5}$$

$$\overline{R}_\beta X = \{x \in U : c([x]_R, X) < 1 - \beta\} \tag{6}$$

Therefore, the new definition of β-positive region, β-negative region and β-boundary region based on VPRS is given in Definition 6.

Definition 6. β-positive, β-negative and β-boundary region based on VPRS

$$POS_\beta X = \underline{R}_\beta X \tag{7}$$

$$NEG_\beta X = U - \overline{R}_\beta X \tag{8}$$

$$BND_\beta X = \overline{R}_\beta X - \underline{R}_\beta X \tag{9}$$

The definitions above are generalizations of the corresponding notions appearing in Pawlak's rough set. In this paper, VPRS is used to partition feature space of training data. Then, the proposed hybrid algorithm can further build kNN and Rocchio classifiers and classify new document effectively and efficiently in new subspaces.

3.2. Text classification technique

In this section, we review two text classification techniques applied in the paper, i.e., kNN and Rocchio algorithms.

3.2.1. kNN algorithm

The kNN is a similarity-based learning algorithm. To classify an unknown document d, the kNN classifier ranks the documents among training set and tries to find its k-nearest neighbors, which forms a neighborhood of d. Then majority voting among the categories of documents in the neighborhood is used to decide the class label of d.

3.2.2. Rocchio algorithm

In Rocchio algorithm, each document d is represented as a vector. Given a training set D, Rocchio algorithm computes a prototype vector $\vec{c}_j$ for each category c_j by means of the formula as follow:

$$\vec{c}_j = \frac{1}{|C_j|}\sum_{\vec{d}\in C_j}\frac{\vec{d}}{\|\vec{d}\|} - \mu\frac{1}{|D-C_j|}\sum_{\vec{d}\in D-C_j}\frac{\vec{d}}{\|\vec{d}\|} \tag{10}$$

$|.|$ denotes the cardinality of the set. μ is a control parameter used for adjusting the relative impact of positive and negative training patterns. Buckley et al. recommends $\mu = 0.25$ (Buckley, Salton, & Allan, 1994). Given a test document d_t, the Rocchio classifier computes the similarity between $\vec{d}_t$ and each prototype vector $\vec{c}_j$, and classifies as the category of prototype vector which is the most similar to. In this paper, we use the cosine measure to compute the similarity.

4. Proposed algorithm

4.1. Algorithm

In this paper, we consider binary text classification that assigns each document d either to the positive class C_p or to its complement negative class C_n. Theoretically, binary text classification is more general than the multi-class one and a multi-class classification problem can be transformed into a set of binary classifications. For the intuitive presentation, an example of binary classification in 2-D space, where documents in class C_p are labeled with “circle sign” and documents in class C_n are labeled with “cross sign”, is shown in Fig. 1. We can characterize the two class document sets of C_p and C_n with respect to a hidden equivalence relation R which may lead the documents belonging to the same class to have the tendency of clustering.

The kNN algorithm is used to create equivalence classes for set C_p and C_n according to the concepts of VPRS. For a document d in training set, the kNN algorithm is used to find its k-nearest neighbors by means of calculating the similarity of d to anyone in training set, which forms a neighborhood of d. If all neighbors are from a single class, e.g., C_p, then there is no uncertainty in the neighborhood. However, if any neighbor belongs to another class C_n, the rough uncertainty arises in the neighborhood. This uncertainty can be captured using the modified majority inclusion relation. For any document d and document set of class C_p in training dataset, modified majority inclusion relation is defined as Definition 7.

Definition 7. Modified majority inclusion relation

$$c(N_d, C_p) = 1 - \frac{|N_d \cap C_p|}{|N_d|} \tag{11}$$

where N_d is the neighborhood region around d and $|.|$ denotes the cardinality of the set. According to VPRS, β-positive region, β-negative region and β-boundary region of class C_p can be obtained. As shown in Fig. 2, the β-positive region of class C_p denotes the document set which lies in the positive region where documents can be for certain classified to class C_p, β-boundary region of class C_p denotes the document set which lies in the boundary region where documents can not be classified uniquely to the class C_p and β-negative region of class C_p denotes the document set can not be surely classified to the class C_p respectively. Similar description is also fit to class C_n.

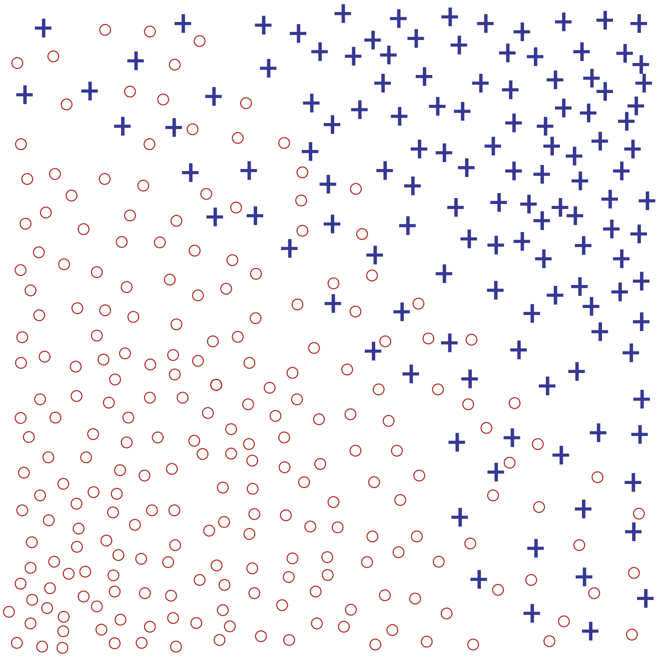

Fig. 1. An example with two classes.

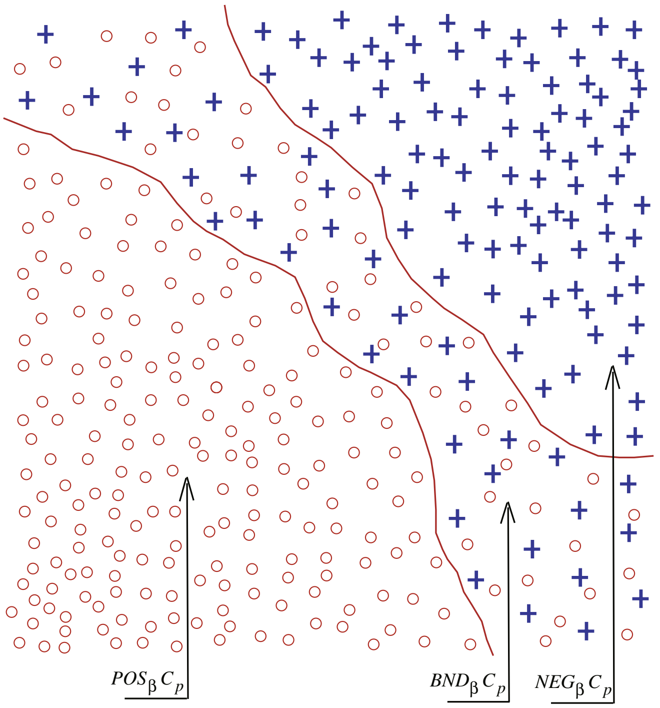

Fig. 2. VPRS based partition for two classes.

After feature space is partitioned into three regions, i.e., β-positive, β-negative and β-boundary region for each class, two Rocchio classifiers are built on these new sets directly. The detailed building classifier algorithm is described as Algorithm 1.

Algorithm 1. Building classifier algorithm

```
Input: the parameter $k$-partition, $\beta$, $\mu$, training set
Output: two Rocchio classifiers
(1) Find the $k$-partition nearest neighbor documents of each document in the training set;
(2) Construct $\underline{R}_\beta C_p$ and $\overline{R}_\beta C_p$ of class $C_p$, $\underline{R}_\beta C_n$ and $\overline{R}_\beta C_n$ of class $C_n$, respectively based on Definition 7 and VPRS;
(3) Build two Rocchio classifiers, i.e., Rocchio classifier for class $C_p$ and Rocchio classifier for class $C_n$, respectively;
```

Rocchio classifier for class C_p:

$$\vec{V}^{+}_{C_p} = \frac{1}{|\underline{R}_\beta C_p|}\sum_{d\in\underline{R}_\beta C_p}\frac{\vec{d}}{\|\vec{d}\|} - \mu\frac{1}{|\overline{R}_\beta C_n|}\sum_{d\in\overline{R}_\beta C_n}\frac{\vec{d}}{\|\vec{d}\|}$$

$$\vec{V}^{-}_{C_p} = \frac{1}{|\overline{R}_\beta C_n|}\sum_{d\in\overline{R}_\beta C_n}\frac{\vec{d}}{\|\vec{d}\|} - \mu\frac{1}{|\underline{R}_\beta C_p|}\sum_{d\in\underline{R}_\beta C_p}\frac{\vec{d}}{\|\vec{d}\|}$$

Rocchio classifier for class C_n:

$$\vec{V}^{+}_{C_n} = \frac{1}{|\underline{R}_\beta C_n|}\sum_{d\in\underline{R}_\beta C_n}\frac{\vec{d}}{\|\vec{d}\|} - \mu\frac{1}{|\overline{R}_\beta C_p|}\sum_{d\in\overline{R}_\beta C_p}\frac{\vec{d}}{\|\vec{d}\|}$$

$$\vec{V}^{-}_{C_n} = \frac{1}{|\overline{R}_\beta C_p|}\sum_{d\in\overline{R}_\beta C_p}\frac{\vec{d}}{\|\vec{d}\|} - \mu\frac{1}{|\underline{R}_\beta C_n|}\sum_{d\in\underline{R}_\beta C_n}\frac{\vec{d}}{\|\vec{d}\|}$$

In the Algorithm 1, kNN is used to find the k-partition nearest neighbors for each document in the training set firstly and then $\underline{R}_\beta C_p$, $\overline{R}_\beta C_p$, $\underline{R}_\beta C_n$ and $\overline{R}_\beta C_n$ are constructed, respectively based on Definition 7 and VPRS. For overcoming the problem of model misfit of Rocchio and improving the classification performance, Rocchio classifier for class C_p is trained with the documents belonging to $\underline{R}_\beta C_p$ and $\overline{R}_\beta C_n$, and Rocchio classifier for class C_n is trained with the documents belonging to $\overline{R}_\beta C_p$ and $\underline{R}_\beta C_n$, respectively. After the Rocchio classifiers are built, a new unseen document d can be classified by using the classification algorithm described as Algorithm 2.

Algorithm 2. Classification algorithm

```
Input: the parameter $k$-boundary, two Rocchio classifiers, set $BND_\beta C_p$, new document $d$
Output: the class label of d
(1) If sim$\left(\vec{V}^{+}_{C_p},\vec{d}\right) \geqslant$ sim$\left(\vec{V}^{-}_{C_p},\vec{d}\right)$ Then
(2)     assign $d$ to class $C_p$ ;
(3) Else If sim$(\vec{V}^{+}_{C_n},\vec{d}) \geqslant$ sim$(\vec{V}^{-}_{C_n},\vec{d})$ Then
(4)         assign $d$ to class $C_n$;
(5)     Else
(6)         use kNN classifier to find $k$-boundary nearest neighbor documents of $d$ in the $BND_\beta C_p$ and classify $d$ via majority voting;
(7)     End If
(8) End If
```

In the Algorithm 2, if sim$(\vec{V}^{+}_{C_p},\vec{d}) \geqslant$ sim$(\vec{V}^{-}_{C_p},\vec{d})$, classification of document d belongs to $\underline{R}_\beta C_p$ which imply that d belongs to class C_p with high confidence. If sim$(\vec{V}^{+}_{C_n},\vec{d}) \geqslant$ sim$(\vec{V}^{-}_{C_n},\vec{d})$, classification of document d belongs to $\underline{R}_\beta C_n$ which imply that d belongs to class C_n with high confidence. If above two conditions are not satisfied, d belongs to $\overline{R}_\beta C_p$ and $\overline{R}_\beta C_n$. This represents that d lies in the boundary region $BND_\beta C_p$ (also belongs to $BND_\beta C_n$ because of $BND_\beta C_p = BND_\beta C_n$ according to VPRS). For this situation, kNN classifier is applied to classify the document in the boundary between classes.

4.2. The architecture of text classification

The proposed architecture of text classification is described in Fig. 3. The architecture comprises four key phases, i.e., preprocessing, partitioning space of training set based on VPRS, building classifiers and classifying new documents.

4.2.1. Preprocessing

The preprocessing phase is done as follows:

Step 1. Text extracting: remove the HTML tag and extract plain text from each Web page.
Step 2. Common word omitting: use a stop list to omit the most common words.
Step 3. Word stemming: standardize word's suffixes.
Step 4. Feature selecting: reduce the dimensionality of the data space by removing irrelevant features. In this paper, information gain is employed as feature selection method for it consistently performs well in most cases (Yang & Pedersen, 1997).
Step 5. List constructing: construct a list which is used as a reference. When converting the text document to a vector of features, each feature in the vector corresponds to a word in the list.
Step 6. Feature weighting: apply the popular TF*IDF (Term Frequency times Inverse Document Frequency) weighting scheme to assign weight values for document's vector. The standard TF*IDF is defined as formula:

$$w^*_{ij} = tf_{ij} \times \log(N \times df_i) \quad (12)$$

where tf_{ij} is the frequency of the term t_i in document d_j; df_i is number of documents in which term t_i occurs; N is the total number of documents. Normalization by vector's length is applied to all vectors as formula:

$$w_{ij} = w^*_{ij}\sqrt{\sum_{t_k\in d_i}(w^*_{ik})^2} \quad (13)$$

4.2.2. Partitioning space of training set based on VPRS

This phase implements the function of partitioning training dataset into β-positive region, β-negative region and β-boundary region for each category of documents.

4.3. Building classifiers

The role of this phase is to train two Rocchio classifiers according Algorithm 1.

4.4. Classifying new documents

This phase implements the function of classifying new unseen documents according to Algorithm 2. All documents to be classified must be preprocessed according to steps described in Section 4.2.1 only excluding of feature selecting step.

5. Experiment results and discussion

5.1. Experimental datasets

To evaluate the proposed approach, we have conducted experiments on two popular corpora in text classification research, i.e., Reuters-21578[1] and 20-newsgroups[2]

[1] http://www.research.att.com/lewis/reuters21578.html.
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/.

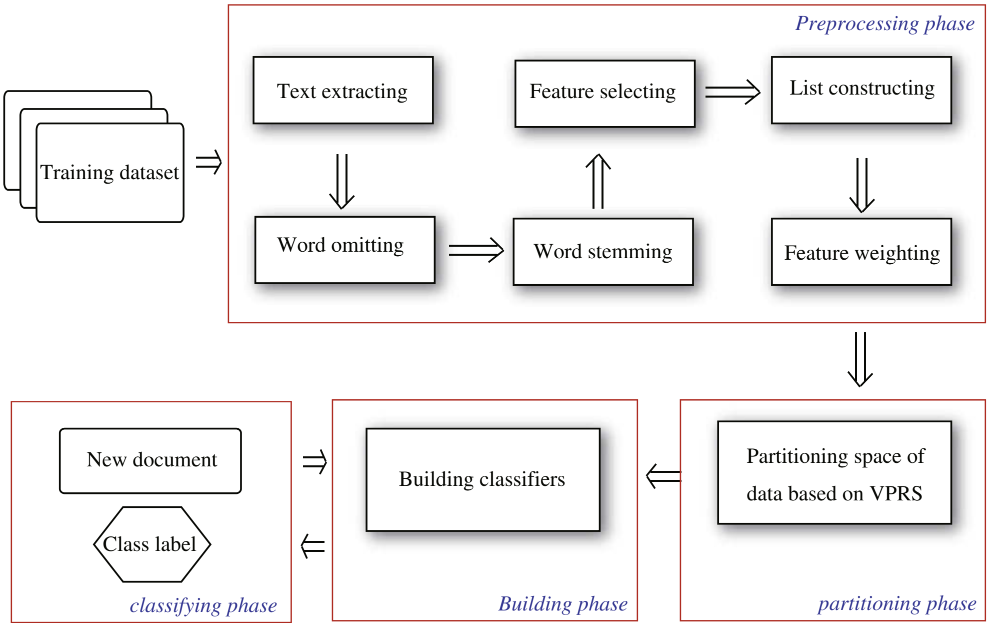

Fig. 3. Schematic view of the architecture of text classification.

Reuters-21578: the Reuters dataset is a standard text classification benchmark, which collected documents by the Carnegie group from the Reuters newswire in 1987. We used the ModApte version of the Reuters-21578 collection for evaluation. In our experiments, only the most populous six categories from this corpus are used as our dataset, i.e., Acq, Crude, Earn, Grain, Interest, Ship. Binary text classification is performed in the experiments.

20-newsgroup: the 20-newsgroup corpus contains approximately 20,000 newsgroup documents being divided nearly evenly among 20 different newsgroups. In the experiments, we used two main categories, i.e., Talk (talk) and Science (sci). The sub-categories of Talk are talk.politics.guns, talk.politics.mideast, talk.politics.misc, talk.religion.misc (respectively abbreviated here as guns, mideast, misc, religion). The sub-categories of Science are sci.crypt, sci.electronics, sci.med, sci.space (respectively abbreviated here as crypt, electronics, medical, space). The classification is performed on the sub-categories within Talk and Science, respectively.

5.1.1. Performance measures

To analyze the performance of classification, we adopt the popular F_1 measure. As shown in Table 1, four cases are considered as the result of classifier to the document (Yang & Liu, 1999).

TP (True Positive): the number of documents correctly classified to that class.
TN (True Negative): the number of documents correctly rejected from that class.
FP (False Positive): the number of documents incorrectly rejected from that class.
FN (False Negative): the number of documents incorrectly classified to that class.

Table 1
Cases of the classification for one class.

Class *C*		Result of classifier	
		Belong	Not belong
Real classification	Belong	*TP*	*FN*
	Not belong	*FP*	*TN*

Using these quantities, the performance of the classification is evaluated in terms of Precision (*pr*), Recall (*re*), and F_1 measure. Precision means the rate of documents classified correctly among the result of classifier and Recall signifies the rate of correct classified documents among them to be classified correctly. F_1 measure is combination of precision and Recall

$$pr = \frac{TP}{TP + FP} \tag{14}$$

$$re = \frac{TP}{TP + FN} \tag{15}$$

$$F_1 = 2\frac{pr \cdot re}{pr + re} \tag{16}$$

The F_1 measure which is the harmonic mean of Precision and Recall is used in this study since it takes into account effects of both quantities, which makes it a more reliable and suitable measure.

For ease of comparison, the macroaveraged F_1 is also used to evaluate the overall performance over the different categories. The macroaveraged F_1 computes the F_1 measure for each category and then takes the average over the per-category F_1 measure. Given a training set with m categories, assuming that the F_1 value for the ith category is $F_1(i)$, the macroaveraged F_1 is defined as:

$$\text{Macro} - F_1 = \frac{\sum_{i=1}^{m} F_1(i)}{m} \tag{17}$$

5.1.2. Results and discussion

To evaluate the efficiency and effectiveness of our proposed algorithm in text classification, two extensively used algorithms in text classification, i.e., kNN and Rocchio, are implemented and used as benchmarks for comparison. We select the first 2000 words with the highest information gains as features. In the first experiment we evaluate the F_1 values of different algorithms on the above two datasets. Performance is evaluated by 10-fold cross validation. The basic settings of parameters for each algorithm in the experiment are summarized in Table 2, where the optimal parameters are chosen for good result based on their performance among all the settings.

Different values of parameters have been tried on each algorithm to ensure that the experimental results faithfully reflect

Table 2
The basic settings of parameters for each algorithm.

Algorithm	Reuters-21578	20-Newsgroup
kNN	$k = 30$	$k = 45$
Rocchio	$\mu = 0.25$	$\mu = 0.25$
Proposed algorithm	$\beta = 0.3$, $\mu = 0.25$ k-partition = 35 k-boundary = 8	$\beta = 0.2$, $\mu = 0.25$ k-partition = 25 k-boundary = 14

the performance of the algorithms. For kNN classifier used as benchmarks for comparison, the value of k varies from 5 to 100 with step 5. For proposed algorithm, the value of k-partition varies from 5 to 50 with step 5 and the value of β varies from 0.05 to 0.45 with step 0.05 in building classifier phase; the value of k-boundary varies from 2 to 20 with step 2 for kNN in classifying phase to classify new document which lies in the boundary region. The F_1 value of each algorithm on each category and the corresponding Macro-F_1 on each dataset are presented in Tables 3 and 4, respectively.

On Reuter-21578, the Macro-F_1 of proposed approach is 85.3%, which is approximately 1.8% higher than that of kNN, and 5.7% higher than that of Rocchio algorithm. On 20-newsgroup, the Macro-F_1 of proposed approach is 87.7%, which beats kNN by about 3.5% and Rocchio algorithm by about 7.1%. Consequently, the proposed algorithm is better than kNN and beats Rocchio algorithm by a wide margin on two datasets.

For the proposed hybrid algorithm, Figs. 4 and 5 illustrate the dependence of the classification performance in term of Macro-F_1 with the parameter β on two datasets, respectively. Note that all parameters are set according to the values described in Table 2 only excluding of β. From the two figures, we can discover a phenomenon that with the increase of β, the proposed hybrid algorithm in the beginning performs better and afterward worse. When β takes 0.3 and 0.2, the proposed algorithm achieves the best results on Reuter-21578 and 20-newsgroup respectively.

Table 3
Comparison of the performances on Reuters-21578 dataset.

Category	kNN	Rocchio	Proposed algorithm
Acq	0.892	0.866	0.923
Wheat	0.761	0.730	0.759
Crude	0.845	0.765	0.871
Earn	0.957	0.911	0.960
Interest	0.752	0.714	0.759
Grain	0.806	0.789	0.849
Macro-F_1	0.835	0.796	0.853

Table 4
Comparison of the performances on 20-newsgroup dataset.

Category	kNN	Rocchio	Proposed algorithm
Guns	0.902	0.836	0.948
Mideast	0.889	0.905	0.933
Misc	0.830	0.743	0.822
Religion	0.797	0.708	0.831
Crypt	0.865	0.832	0.918
Electronics	0.756	0.672	0.747
Medical	0.875	0.903	0.946
Space	0.821	0.847	0.875
Macro-F_1	0.842	0.806	0.877

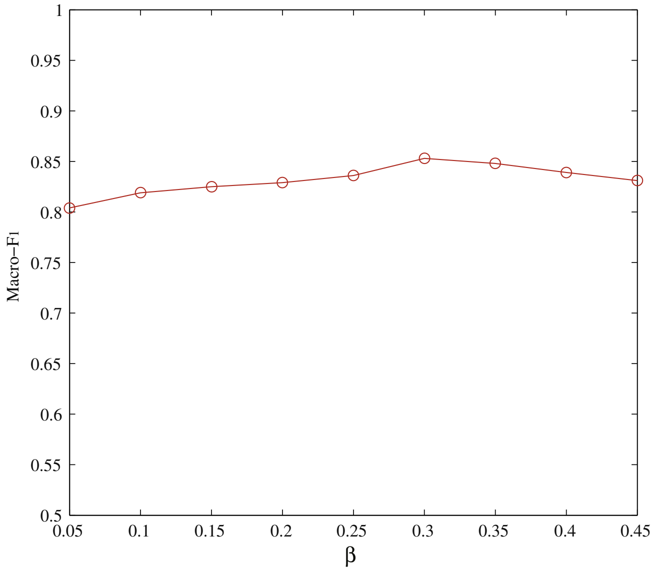

Fig. 4. The performance of the proposed algorithm with the β on Reuter-21578.

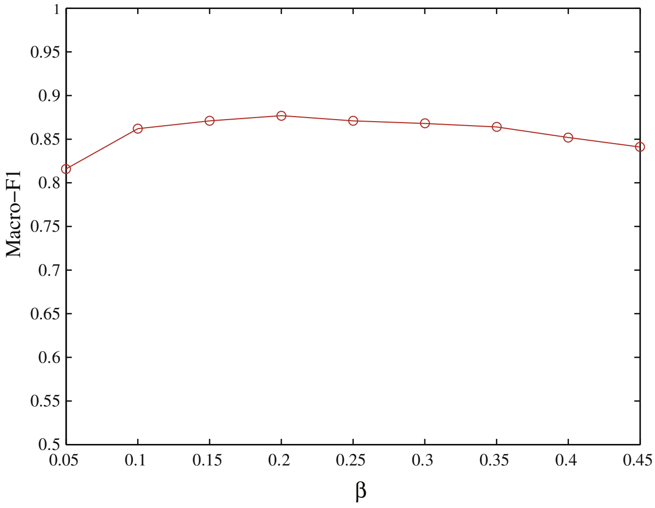

Fig. 5. The performance of the proposed algorithm with the β on 20-newsgroup.

The computational efficiency of text classifiers is often the key element to be considered in many applications such as dynamically mining large web repositories (Sebastiani, 2002). In the second experiment, we conduct a test to evaluate each algorithm's efficiency in classification. With the same settings of the first experiment, two subsets were created for this test. One subset contains 1000 documents randomly chosen from the Reuters-21578 collection, and the other contains 3000 documents randomly chosen from the 20-newsgroup collection. The experimental results are summarized in Table 5. The values in columns represent times in seconds spent on testing by different algorithms.

From the experimental results, the total time costed by the proposed algorithm is about only 21.7% (116.5/537.5) of that costed by kNN, and thus the proposed algorithm is more efficient than kNN.

Above all, the proposed algorithm outperforms kNN and Rocchio in performance on both the Reuters-21578 and 20-newsgroup datasets, and it is more efficient than kNN. Thus, the proposed approach is a good alternative for Rocchio algorithm and kNN in some scenarios of text classification such as dynamically mining large web repositories where kNN is not suitable due to its lower efficiency.

Table 5
Comparison of the efficiency on two datasets.

Dataset	kNN	Rocchio	Proposed algorithm
Reuters-21578	50.7	13.5	18.6
20-Newsgroup	1024.2	156.8	214.3
Total	537.5	85.2	116.5

6. Conclusion

In this paper, two widely used techniques for text classification, i.e., the kNN and the Rocchio algorithm, are analyzed and some shortcomings of each are identified. Based on the analysis, a hybrid algorithm based on VPRS is proposed to combine the strengths of kNN and the Rocchio classifier nd overcome the problems of low efficiency of kNN and model misfit of Rocchio. Extensive experiments conducted on two common document corpora: the Reuters-21578 collection and the 20-newsgroup collection show that the proposed algorithm outperforms the Rocchio and kNN in performance, and is therefore a quite competitive method which can be a good alternative to kNN and Rocchio in some applications. Our future effort is to find new ways to improve the classification performance further and to apply the proposed hybrid approach to integrate other classifiers.

Acknowledgements

This study is supported by the National Natural Science Foundation of China (Granted Nos. 60775036 and 60475019) and the Ph.D. programs Foundation of Ministry of Education of China (No. 20060247039).

References

Buckley, C., Salton, G., & Allan, J. (1994). The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the seventeenth annual ACM SIGIR conference*.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27.

Dietterich, T. (2000). Ensemble methods in machine learning. In J. Kittler, & F. Roli (Eds.), *First international workshop on multiple classifier systems*, *Lecture notes in computer science* (pp. 1–15).

Han, E., & Karypis, G. (2000). Centroid-based document classification analysis and experimental results. <http://www.cs.umn.edu/wkarypis>.

Joachims, T. (1997). A probabilistic analysis of the Rochhio algorithm with TFIDF for text categorization. In *Proceedings of the fourteenth international conference on machine learning*.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *The 10th European conference on machine learning* (pp. 137–142). New York: Springer.

Lam, W., & Ho, C. (1998). Using a generalized instance set for automatic text categorization. *SIGIR'98*. pp. 81–89.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *The 10th European conference on machine learning* (pp. 4–15). New York: Springer.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences, 11*(5), 341–356.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Sarkar, M. (2007). Fuzzy-rough nearest neighbor algorithms in classification. *Fuzzy Sets and Systems, 158*(19), 2134–2152.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Tang, S. B. (2007). Large margin DragPushing strategy for centroid text categorization. *Expert Systems with Applications, 33*(1), 215–220.

Tang, Y. H., & Gao, J. H. (2007). Improved classification for problem involving overlapping patterns. In *IEICE transaction on information and systems* (Vol. E90-D, No.11, pp. 1787–1795).

Ziarko, W. (1993). Variable precision rough set model. *Journal of Computer and Systems Sciences, 46*(1), 39–59.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *The 22nd annual international ACM SIGIR conference on research and development in the information retrieval*. New York: ACM Press.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *ICML*, 412–420.