Decreasing Uncertainty for Improvement of Relevancy Prediction

Libiao Zhang[1,2] **Yuefeng Li**[1,3] **Moch Arif Bijaksana**[1,4]

[1]School of Electrical Engineering and Computer Science,
Queensland University of Technology, Brisbane, QLD 4001, Australia.

[2]L39.zhang@hdr.qut.edu.au [3]y2.li@qut.edu.au [4]arifbijaksana@gmail.com

Abstract

As one of the key techniques of Information Retrieval (IR) and Information Filtering (IF), Text Classification focuses on classifying textual documents into predefined categories through relative classifiers learned from labelled or unlabelled training samples. Binary text classifiers is the main branch of Text Classification, involving the relevance prediction of documents to users or categories. However, the current binary text classifiers cannot clearly describe the difference between relevant and irrelevant information because of knowledge uncertainty owing to the imperfection of the knowledge mining techniques and the limitation of feature selection methods. This paper proposes a relevance prediction model by decreasing the relative uncertainty to improve the performance of binary text classification. It tries to form and train the decision boundary through partitioning the training samples into three regions (the positive, boundary and negative regions) to assure the discrimination of extracted knowledge for describing relevant and irrelevant information. It then produces six decision rules corresponding with six different situations of the related objects to help make relevance predications for those objects. A large number of experiments have been conducted on two standard datasets including RCV1 and Reuters21578. The experiment results show that the proposed model has significantly improved the performance of binary text classification, thus proved to be effective and promising.

Keywords: Relevance prediction, Text classification, Uncertainty, Decision boundary, Decision rule.

1 Introduction

With the explosive growth of electronic textual documents, text analysis and classification is getting increasingly important and attracting extensive attention in the similar research fields in recent years. Relevance prediction is a big research issue [17, 21] for text analysis and classification, which focuses on predicting a document's relevance to a query, a category that a user concerns. Text classification is the process of classifying an incoming stream of textual documents into predefined categories through the classifiers learned from the training samples, labelled or unlabelled. Different kinds of text classification technologies have been invented and developed in different level and utilized to automatically classify the textual documents, such as k-Nearest Neighbors [7], Support Vector Machines [30], Naive Bayes [14], Rocchio Similarity [27] and rule-based methods. With the continuous improvement of text classification technology, its application has been prevalent in the real world and many applications of text classification have been developed in recent years such as the classification of news stories, e-mail message, customer reviews, academic papers or medical records, filtering of spam and porn, and the application in Bioinformatics and customer service automation [30]. A binary text classifier can be used to help gain relevant information to a category or a user's interest, which assigns one of two predefined classes (e.g., relevant category or irrelevant category) to incoming documents since relevance is a single class problem [12]. The most common solution to the multi-class problem is to decompose it into several independent binary classifiers.

A binary classifier usually defines a decision boundary to group documents into two categories: the relevant and irrelevant categories. However, the decision boundary contains a lot of uncertain information because of a number of reasons such as noise of knowledge mining and deficient strategy of feature extraction for text classification.

Text feature selection is the essential step to decrease computational complexity by eliminating noises for building a satisfactory classifier [2]. Over the years, a variety of text feature selection methods have been proposed [21]. The effective way of feature selection for relevance prediction is based on a feature weighting function which indicates the critical degree of information represented by the feature occurrences in a document and reflects the relevance of the feature to the related topic or category.

For many years, we have observed that after a set of features are selected and weighted, documents can be easily grouped into three regions rather than two categories by using a binary classifier. Even training documents previously labelled as relevant or irrelevant can not be reclassified into their original categories when applied any binary classifier [38]. Therefore, it is hard to find a clear boundary by any classic text classifier, which can be accurately described by means of mathematics, between relevant and irrelevant groups of documents as shown in Figure2, in which the "+" denotes the relevant documents and the "-" denotes the irrelevant ones, because it is almost impossible to define a curve for relevancy separation with any exact math equation as there are always many strange cases of unexpected or irregular data points. Even existing similar boundary, it is still not easy to be applied to the prediction of the incoming testing documents because of the different

conditions of different types of testing document sets [38]. In order to deal with the probable uncertainty which is difficult to be solved through traditional text classification way, the proposed relevancy prediction model tries to indirectly achieve the final purpose by partitioning the result list into different three regions for further processing and refinement by stepwise. In this paper, we propose the model for dealing with the uncertain boundary to improve the performance of binary text classification. It aims to form and train the decision boundary through partitioning the training samples into three regions (the positive region (POS), boundary region (BND), and negative region (NEG)) in order to assure the discrimination of extracted knowledge for describing relevant and irrelevant information (see Section 3). The proposed approach iteratively enhances the certainty of the two regions representing relevant and irrelevant objects, and absorbing and resolving the uncertain objects in the third region BND so as to make the knowledge on document relevancy and irrelevancy more precise and unambiguous. It starts from calculation of two main centroid vectors C_P and C_N by clustering the relevant and irrelevant training subsets, and further regroups the training samples into three regions using the two centroid vectors at basal level, with all the indeterminate objects collected into a boundary region BND, the objects with most relevant possibility to the topic stored into the POS region, and those with most irrelevant possibility to the topic collected into the NEG region.

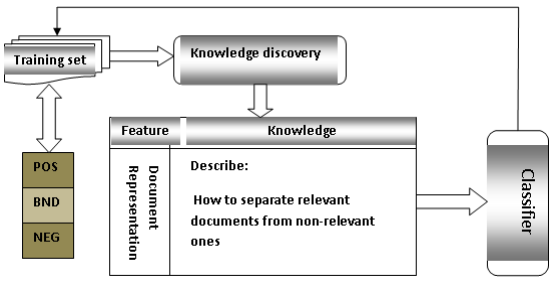

Figure 1: Schematic diagram of classifier training process

However, there must be some documents which are close to both of the two centroid vectors so as not easy to be separated clearly, and the pair of centroid vectors are not accurately located and usually closely spaced at beginning of training process. It is the key issue to approximately optimize the two pairs of centroid vectors gradually and use them for more precise text classification further, thus how to realize and improve the proposed strategy to reach the purpose is the main problem. Under such situation, one clear decision boundary is not easy to be gained for separating the documents as expected. Therefore, it is more practical to find uncertain boundary enclosed by two lines that can at least separate most of the relevant documents from the irrelevant ones during the training process, during which the uncertain or indeterminate documents are gradually absorbed into BND regions and the other two regions including POS and NEG will be enhanced the certainty, as shown in Figure2. Through the above training process, it filters as many uncertain objects gradually and save them into BND region, and makes the other two regions POS and NEG of greater certainty. During the training process the two main centroid vectors C_P and C_N and two other auxiliary centroid vectors B_P and B_N formed from the BND region are expected to be trained and optimized successively in the multi-learning process to reach the optimal condition. Figure 1 demonstrates the overall process of the classifier training. Simultaneously, the knowledge is also proposed to be updated and ultimately used for predicting the relevancy of each incoming document to the same topic so that the polarity prediction accuracy of the incoming documents will be improved. Development of vector space theory make it possible to represent and operate the documents in the type of vectors[5]. Although Rocchio classification[27] also involves the operation of centroids, the centroids have not been optimized through further learning process. We also analyze six situations based on which six decision rules are generated correspondingly to help make polarity predictions for incoming documents (see Section 4).

We have completed two series of tests based on different features including TF*IDF [28] and BM25 [31] respectively. A large number of experiments have been conducted based on the proposed approach for text classification using two standard datasets: R-CV1 [22] and Reuters21578, including the comparison analysis among the proposed model and seven other state-of-the-art baseline models (see Section 5). The experimental results show that the proposed model can significantly improve the performance of text classification in the measures of F_1 and *Accuracy*.

The evaluation of the text classification is another key issue that the paper addresses. We have chosen F_1 and *Accuracy* as the key evaluation measures. *Accuracy* reflects the accuracy degree of relevancy prediction for both relevant and irrelevant documents, and can be very high even when the number of relevant documents is zero as the number of relevant documents is usually quite low because the datasets with imbalanced class structure are used in most cases, but F_1 is an integrated, comprehensive assessment measure so as to be able to better reflect the real improvement situation of the classifier than *Accuracy*. Therefore, compared with *Accuracy*, the F_1 measure is emphasized and used for both the performance assessment of the proposed model and comparison analysis with the baseline models. Therefore, the proposed model aims to pursue substantial improvement on F_1 with the *Accuracy* guaranteed not to be reduced. Suppose we do the testing of binary text classification based on the usual datasets with imbalanced number of relevant and irrelevant targeted documents, the *Accuracy* measure may produce misleading results and is not able to reflect the real improvement degree, because even all the relevant documents are wrongly predicted as irrelevant, the *Accuracy* value will not be subjected to a big negative effect and can still be very high because of the quite low proportion of the true relevant documents. However, the calculation of F_1 depends on two factors, the *Precision* and *Recall* which can together reflect the real situation of relevant and irrelevant ratio and their improvement degrees in the testing process.

In this paper, section 2 introduces the related technologies and algorithms in text classification area; there is a detailed description of the general idea of the proposed successive approximation approach and its implementation process along with different algorithms for each step including centroid generation and training in section 3, centroid optimization in section 4, feature updating and performance improvement by Cosines laws and statistical method derived from Standard Deviation theory in section 5. The related evaluation metrics, datasets, baseline models and the experiment results are introduced to help show the

effectiveness of the model development and improvement process in section 5. Section 6 concludes the whole paper.

2 Related Work

Relevance prediction is a big research issue [17, 22] for text analysis and classification, which mainly discusses how to predict a document's relevance to a user or a category. However, the knowledge uncertainty caused by the usual knowledge mining techniques, document representation through traditional feature selection ways and traditional classification algorithms are not effective for solving relevance prediction issue because relevance is a single class problem [12].

Text classification is the process of classifying textual data into predefined categories by using classifiers learned from training samples. Text classification involves many key technologies which have certain relations with the topic and most possibly contribute to the core issues discussed in this paper. As one crucial technique of text classification, feature selection and its related methods are reviewed firstly. Then comes the analysis of some popular text classification technologies, especially some major algorithms related to this project. To date, many text classifiers such as AdaBoostM1, J48, Instance-Based Learning, kNN, Naive Bayes, SVM and Rocchio have been developed.

Document representation is one of the most important steps for text classification, in which related documents are represented by single or multiple informative features to ease the automatic operation of the documents in the subsequent steps. Feature selection can increase the performance of text classification and decrease computational complexity by eliminating noise features [2]. Feature selection is one of the important steps for text classification [30] which is the task of assigning documents to predefined classes. Feature selection plays a significant role in document representation for the purpose of text classification because a document vector is composed of a set of weighted features, and the feature number and feature quality affect the performance of text classification. The features can be simple structures (words), complex linguistic structures, statistical structures, supported information, named entities, etc. in the document. Feature selection aims to help build up the documents' vectors by selecting a subset of the key features for describing all the related documents, and remove irrespective or noise features according to corpus statistics to increase the scalability, efficiency and accuracy of a text classifier. The process of feature selection is based on a feature weighting function. A feature weighting function indicates the correlation degree of the features represented by the feature occurrences in a document and reflects the importance of the features to the document. A number of popular term weighting functions have been developed and used such as $tf*idf$ (term frequency inverse document frequency) [28], Latent Semantic Analysis (LSA) [9], Probabilistic LSA (pLSA) [13], Latent Dirichlet Allocation (LDA) [3], Chi-Square [35, 34], Information Gain [19, 35, 34], Mutual Information [11, 20], semantic structure [29], NGL coefficient [23], belief revision method [18], relevance frequency (RF) [16], pattern deploying method [32], Rocchio algorithm, Okapi BM25[26], and distributional feature [33].

Vector space model is an algebraic model for representing text documents as vectors of identifiers. Under such model, the documents are required to be represented in vectors, for example:

$$d_n = (t_{1n}, t_{2n}, \ldots, t_{mn})$$

Where d_n refers to the name of any text document, and t_{in} refers to the feature weight of any selected feature for document representation. **TF*IDF** is the basic and most effective way to calculate the feature weights. TF means the term frequency in the document, and IDF means the inverse document frequency. This method is commonly applied to weight each term in the document, which means it captures the relevancy among terms, documents and certain categories [1]. The classic formula of **TF*IDF** used for term weighting is described by the following equation:

$$w_{ij} = tf_{ij} \times log(\frac{N}{df_i}) \quad (1)$$

where w_{ij} denotes the weight of term i in document j, N denotes the total number of documents in the document set, tf_{ij} means the occurrence frequency of term i in document j, and df_i means the document frequency of term i in the document set, which represents the number of documents where a term occurs in the whole document set. It has been proven that the TF*IDF scheme is extraordinarily robust and difficult to be beaten, even by much more models and theories worked out carefully [25].

BM25 [31] is a well-known probabilistic scoring function for feature selection. From the experiments completed on the proposed model in the paper, it is found that the BM25 performs better than TF*IDF. We use the scoring function to estimate the weight of term t extracted from relevant documents as follows:

$$W(t) = \frac{tf \cdot (k_1 + 1)}{k_1 \cdot ((1-b) + b\frac{DL}{AVDL}) + tf} \cdot \log \frac{\frac{(r+0.5)}{(n-r+0.5)}}{\frac{(R-r+0.5)}{(N-n-R+r+0.5)}} \quad (2)$$

where N is the total number of training documents; R is the number of relevant documents; n is the number of documents which contain term t; r is the number of relevant documents which contain term t; tf is the term frequency; DL and $AVDL$ are the document length and average document length, respectively; and k_1 and b are the experimental parameters. We also use the BM25 with the parameters tuned in [39] (i.e.,$k_1 = 1.2$ and $b = 0.75$).

Classification algorithm is another key component of a text classifier. The document classification can usually be categorized in three ways including unsupervised, supervised and semi supervised methods. In the past few years, lots of classification algorithms have been developed for classifying electronic documents. We main focus on the supervised classification methods such as Naive Bayes, Support Vector Machines (SVM), Rocchio and k-Nearest Neighbour (kNN). Support Vector Machine (SVM) can be applied to classify both linear and nonlinear data. The algorithm of SVM transforms the training samples to a higher dimensional feature space through a nonlinear mapping process. SVMs is relatively successful, but the complexity of the training and categorizing algorithm cause high time and memory consumption during the training and classifying stages. [15]. Also, SVMs is highly dependent on the size of the training samples, they are not the best practice in large-scale data mining such as pattern recognition and machine learning.[36]

Bayesian classifiers can be regarded as probabilistic models. Bayesian approaches to supervised learning generally utilize Bayes law to calculate the reverse probability of the model parameters given function

input-output examples, which known as training samples [6]. Naive Bayes is similar to independent events in mathematics. It assume that all the features in a certain class are irrelevant to each other and one feature does not affect other features. These assumption bring the computation of Bayesian classifiers more efficiency but with the cost of limited applicability.

Rocchio algorithm of classification is a vector space model for text classification presented by Rocchio in 1971 [37]. This method merges relevance feedback information into the vector space model in information retrieval by building prototype vector for each class with training samples. This method is easy to implement as well as efficient in computation, but it has a potential disadvantage that the performance will be reduced when the documents belonging to a category naturally form separate clusters.

The k-Nearest Neighbour algorithm is a statistical approach to realize text classification. It ranks a document's nearest neighbours by calculating the degree of similarity between the documents and uses the top k ranked neighbours to predict the polarity of a new document. Generally, this method is efficient, but in [10] it points out that the *Accuracy* of kNN classifier depends on the value of k in turn affected by the training samples.

3 Theoretical Basis and Model Construction

Let CF be a binary text classifier, D be a training set in which all documents are labelled as either relevant D^+ or irrelevant D^-, and $F = \{f_1, f_2, \ldots, f_n\}$ is a set of terms (e.g., keywords) extracted from D. For each document $d \in D$, it can be represented as a vector $\vec{d} = (w(f_1), w(f_2), \ldots, w(f_n))$ by using the terms of F and their weights expressed by the term weighting function w.

Based on the above definitions, classifier $CF : D \longrightarrow \{R, iR\}$ will partition D into two groups: the possible relevant group R and possible irrelevant group iR for a given decision boundary or a threshold. However, it is hard to find a clear boundary by any text classifier, between relevant and irrelevant documents. Therefore, normally we have $D^+ \neq R$ and $D^- \neq iR$.

For modelling the uncertainty between relevant and irrelevant documents, we extend the classifier $CF \Longrightarrow CF'$, where $CF' : D \longrightarrow \{POS, NEG, BND\}$ is called an extended classifier, which is able to classify $d \in D$ into three regions: positive (POS, possible relevant), negative (NEG, possible irrelevant) and boundary (BND, uncertain) regions by the following definitions:

Definition 1. *If ($CF(d)$ = "R" and $d \in D^+$) Then $CF'(d)$ = "POS"; Else, If ($CF(d)$ = "iR" and $d \in D^-$) Then $CF'(d)$ = "NEG"; otherwise, $CF'(d)$ = "BND".*

Based on the above definitions, some properties about the three regions can be derived as follows:

Property 1. *If $d \in POS$ then $d \in D^+$.*

Property 2. *If $d \in NEG$ then $d \in D^-$.*

Property 3. *If $d \in D^+$ and $d \in BND$ then $CF(d)$ = "iR".*

Property 4. *If $d \in D^-$ and $d \in BND$ then $CF(d)$ = "R".*

The boundary region BND includes the uncertain decisions for relevant documents and irrelevant documents, which can be further divided into two groups: $B^+ = BND \cap D^+$ and $B^- = BND \cap D^-$.

If every document $d \in D$ is represented as a vector of term-weights, the four groups (POS, NEG, B^+ and B^-) can generate 4 centroid vectors. Let C'_P be the centroid vector of POS and C'_N be the centroid vector of NEG, B_P be the centroid vector of B^+ and B_N be the centroid vector of B^-. We also assume there is a central line (a decision boundary) between R and iR. Theorem 1 indicates the relations between them.

Theorem 1. *Let $B^+ = BND \cap D^+$ and $B^- = BND \cap D^-$, all the documents in B^+ must be below the central line, whereas all the documents in B^- must be above the central line.*

Proof. **If** there is a document $d \in B^+$, **then** according to the definition of B^+, it should be $d \in D^+$, **suppose** it is above the central line, i.e., $CF(d)$ = "R"; then it must be $d \in POS$ by Definition 1, that is against the property of B^+: $d \in BND$, **therefore** d is below the central line. **In the same way,** we can prove that any document $d \in B^-$ must be above the central line. □

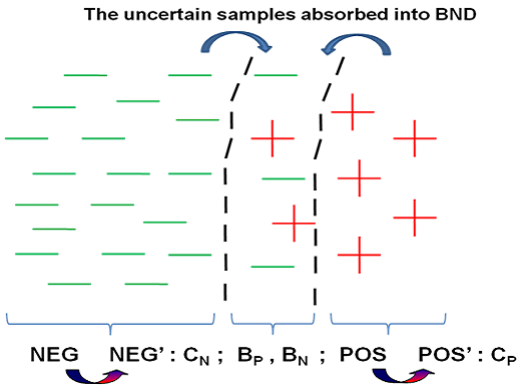

Figure 2: Training process for modelling uncertain boundary

4 Decision Rules Generation

The extended classifier CF' firstly generates two basic centroid vectors (C'_P and C'_N) to represent relevant and irrelevant information; however, there is a uncertain boundary (B^+ and B^-) between C'_P and C'_N. To assure the discrimination of C'_P and C'_N for describing relevant and irrelevant information, we propose a optimization process to iteratively update two basic centroid vectors. The process make the boundary region gradually absorb as many uncertain training documents as possible so that the two basic centroid vectors are moving away from each other accordingly until the distance between them no longer changes.

When the extended classifier CF' is produced, it is then used back to classify the training set again to update POS, NEG and BND. More uncertain documents will possibly be found and put into BND. Figure 4 shows the result example in which we can clearly see that C'_P and C'_N have been changed to C_P and C_N. The larger the gap between C_P and C_N is, the easier it would be made to separate documents apart into binary categories by comparing their distances to

the centroid vectors. In the process of this case, the size of the boundary region keeps growing and the distance between the centroid vectors C'_P and C'_N keeps increasing synchronously to reach the maximum when the training process ends.

A schematic diagram on the training process is given in Figure 3 which roughly demonstrates all the steps of the whole training and optimization process.

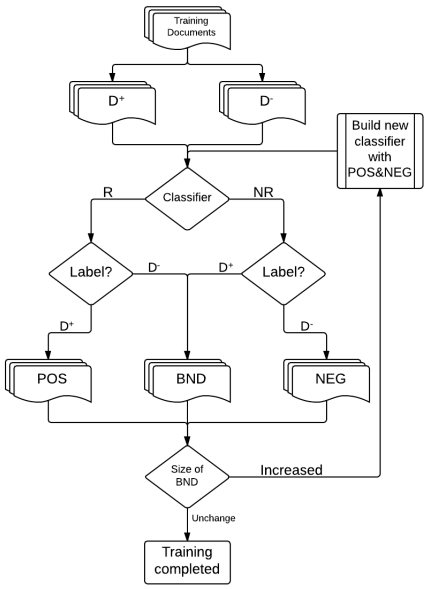

Figure 3: Centroid training and optimization process

Figure 4 shows the relations between centroid vectors, where C_P and C_N are the optimization of vectors C'_P and C'_N. We will discuss them in next section.

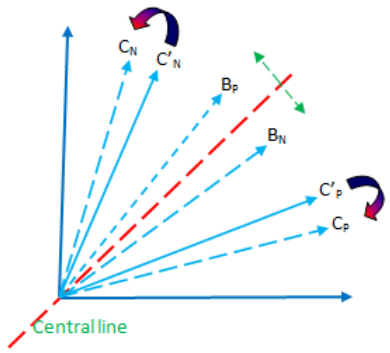

Figure 4: Four kinds of centroid vectors

For a given incoming document u, it will be compared with the two centroid vectors C_P and C_N in order to decide its relevance by using the central line and the Euclidean distance. However, the performance is poor because of the uncertain boundary. In this section, we present six decision rules to improve the performance.

Let F be the selected feature set, and $\vec{u} = (w_1, w_2, ..., w_{|F|})$ be the vector of document u and $\vec{v} = (w'_1, w'_2, ..., w'_{|F|})$ be a centroid vector. We use the following definitions to measure the distance between documents and centroid vector.

$$dis(\vec{u}, \vec{v}) = \sqrt{\sum_{j=1}^{|F|}(w_j - w'_j)^2} \quad (3)$$

$$meanDis(v) = \frac{k}{|D_v|}\sum_{d \in D_v} dis(\vec{d}, \vec{v}) \quad (4)$$

where $D_v = D^+$ if $\vec{v} = C_P$, else $D_v = D^-$ if $\vec{v} = C_N$, and k is an experiment parameter.

To predict the polarity of each incoming document $\vec{u}$, we need to understand the possible relationship between u and centroid vectors. We describe the relationship in trigonometry and use the law of cosines to display the relations. Figure 5 shows the relationship, where the round dot denotes u, "+" denotes C_P and "-" denotes C_N.

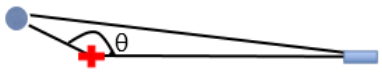

Figure 5: Example of cosines law

Below is the formula of the law of cosines:

$$\cos\theta = \frac{dis(u, C_P)^2 + dis(C_P, C_N)^2 - dis(u, C_N)^2}{2 \times dis(u, C_P) \times dis(C_P, C_N)} \quad (5)$$

Based on the law of cosines and the positions of C_P, B_N, B_P and C_N, we have six scenarios (rules) that cover all typical spatial location of the incoming document vectors for relevant analysis and decision-making of polarity prediction, as illustrated in Figure 6, where the dotted line refers to the central line; and u_1, u_2, u_3, u_4, u_5 and u_6 denote the six types of incoming document vectors in different six situations corresponding with different orientation and distance to centroid vectors, three of which are located at the left side of the central line and closer to C_P, and others are closer to C_N.

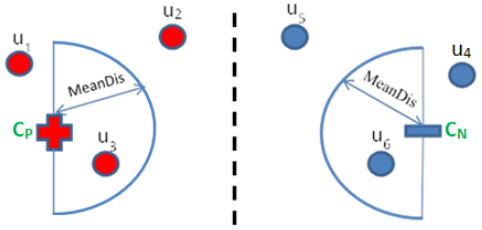

Figure 6: Six scenarios for polarity prediction

The following are the six decision rules (scenarios) for predicating the polarity (relevant or irrelevant) of each incoming document.

Rule 1 - for document u_1:

u_1 is on the left side of positive centroid, it means that u_1 is close to positive centroid and far away from the negative centroid ($dis(u_1, C_P) << dis(u_1, C_N)$). If $\cos\theta \leq 0$ ($\theta \geq \frac{\pi}{2}$, an obtuse triangle) document u_1 is predicted as relevant.

Rule 2 - for document u_2:
u_2 locates between the centroid vectors C_P and C_N but around the centroid vector B_N, specifically between B_N and the central line. Under such circumstance, we can also know that $dis(u_2, C_P) < dis(u_2, C_N)$ and θ is smaller than $\frac{\pi}{2}$, but the $dis(u_2, C_P)$ is greater than $meanDis$. Then u_2 is predicted as irrelevant.

Rule 3 - for document u_3:
u_3 is similar to u_2, but it actually locates between C_P and B_N, and the distance $dis(u_3, C_P)$ is not greater than Dis. In this case, it has a greater chance that u_3 is relevant. Therefore, u_3 is predicted as relevant.

Rule 4 - for document u_4:
u_4 is quite similar with u_1, however, it is on the right side of the negative centroid, showing that u_4 is close to C_N and far away from C_P ($dis(u_4, C_N) << dis(u_4, C_P)$). Therefore, it is predicted as irrelevant.

Rule 5 - for document u_5:
u_5 is quite similar with u_2, so the similar decision making can also be applied for it. So, it is predicted as relevant.

Rule 6 - for document u_6:
u_6 is quite similar with u_3, but the document u_6 locates between C_N instead of C_P, and B_P instead of B_N, and the distance $dis(u_6, C_P)$ is not greater than meanDis. Therefore, u_6 is predicted as irrelevant.

5 Experiments and Evaluations

5.1 Data Set

We used two popular datasets to test the proposed model: RCV1 (Reuters Corpus Volume 1), a very large data collection; and Reuters-21578, a relatively small one. RCV1 consists of all and only English language stories produced by Reuter's journalists between August 20, 1996, and August 19, 1997. RCV1 includes 806,791 documents that cover a broad spectrum of issues or topics. TREC (2002) has developed and provided 50 assessor topics for RCV1. These topics were evaluated by human assessors at the National Institute of Standards and Technology (NIST). The relevance judgements of these topics on RCV1 have also been made by the NIST assessors. For each topic, a subset of RCV1 documents is divided into a training set and a testing set. RCV1 and TREC assessor topics are standard data collections [22].

Reuters-21578 (R21578) corpus is a widely used test collection for text mining and information retrieval researches. The data was originally collected and labelled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system[1]. In this experiment, we picked up the set of 10 classes for testing since the class distribution for documents is too skewed. According to Sebastiani's convention [8], it was called the set R8 because two classes *corn* and *wheat* are intimately related to the class *grain*, and they were appended to class *grain*. In our experiments, each class is paired with other seven classes to get more testing cases (in total, we have 56 cases). For each case, documents in the class are relevant and in another class are irrelevant.

[1] Reuters-21578, http://www.daviddlewis.com/resources/

Table 1: The algorithms of the baseline models

No	Algorithm type	Classifier
1	Function based	SVM
2	Classifiers committee based	AdaBoost
3	Decision tree based	J48 ; Random Forest
4	Probabilistic based	Naive Bayes
5	Instance-based (lazy learner)	IBk (KNN)
6	Representative based	Rocchio

To avoid bias in experiments, all of the meta-data information has been ignored. Documents are treated as plain text documents. The preprocessing tasks include removing stop words from each document according to a given list of the predefined stop words, and stemming all the terms by applying the Porter Stemming algorithm.

5.2 Baseline Models

In order to make a comprehensive evaluation, we have chosen seven types of classifiers with different algorithms from total 22 models and determined them as the baseline models (see Table 1). The selected baseline models (also see Section 2) are the state of art influential ones including Support vector machine (SVM), AdaBoostM1, J48 [24], Naive Bayes [14], Random forest [4], IBk (Instance-Based Learning), Rocchio.

Precision (p), Recall(r) are two basic parameters for evaluation of the proposed model. In the paper, the effectiveness of text classification is measured by two key measures: F_1 measure and *Accuracy* (*Acc*). F_1 is stressed as it is one of the most important metrics of comprehensive assessment [30].

$$F_1 = \frac{2PR}{P+R}\ , \quad F_1^M = \frac{\sum_{i=1}^{|\mathcal{C}|} F_{1,i}}{|\mathcal{C}|}$$

where F_1^M is the macro average of F_1 for all the tested topics, and $F_{1,i}$ is the F_1 of topic i. For the calculation of *Accuracy*,

$$Acc = \frac{TP+TN}{TP+FP+TN+FN}\ , \quad Acc^M = \frac{\sum_{i=1}^{|\mathcal{C}|} Acc_i}{|\mathcal{C}|}$$

where Acc^M is the macro average of *Accuracy* for all the the tested topics, and Acc_i is the *Accuracy* of topic i.

5.3 Experiment Results

The comparison between the proposed model (UBD) and the baseline models has been completed mainly by the two measures of F_1 and *Accuracy*. UBD is compared with seven baseline models as shown in Table 2 based on RCV1 Dataset and Table 3 based on R21578 Dataset. In Table 2, we found that the proposed model has got an average increase of 5.48% for *Accuracy* and 43.36% for F_1 compared with the other seven baseline models. The *Accuracy* value got by the proposed model exceeds SVM model which has the highest *Accuracy* value in all the baseline models, and the F_1 value has also been extremely improved by the proposed model at 116.70% compared with SVM model. In Table 3, we found that the proposed model has gained an average increase of 5.82% for *Accuracy* and 21.85% for F_1 compared with the other seven baseline models.

Table 2: The results of experiments on RCV1

No	Models	F_1	Accuracy
1	SVM	19.39%	85.45%
2	AdaBoostM1	35.46%	84.54%
3	J48	34.25%	82.85%
4	NaiveBayes	26.87%	81.62%
5	RandomForest	27.60%	84.79%
6	IBk	37.22%	82.26%
7	Rocchio	33.86%	70.13%
8	CVTO-SD-BM25-TF	**42.02%**	**85.79%**
9	Average %chg	**43.36%**	**5.48%**

Table 3: The results of experiments on R21578

No	Models	F_1	Accuracy
1	SVM	60.96%	85.46%
2	AdaBoostM1	56.79%	81.26%
3	J48	64.12%	85.39%
4	NaiveBayes	79.54%	82.49%
5	RandomForest	66.01%	85.25%
6	IBk	75.10%	86.45%
7	Rocchio	69.39%	71.16%
8	CVTO-SD-BM25-TF	**81.20%**	**86.95%**
9	Average %chg	**21.85%**	**5.82%**

Table 2 and Table 3 indicate that the proposed model has the highest score in both F_1 and *Accuracy* on two datasets, especially in F_1 that best reflects the real situation of text classification performance. Therefore, the proposed partitioning approximation approach has gained the best performance on RCV1 and R21578 compared with all the collected seven influential baseline models.

6 Conclusion

This paper proposed the method for dealing with uncertain decision boundary for finding relevant information. The experimental results show that the proposed model can significantly improve the performance of binary text classification in both F_1 and *Accuracy* compared with seven other influential baseline models. The proposed model is promising and has the following contributions:

- Developed a model to understand the difference between relevant and irrelevant information by dividing training documents into three regions to reduce the impact of the uncertain information for text classification.
- Presented six decision rules to improve the performance of binary text classification on F_1 measure and *Accuracy*.

References

[1] B. Baharudin, L. H. Lee, and K. Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20, 2010.

[2] R. Bekkerman and M. Gavish. High-precision phrase-based document classification on a modern scale. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 231–239. ACM, 2011.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[6] J. L. Carroll. *A bayesian decision theoretical approach to supervised learning, selective sampling, and empirical function optimization*. PhD thesis, Brigham Young University, 2010.

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

[8] F. Debole and F. Sebastiani. An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and technology*, 56(6):584–596, 2005.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[10] H. Doshi and M. Zalte. Comparison of supervised learning techniques for binary text classification. *IJCSIS) International Journal of Computer Science and Information Security*, 10(9), 2012.

[11] S. T. Dumais, J. C. Platt, D. Hecherman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM*, pages 148–155, 1998.

[12] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, pages 50–57. ACM, 1999.

[14] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[15] L. Khan, M. Awad, and B. Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal—The International Journal on Very Large Data Bases*, 16(4):507–521, 2007.

[16] M. Lan, C. L. Tan, J. Su, and Y. Lu. Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transations on Pattern Anaalysis and Machine Intelligence*, 31:721–735, April 2009.

[17] R. Y. Lau, P. D. Bruza, and D. Song. Towards a belief-revision-based adaptive and context-sensitive information retrieval system. *ACM Transactions on Information Systems (TOIS)*, 26(2):8, 2008.

[18] R. Y. K. Lau, P. Bruza, and D. Song. Belief revision for adaptive information retrieval. In *SIGIR*, pages 130–137, 2004.

[19] D. D. Lewis. Feature selection and feature extraction for text categorization. In *Speech and Natural Language Workshop*, pages 212–217, San Francisco, 1992.

[20] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

[21] Y. Li, A. Algarni, and N. Zhong. Mining positive and negative patterns for relevance feature discovery. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 753–762. ACM, 2010.

[22] Y. Li, A. Algarni, and N. Zhong. Mining positive and negative patterns for relevance feature discovery. In *Proceedings of SIGKDD*, pages 753–762. ACM, 2010.

[23] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *SIGIR*, pages 67–73, 1997.

[24] J. R. Quinlan. *C4.5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[25] S. Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[26] S. E. Robertson and I. Soboroff. The trec 2002 filtering track report. In *TREC*, volume 2002, page 5, 2002.

[27] J. Rocchio. Relevance feedback in information retrieval. *SMART Retrieval System Experimens in Automatic Document Processing*, pages 313–323, 1971.

[28] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[29] H. Schütze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *SIGIR*, pages 229–237, 1995.

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

[31] J. S. Whissell and C. L. Clarke. Improving document clustering using okapi bm25 feature weighting. *Information retrieval*, 14(5):466–487, 2011.

[32] S. Wu, Y. Li, and Y. Xu. Deploying approaches for pattern refinement in text mining. In *Proceedings of ICDM'06. Sixth International Conference on Data Mining.*, pages 1157–1161, 2006.

[33] X.-B. Xue and Z.-H. Zhou. Distributional features for text categorization. *IEEE Transactions on K*, 21(3):428 – 442, 2009.

[34] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999.

[35] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997.

[36] H. Yu, J. Yang, J. Han, and X. Li. Making svms scalable to large data sets using hierarchical cluster indexing. *Data Mining and Knowledge Discovery*, 11(3):295–321, 2005.

[37] A. Zeng and Y. Huang. A text classification algorithm based on rocchio and hierarchical clustering. In *Advanced Intelligent Computing*, pages 432–439. Springer, 2012.

[38] L. Zhang, Y. Li, C. Sun, and W. Nadee. Rough set based approach to text classification. In *2013 WI/IAT and IEEE/WIC/ACM International Joint Conferences*, volume 3, pages 245–252. IEEE, 2013.

[39] N. Zhong, Y. Li, and S. Wu. Effective pattern discovery for text mining. *Knowledge and Data Engineering, IEEE Transactions on*, 24(1):30–44, 2012.