

# An Efficient Feature Ranking Measure for Text Categorization

Songbo Tan<sup>1</sup>, Yuefen Wang<sup>2</sup> and Xueqi Cheng<sup>1</sup>

<sup>1</sup> Information Security Center, Institute of Computing Technology, Chinese Academy of Sciences, China

<sup>2</sup> Information Center, Chinese Academy of Geological Sciences, China

tansongbo@software.ict.ac.cn, tansongbo@gmail.com

## ABSTRACT

A major obstacle that decreases the performance of text classifiers is the extremely high dimensionality of text data. To reduce the dimension, a number of approaches based on rough-set theory have been proposed. However, these works often suffer from two problems: the first is that they cannot directly deal with continuous text features; the second is that they often incur considerable running time. To deal with the first issue, we make some extensions to discernibility matrix so that it can work with continuous features. To cut down running time, we employ centroids rather than examples to construct discernibility matrix, which reduce the time complexity from  $O(T^2W)$  to  $O(K^2W)$  where  $T$  denotes the size of training examples,  $K$  denotes the number of training classes and  $W$  denotes the size of vocabulary. The experimental results indicate that proposed method not only yields much higher accuracy than Information Gain when the number of selected features is smaller than 6000, but also incurs much smaller CPU time than Information Gain.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval-search process; I.2 [Artificial Intelligence]: Learning; I.5 [Pattern Recognition]: Applications

## General Terms

Algorithms, Performance, Experimentation

## Keywords

Feature Selection, Document Categorization, Information Retrieval, Machine Learning

## 1. INTRODUCTION

In recent years we have seen a tremendous growth in volumes of text documents available on the Internet. Accordingly the management and organization of text has become an important task. One main component of this task is the assignment of documents into a set of predefined categories, which is known as Text Categorization (TC). A number of machine learning

algorithms have been introduced to deal with Text Classification, such as K-nearest Neighbor [16], Centroid Classifier [7], Naive Bayes (NB) [20], Winnow [21] and Support Vector Machines (SVM) [11].

A common and often overwhelming characteristic of text data is its extremely high dimensionality. In text classification community, a document is represented by a "bag-of-words" [18], that is a vector with the same size as the vocabulary containing word frequency counts. Even a moderately-sized document collection can lead to a dimensionality in thousands [5]. As a result, high dimensionality posed an open challenge for classification algorithms. Therefore, reducing the dimensionality without sacrificing classification performance is very important for text categorization.

Most current dimension reduction approaches fall into two categories: Feature Extraction (FE) [13][14] and Feature Selection (FS) [6][8]. FE methods attempt to reduce the dimension of data by transforming original feature space to another low-dimension feature space. On the other hand, FS algorithms reduce the dimension of data by selecting features from the original space directly.

In the recent years, a number of researches about using rough-set theory to select text features have been conducted, such as the researches of Bao [4] and Chouchoulas [5]. However, these works often suffer from two problems. The first is that these methods do not make any substantial improvements to rough-set based attribute reduction so that they can not work directly with continuous features; the second is that the computational cost is in scale with the square of the size of training set so that they can not address large real-world datasets.

To address these two questions, we propose a novel feature ranking measure for feature selection. The key idea is borrowed from attribute reduction based on discernibility matrix in rough-set theory [6]. To deal with the first issue, we make some extensions to discernibility matrix so that it can work with continuous features. With the aim to decrease running time, we employ centroids rather than examples to construct discernibility matrix, which reduce the time complexity from  $O(T^2W)$  to  $O(K^2W)$  where  $T$  denotes the size of training examples,  $K$  denotes the number of training classes and  $W$  denotes the size of vocabulary.

Extensive experiments are conducted with two commonly used categorization methods, i.e., Naïve Bayes (NB) and SVM. The experimental results indicate that proposed method (called as DB2 in the rest of this paper) yields high accuracy, which is nearly the same as Information Gain (IG). When the number of selected features is smaller than 6000, DB2 can beat IG in accuracy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

Copyright 2008 ACM 978-1-59593-753-7/08/0003...\$5.00.

Furthermore, the time requirement of DB2 is about the same as that of DF, MI or CHI, but obviously smaller than that of IG.

The rest of this paper is organized as follows: rough-set theory are introduced in the next section; Section 3 describes the proposed feature ranking measure and two feature selection methods, i.e., DB1 and DB2. Section 4 presents the comparison to other related works. Experimental results are given in section 5. Finally section 6 concludes this paper.

## 2. BACKGROUND

In this section we provide some basic definitions of rough set theory. Detail description of the theory can be found in [22-23].

**Information system** is a pair  $A=(U, A)$ , where  $U$  is a non-empty, finite set of objects called the **universe** and  $A$  is a non-empty finite set of **attributes**, i.e.,  $a:U \rightarrow V_a$  for  $a \in A$ , where  $V_a$  is usually called a decision **value set** of  $a$ . For text collection, an “object” indicates one text document.

An information system  $A=(U, A \cup \{d\})$ , where  $d \notin A$ , is usually called a **decision table**. The elements of  $A$  are called the **conditional attributes** and  $d$  is called **decision attribute**. For text datasets, the conditional attributes are represented by words (or terms, features); the decision attribute indicates the class label. For example, the text “He is a manager” consists of four attributes: “he”, “is”, “a”, “manager”.

The **discernibility matrix** [17] is a symmetric  $|U| \cdot |U|$  matrix which can capture the discrimination information involved with all conditional attributes in an information system. Its entries  $c_{ij}$  can be defined as  $\{a \in A / a(x_i) \neq a(x_j)\}$  if  $d(x_i) \neq d(x_j)$ ,  $\emptyset$  otherwise.

We take the following decision table as an example. The letters  $a_1, a_2, a_3$  denote condition attributions and the letter  $d$  denotes decision attribute.

**Table 1: The Simple Decision Table**

Attribute \ Object	$a_1$	$a_2$	$a_3$	$d$
$u_1$	True	True	Very High	1
$u_2$	False	True	Normal	0
$u_3$	False	False	High	0
$u_4$	False	True	Very High	1

According to preceding definition of discernibility matrix, we can construct the discernibility matrix of Table 1 as Table 2.

**Table 2: The Discernibility Matrix of Table 1**

	$u_1$	$u_2$	$u_3$	$u_4$
$u_1$	0	$a_1a_3$	$a_1a_2a_3$	0
$u_2$		0	0	$a_3$
$u_3$			0	$a_2a_3$
$u_4$				0

In Table 2, the item “ $a_1a_3$ ” indicates that both conditional attribute “ $a_1$ ” and “ $a_3$ ” can discriminate object “ $u_1$ ” and object “ $u_2$ ” separately. Obviously, the more times an attribute occurs in a

discernibility matrix, the stronger discriminating power it has. As a result, the occurrence frequency is often adopted as heuristic information for attribute (feature) reduction.

The goal of attribute reduction based on discernibility matrix is: to pick out a subset of attributes that contains at least one attribute of each item (such as  $a_1a_3$  or  $a_1a_2a_3$ ) in discernibility matrix, that is, to select a subset of attributes which can discriminate all couple of objects with different class label. To achieve this goal, researchers have developed a number of attribute-reduction algorithms based on various heuristic information [9][10][12]. Up to now, the most popular heuristic information is the attribute’s occurrence number in discernibility matrix.

## 3. PROPOSED METHODS

The above attribute-reduction algorithms based on attribute occurrence frequency in discernibility matrix can also be viewed as a kind of feature selection method. Now we consider introducing it to text feature selection. First we need to generalize discernibility matrix from discrete attributes to continuous features. For the sake of convenience, we transfer discernibility matrix (Table 2) into the following form,

**Table 3: The Transferred Discernibility Matrix**

Object Couple		Class label is the same?	$a_1$	$a_2$	$a_3$
$u_1$	$u_2$	No	1	0	1
$u_1$	$u_3$	No	1	1	1
$u_1$	$u_4$	Yes	0	0	0
$u_2$	$u_3$	Yes	0	0	0
$u_2$	$u_4$	No	0	0	1
$u_3$	$u_4$	No	0	1	1

The third column indicates whether two objects have the same class label. The last three columns are equivalent to discernibility matrix (Table 2): “1” denotes that the column attribute is contained by corresponding item in discernibility matrix; “0” denotes that the column attribute is not contained by corresponding item. For example, “101” of the row “ $u_1, u_2$ ” means “ $a_1a_3$ ”. Substantially, this so kind of expression (for discernibility matrix) indicates one kind of Boolean distance,

$$dist(a_t(x), a_t(y)) = \begin{cases} 1 & a_t(x) \neq a_t(y) \\ 0 & a_t(x) = a_t(y) \end{cases} \text{ if } d(x) \neq d(y).$$

where  $t$  is the index of condition attributes. The significance (frequency) of each attribute can be computed by summing the corresponding column. The larger the sum is, the more example-couples the attribute can discriminate, that is, the stronger the discrimination is. For example, according to Table 3, the significance of  $a_3$  is 4; the significance of both  $a_1$  and  $a_2$  is 2. Hence the distinguishing ability of  $a_3$  is the strongest.

The transferred discernibility matrix (such as Table 3) can be very easily generalized to continuous attributes only if we adopt one kind of distance function that can handle continuous attributes, such as absolute distance,

$$dist(a_t(x), a_t(y)) = \begin{cases} |a_t(x) - a_t(y)| & a_t(x) \neq a_t(y) \\ 0 & a_t(x) = a_t(y) \end{cases} \text{ if } d(x) \neq d(y),$$

that is

$$\text{dist}(a_t(x), a_t(y)) = |a_t(x) - a_t(y)| \text{ if } d(x) \neq d(y).$$

Let's see an example. If we substitute the attribute value "True", "False", "Normal", "High" and "Very High" of Table 1 with "0.9", "0.0", "0.1", "0.5" and "0.9" respectively, we obtain the following continuous decision table,

**Table 4: The Simple Continuous Decision Table**

	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	d
u <sub>1</sub>	0.9	0.9	0.9	1
u <sub>2</sub>	0	0.9	0.1	0
u <sub>3</sub>	0	0	0.5	0
u <sub>4</sub>	0	0.9	0.9	1

Then we can construct the transferred discernibility matrix of Table 4 according to above definition of absolute distance,

**Table 5: The Transferred Continuous Discernibility Matrix**

Object Couple		Class label is the same?	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>
u <sub>1</sub>	u <sub>2</sub>	No	0.9	0	0.8
u <sub>1</sub>	u <sub>3</sub>	No	0.9	0.9	0.4
u <sub>1</sub>	u <sub>4</sub>	Yes	0	0	0
u <sub>2</sub>	u <sub>3</sub>	Yes	0	0	0
u <sub>2</sub>	u <sub>4</sub>	No	0	0	0.8
u <sub>3</sub>	u <sub>4</sub>	No	0	0.9	0.4

According to above transferred continuous discernibility matrix, we can figure out the significance of three attributes: a<sub>3</sub> is 2.4; both a<sub>2</sub> and a<sub>1</sub> is 1.8. At this moment, the description of discrimination becomes more precise in number value. For example, the discrimination of a<sub>3</sub> is still the strongest. Straightforward, the discrimination of attributes can be expressed as following,

$$\text{Dist}(t) = \sum_{j=1}^{U-1} \sum_{k=j+1}^U |a_t(x_j) - a_t(x_k)| \cdot \delta(d(x_j), d(x_k)) \quad (1)$$

where

$$\delta(p, q) = \begin{cases} 1 & p \neq q \\ 0 & p = q \end{cases}$$

According to formula (1), we design the first feature selection method. Since the discriminative power of attributes is calculated via distance function, we call this algorithm as Distance Based Feature Selection (DB1). Note that "FeaNum" in following figures denotes the predefined size for reduced feature-set.

- 
- (1) Load the text data and parameter FeaNum;
  - (2) For each word, calculate its discrimination using formula (1);
  - (3) Sort the discrimination of all words;
  - (4) Pick out FeaNum words.
- 

**Figure 1: The outline of algorithm DB1**

Now, let's analyze its complexity of time. The step (2) can be done in O(T<sup>2</sup>W); the step (3) consumes O(Wlog(W)). Therefore the total time requirement is O(T<sup>2</sup>W+Wlog(W)).

Obviously, the time requirement of DB1 scales squarely with the number of examples. Due to the rapid growth in volumes of text document, the practicability of algorithm DB1 is limited to a great extent. With the aim to address this issue, we consider taking some techniques to reduce its complexity of time.

One kind of very straightforward solution is to employ the centroids rather than examples to calculate the discrimination of attributes. The centroid C<sub>k</sub> (of class k) is computed by the average centroid formula,

$$C_k = \frac{1}{\text{Count}(S_k)} \sum_{d \in S_k} d$$

where *d* denotes the document and *Count*(Z) indicates the cardinality of set Z. It is worth noticing that the decision attributes are just the labels of training classes. Hence we can write down the centroid-based discrimination-calculation formula as following,

$$\text{Dist}(t) = \sum_{k=1}^{K-1} \sum_{l=k+1}^K |a_t(C_k) - a_t(C_l)| \quad (2)$$

As such, according to formula (2), we design the second text feature selection method. For the sake of convenience, we call this algorithm as Distance Based Feature Selection (DB2).

The computation of centroid vector can be done O(TW). The step (3) costs O(K<sup>2</sup>W). In total, the time requirement of DB2 is O(TW+K<sup>2</sup>W+Wlog(W)). As to algorithm DB1, the computation of discrimination is greatly cut down.

- 
- (1) Load the text data and parameter FeaNum;
  - (2) For each class, compute centroid vector;
  - (3) For each word, calculate its discrimination using formula (2);
  - (4) Sort the discrimination of all words;
  - (5) Pick out FeaNum words.
- 

**Figure 2: The outline of algorithm DB2**

## 4. COMPARISON TO RELATED APPROACHES

So far as we know, a few researches about using rough-set theory to select text features have been conducted. In this section, we mainly review these researches of Chouchoulas [3-4] and Bao [1-2].

Chouchoulas [3] attempted to address the high-dimensionality problem of text data by rough-set based attributes reduction. He developed an email classification system, which combined the strength of keyword acquisition, rough-set based dimensionality reduction and traditional classifier. The keyword acquisition algorithm employed one of the four different weighting methods: existential, frequency, TFIDF (term frequency-inverse document frequency), and FRM (fuzzy relevance metric). In classification stage, he used one of the following three classifiers: Boolean Inexact Classifier, Vector Space Classifier, and Fuzzy Reason Classifier. It is worth mentioning that before reduction of attributes, the dataset should be quantised.

Bao [1-2] proposed two rough-set based classifiers for text collection. In 2002, he designed a hybrid algorithm by integrating LSI (Latent Semantic Indexing). After keyword acquisition, LSI

is employed to “group” keywords, and then the traditional rough-set classifier is used to produce decision rules. According to rough-set theory, the traditional rough-set classifier includes attribute reduction (decreasing the dimensionality) as well as value reduction (decreasing the rules). This technique is verified using six small datasets downloaded from Yahoo.

In 2003, he published another rough-set based algorithm. In this method, he generated several reduction bases for one dataset, hoping that the combination of multiple reduction bases result in better performance. To test this approach, six very small corpora are selected from Reuter-21578.

Compared with above related work, we could write down the differences of our work:

First, related works don’t make any substantial improvements or extensions to rough-set based attribute reduction. On the contrary, our methods make some extension to discernibility matrix so that it can work directly with continuous features without quantisation.

The second improvement is to use centroids instead of examples to construct discernibility matrix, which considerably cuts down the running time. This improvement enables our method to perform very well in large real-world datasets. However, related works are conducted on very small datasets, especially the Bao’s work. Moreover, the category number is severely limited.

## 5. EXPERIMENTAL RESULTS

### 5.1 Data Collections

In our experiment, we use two English corpora: Industry Sector<sup>1</sup> and TDT-5<sup>2</sup>.

**Sector-48** The Industry Section dataset is based on the data made available by Market Guide, Inc. ([www.marketguide.com](http://www.marketguide.com)). The set consists of company homepages that are categorized in a hierarchy of industry sectors, but we disregard the hierarchy. We use a subset called as Sector-48 consisting of 48 categories and in all 4,581 documents. The size of category ranges from 39 to 105.

**TDT-5** TDT-5 is the NIST Topic Detection and Tracking text corpus version 1.1 released in September 10, 2004. This corpus contains news data collected daily from news sources in three languages (American English, Mandarin Chinese and Arabic), over a period of six months (April 1–September 30 in 2003). The documents were manually annotated using 250 target topics, approximately 25% of the topics are monolingual English (ENG), 25% are monolingual Mandarin Chinese (MAN), 25% are monolingual Arabic (ARB), and 25% are multilingual. We selected the English documents having annotated topics. The resulting dataset contains 126 categories and in total 6,364 documents. The size of category ranges from 1 to 809. There are 55 categories whose cardinality is smaller than 10, and the top 10 categories contain about 51% of total training examples.

1 <http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/wkwb>.

2 <http://www ldc.upenn.edu/Projects/TDT5/>.

### 5.2 The Performance Measure

We employ the F1 measure [19] to evaluate the performance of text classifiers. This measure combines recall and precision in the following way:

$$\text{Recall} = \frac{\text{number of correct positive predictions}}{\text{number of positive examples}}$$

$$\text{Precision} = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}}$$

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})}$$

For ease of comparison, we summarize the F1 scores over the different categories using the Micro- and Macro-averages of F1 scores [15]:

Micro - F1 = F1 over categories and documents

Macro - F1 = average of within - category F1 values

The MicroF1 gives equal weight to every document, while MacroF1 gives equal weight to every category, regardless of its frequency.

### 5.3 Experimental Design

We split the each dataset into three parts. Then we use two parts for training and the remaining third for test. We conduct the training-test procedure three times and use the average of the three performances as final result.

We employ TFIDF as input features. The formula for calculating the TFIDF can be written as follows:

$$W(t, \vec{d}) = f(t, \vec{d}) \times \log(N/n_t)$$

where  $N$  is the total number of training documents, and  $n_t$  is the number of documents containing the word  $t$ .

For experiments involving SVM we employed SVMtorch, which uses one-versus-the-rest decomposition and can directly deal with multi-class classification problems. (<http://www.idiap.ch/~bengio/projects/SVMtorch.html>).

For Naïve Bayes, we adopt multinomial model that always performs much better than multi-variate Bernoulli event model in text classification [20].

Despite of simplicity in theoretical analysis, many FS algorithms have proved to be more popular than FE algorithms for dimension reduction in the context of real-life text data. Among current various FS methods, Mutual Information (MI), Information Gain (IG), CHI Statistics (CHI), Document Frequency (DF) are the most popular ones partially due to their fast running speed and acceptable performance [8]. As a result, we use them as baselines in this work.

### 5.4 Comparison and Analysis

Table 6 reports the results when using 3000 features. From this table, we can observe that DB2 consistently yields much better performance than IG. More encouragingly, on Sector-48 DB2 beats IG by a wide margin: the MacroF1 of SVM using DB2 is approximately 7% higher than MacroF1 of SVM using IG; the

MicroF1 of SVM using DB2 is approximately 18% higher than MicroF1 of SVM using IG.

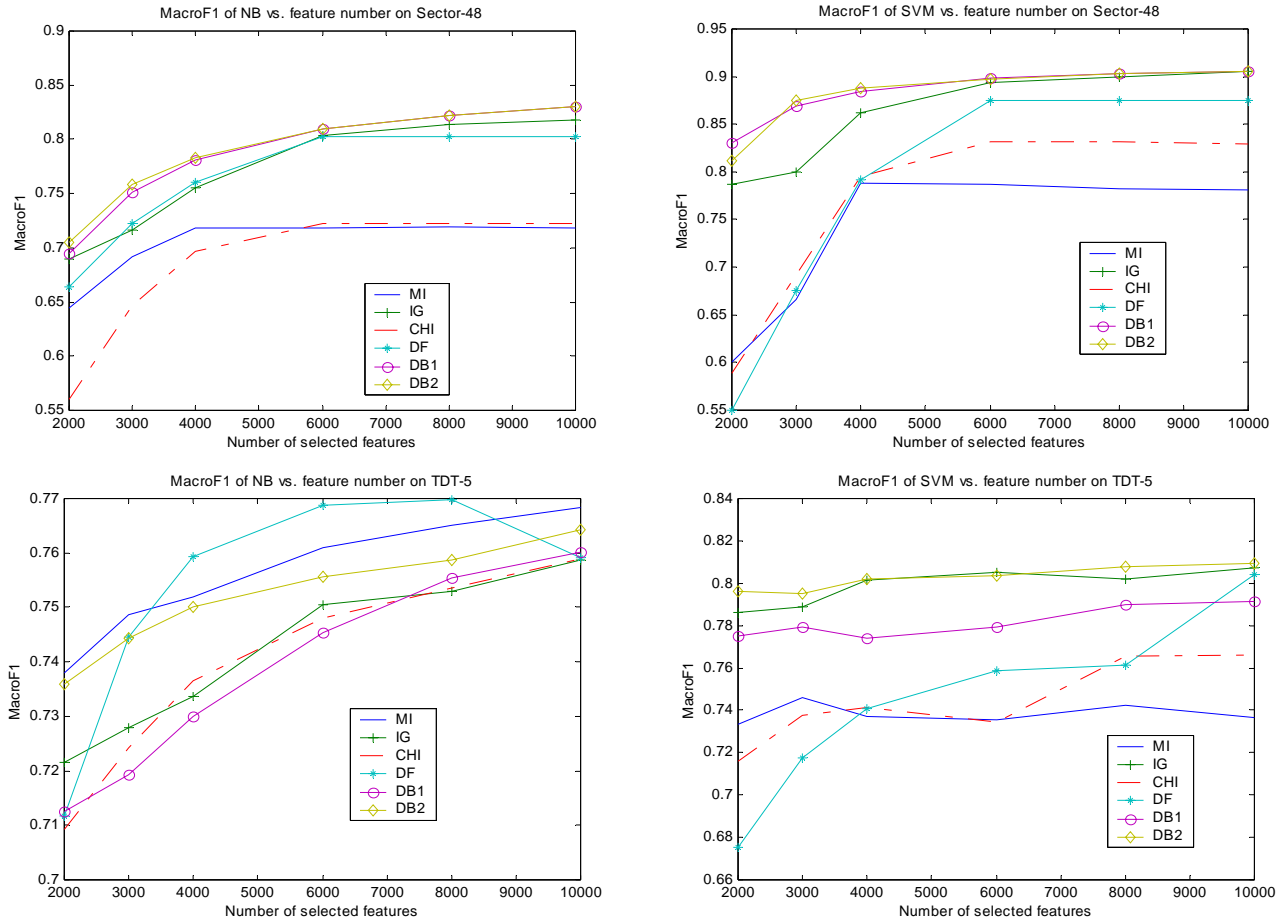
Under any conditions, DB2 performs better than DB1 when feature number is fixed as 3000. The average performance improvement is about one percent. This observation indicates that

in text community centroids can capture most discriminating information contained in original corpus.

An exceptional phenomenon is that on TDT-5 the MacroF1 of NB using MI is a little higher than the MacroF1 of NB using DB2.

**Table 6: The Performance of Different Methods Using 3000 Features**

	MI	IG	CHI	DF	DB1	DB2
MacroF1 of NB on Sector-48	0.6921	0.7159	0.6464	0.7221	0.7512	<b>0.7586</b>
MacroF1 of SVM on Sector-48	0.6657	0.7997	0.6915	0.6750	0.8692	<b>0.8745</b>
MacroF1 of NB on TDT-5	<b>0.7487</b>	0.7279	0.7243	0.7446	0.7194	0.7442
MacroF1 of SVM on TDT-5	0.7459	0.7890	0.7378	0.7178	0.7796	<b>0.7952</b>
MacroF1 of NB on Sector-48	0.6861	0.7062	0.6359	0.7121	0.7405	<b>0.7479</b>
MacroF1 of SVM on Sector-48	0.5276	0.6935	0.5649	0.5484	0.8664	<b>0.8714</b>
MacroF1 of NB on TDT-5	0.8925	0.9010	0.8819	0.8487	0.8962	<b>0.9026</b>
MacroF1 of SVM on TDT-5	0.8827	<b>0.9389</b>	0.8782	0.8432	0.9135	<b>0.9389</b>



**Figure 3: the MacroF1 curves of NB and SVM vs. feature number**

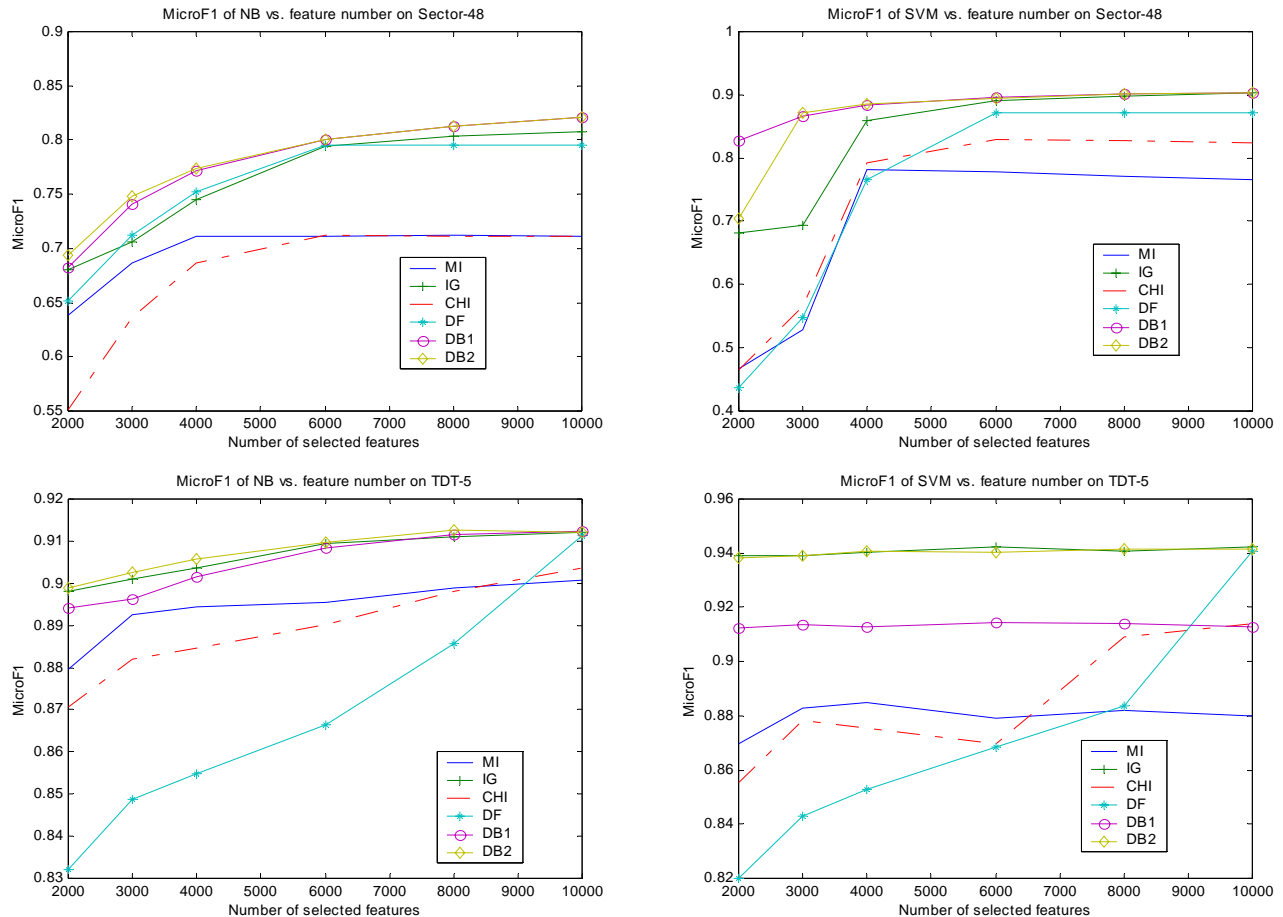


Figure 4: the MicroF1 curves of NB and SVM vs. feature number

Figure 3-4 display the performance curves of different classifiers after term selection using MI, IG, CHI, DF, DB1 and DB2 on two corpora respectively. From these figures, we can come to following conclusions:

First, DB2 consistently delivers top performance nearly as well as IG. Especially for SVM, IG and DB2 beat other methods by a wide margin.

Second, DB2 often produces better performance than IG when the number of selected features is smaller than 6000. This phenomenon indicates that DB2 could pick out more discriminative feature subset than IG.

Third, DB2 always performs better than DB1. Although examples-based discernibility matrix is more all-sided, it is more sensitive to imbalance of corpus than centroids-based discernibility matrix. In fact, the majority class consists of much larger amount of examples than minority class, and thus it should exert more influence on the score of discrimination of features. The employment of centroids can offset the influence of imbalance of corpus because it assigns a nearly equivalent weight for each class.

Table 7 reports the CPU time of different feature selection methods on two text data sets. The comparison is conducted on a personal computer with a single Intel Pentium 3.0G (MHZ) CPU

and 512M memory. Note that the running time does not include the seconds for loading data from hard disk. The number of features is set to 10,000.

From table 7, we can observe that DB2 runs much faster than DB1: on TDT-5, the running time of DB1 is about twenty times the running time of DB2; on Sector-48, the running time of DB1 is about seven times larger than that of DB2. Therefore replacement of examples with centroids in construction of discernibility matrix is able to dramatically cut down the running time.

MI, CHI, DF and DB2 require nearly the same amount of seconds that is much smaller than IG. In summary, these experiments have shown that DB2 offers alternative choice for text feature selection.

Table 7: The running time (in seconds)

DataSet	MI	IG	CHI	DF	DB1	DB2
Sector-48	6.453	43.266	5.354	5.229	46.062	6.479
TDT-5	3.312	13.984	2.901	2.188	65.406	3.661

## 6. CONCLUSION REMARKS

In this work we proposed a novel feature ranking measure based on rough set theory. Based on this measure, two feature

selection algorithms are developed. A comparison between proposed methods and four other feature selection methods, i.e., MI, IG, CHI and DF, is conducted on two English corpora with two classification methods, i.e., NB and SVM. Our main contributions are:

First, we make some extensions to discernibility matrix so that it can handle continuous features in text data. Based on this method, we design algorithm DB1. DB1 yields high performance approaching IG, whereas it runs a little slower than IG.

Second, in order to cut down the running time, we replace the examples with the centroids in constructing the discernibility matrix. Based on this strategy, we develop another algorithm DB2 that runs much faster as well as performs better than IG.

## 7. ACKNOWLEDGMENTS

This work was mainly supported by special fund of Chinese Academy of Sciences, "Research on Opinion Mining of Web Text", under grant number 0704021000 and two projects, i.e., 2007CB311100 and 2007AA01Z441.

## 8. REFERENCES

- [1] Bao, Y., Aoyama, S., Du, X., Yamada, K. and Ishii, N. A Rough Set-Based Hybrid Method to Text Categorization. WISE (1) 2001: 254-261
- [2] Bao, Y., Asai, D., Du, X., Yamada, K. and Ishii, N. An Effective Rough Set-Based Method for Text Classification. IDEAL 2003: 545-552
- [3] Chouchoulas, A., Shen, Q. A Rough Set-Based Approach to Text Classification. RSFDGrC 1999: 118-127
- [4] Chouchoulas, A. A Rough Set Approach to Text Classification. Thesis. 1999.
- [5] Dhillon, I., Mallela, S., and Kumar, R. A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. Journal of Machine Learning Research, 2003, 1265-1287.
- [6] Gilad-Bachrach, R., Navot, A. and Tishby, N. Margin based feature selection - theory and algorithms. ICML. Banff, Alberta, Canada. 2004.
- [7] Han, E., Karypis, G. Centroid-Based Document Classification Analysis & Experimental Result. PKDD. 2000.
- [8] Yang, Y. and Pedersen J. A Comparative Study on Feature Selection in Text Categorization. ICML. 1997, 412-420.
- [9] Hoa, N., Son, N. Some Efficient Algorithms For Rough Set Methods. 1996.
- [10] Hu, K., Diao, L., Lu, Y. and Shi, C. Sampling for Approximate Reduct in very Large Datasets. PKAW 2000.
- [11] Joachims, T. Text categorization with support vector machines: learning with many relevant features. ECML. 1998, 137-142.
- [12] Johnson, D. Approximation algorithms for combinatorial problems. Journal of Computer and System Sciences, vol. 9, pp. 256-278, 1974.
- [13] H. Liu and H. Motoda. Feature Extraction, Construction and Selection: A Data Mining Perspective. Kluwer Academic, Norwell, MA, USA, 1998.
- [14] Karypis, G., Han, E. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Technical report tr-00-0016, University of Minnesota, 2000.
- [15] Lewis, D., Schapire, R., Callan, J. and Papka, R. Training algorithms for linear text classifiers. SIGIR. 1996, 298-306.
- [16] Yang, Y., Lin, X. A re-examination of text categorization methods. SIGIR. 1999, 42-49.
- [17] Skowron, A., Rauszer, C. The discernibility matrices and functions in information system. In: Intelligent Decision Support: Handbook of Applications and Advances of The Rough Sets Theory. Dordrecht: Kluwer Academic Publishers, 1992, 331-362.
- [18] Salton, G., McGill, M. Introduction to Modern Retrieval. McGraw-Hill Book Company, 1983.
- [19] Rijsbergen, C. Information Retrieval. Butterworths, London, 1979.
- [20] McCallum, A., Nigam, K. A Comparison of Event Models for Naive Bayes Text Classification. AAAI/ICML-98 Workshop on Learning for Text Categorization[C]. Menlo Park, CA: AAAI Press. 1998, 41-48.
- [21] Mun, P. Text Classification in Information Retrieval using Winnov. <http://citeseer.ist.psu.edu/cs>
- [22] Pal, S., Skowron, A. Rough Fuzzy Hybridization-A new trend in decision-making, Springer, 1999.
- [23] Pawlak, Z. Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht, 1991