Journal of Southeast University (English Edition) Vol. 23, No. 1, pp. 63 – 69 Mar. 2007 ISSN 1003—7985

New text classification algorithm based on interdependence and equivalent radius

Wang Hongwei[1] Yi Lei[2] Wang Jianhui[3]

([1]School of Economics and Management, Tongji University, Shanghai 200092, China)

([2]Department of Mathematics, Fudan University, Shanghai 200433, China)

([3]School of Information Science and Engineering, Fudan University, Shanghai 200433, China)

Abstract: To improve the traditional classifying methods, such as vector space model (VSM)-based methods with highly complicated computation and poor scalability, a new classifying method (called IER) is presented based on two new concepts: interdependence and equivalent radius. In IER, the attribute is selected according to the value of interdependence, and the classifying rule is based on equivalent radius and center of gravity. The algorithm analysis shows that IER is good at classifying a large number of samples with higher scalability and lower computation complexity. After several experiments in classifying Chinese texts, the conclusion is drawn that IER outperforms k-nearest neighbor (kNN) and classifcation based on the center of classes (CCC) methods, so IER can be used online to automatically classify a large number of samples while keeping higher precision and recall.

Key words: classification; equivalent radius; vector space; interdependence; interdependence and equivalent radius

With the increasing interest in automatic classification, some algorithms are presented, such as Bayes[1], SVM[1-2], Boosting[3], kNN[4]. Most of them are based on VSM, which has been applied to text classification successfully to some degree[5]. However, these algorithms are highly complicated in computation, and can hardly be used when classifying a large number of samples. Moreover, as far as these algorithms are concerned, the classifier must be rebuilt when increasing the corpora of the training samples. As a result, they possess tough scalability. It is essential to find a new method to classify a large scale of samples with higher precision and recall[6].

This paper presents a simple and efficient algorithm to classify a large scale of texts based on interdependence and equivalent radius (IER for short). IER is lower in computation complexity. It enables a quick response to classification and a good scalability. Distinguished from VSM-based algorithms, IER uses the equivalent radius to construct a judging function, which returns a relative value according to the distribution of training samples, rather than an absolute value based on the number of training samples. As a result, the misjudgment caused by the great disparity among categories in the number of samples or in the distribution scope can be avoided. Thus, IER makes it possible to classify a large number of online texts, while keeping a high ratio of precision and recall.

Received 2006-04-28.

Foundation items: The National Natural Science Foundation of China (No. 70501024, 70501022), the Humanity & Social Science Research Program of Ministry of Education of China (No. 05JC870013).

Biography: Wang Hongwei (1973—), male, doctor, lecturer, hwwang@mail.tongji.edu.cn.

1 IER Algorithm

1.1 Attribute selection based on interdependence

In information processing, information samples are always pre-processed in advance to better the expression of samples, where a key step is selecting attributes. Generally, algorithms for attribute selection fall into two categories: statistics-based algorithms and data dictionary-based algorithms[7-8]. They both have their own advantages and disadvantages. For example, the former is domain-independent without requiring a data dictionary. However, it is lower in precision. However, the latter needs the support of a data dictionary, offering more precision. As we know, it is impossible to express the whole real world at the semantic level just by a single data dictionary, so data dictionary-based algorithms must be domain-specific. Moreover, it is really time-wasting to establish a data dictionary. So, statistics-based algorithms are widely used for attribute selection in practice, such as *N*-gram. But the *N*-gram algorithm is not satisfactory yet, often giving some nonsensical word segmentations. Even if Ref. [6] proposed two rules for getting word segmentations in length as long as possible, such a problem has not been entirely solved yet.

Considering the problem above, this paper compares characteristics of mutual information and a dependent value model, respectively, and constructs the interdependence model combining two-part advantages and discarding disadvantages. The interdependence model can determine to what degree two morphemes are dependent on each other, so it can be used for selecting key attributes in text classification. The interdependence-based algorithm of attribute selection would be helpful not only in selecting the right attributes, but also in decreasing the dimension. It can also ignore those attributes with little contribution to text classification.

In Priori, any non-empty subset of a frequent item set must be frequent. Based on that feature, this paper proposes a segmentation rule (see rule 1), by which the words in maximal length can be obtained for decreasing the dimension. Algorithm 1 uses an interdependence model to process words, which is similar to rule 1.

Definition 1 (interdependence) Interdependence between two variables χ and η is defined as

$$I(\chi,\eta)=[F(s)\times L-F(\chi)\times F(\eta)]\times \log\left(\frac{F(S)\times L}{F(\chi)\times F(\eta)}\right)^{\frac{1}{L^2}} \quad (1)$$

where $F(\chi)$ and $F(\eta)$ are the frequency of χ and η occurring, respectively; $F(s)$ is the frequency of χ and η occurring together; L is the length of samples. All these frequencies are directly obtained from samples to be processed. Thus, the strongpoint is that the data dictionary is not required, and the words segmented from samples are closer to domain knowledge.

Rule 1 If there exist two N-gram items t_i and t_j, satisfying $t_i \supset t_j$ and $s(t_i)=s(t_j)$, then either t_i or t_j is redundant. So remain t_i by default. Where $s(*)$ is the score of an N-gram item, which is equal to the occurring frequency of the N-gram item. Rule 1 is for eliminating redundant words.

Theorem 1 The interdependence between χ and η ranges within $\left[0,\frac{1}{4}\log L\right]$, where L is the length of samples.

Proof of theorem 1 refers to Ref. [9]. According to theorem 1, we set an upper limit μ_U and lower limit $\mu_L\left(0<\mu_L<\mu_U<\frac{1}{4}\log L\right)$ for selecting attributes with interdependence ranging within $[\mu_L, \mu_U]$. The basic thought for attribute selection is described as follows:

1) Use the N-gram algorithm to segment words, discard the words with higher or lower frequency, then get a word set $U(x)=\{x \mid x\in U_j, 1\leqslant j\leqslant n\}$, where U_j are the words of the original text with a value of j in length.

2) Judge the correlation between the words in $U(x)$ according to interdependence: First, process n-gram words, then $(n-1)$-gram, …, until to 1-gram. When processing an i-gram word $x_i\in U(x)(2\leqslant i\leqslant n)$, segment x_i into x_i^j equal to j in length($1\leqslant j<i$) and x_i^k equal to $k(k=i-j)$ in length. If $x_i^j\in U(x)$ and $x_i^k\in U(x)$, then compute $I(x_i^j, x_i^k)$. When selecting attributes, use μ_U and μ_L to process by the following steps:

- If $I(x_i^j, x_i^k)>\mu_U$, then subtract the frequency of x_i from the frequency of x_i^j and the frequency of x_i^k respectively;
- If $I(x_i^j, x_i^k)<\mu_L$, then delete x_i from $U(x)$;
- If $I(x_i^j, x_i^k)$ ranges within $[\mu_L, \mu_U]$, then $U(x)$ does not change;

3) Finally, delete the words with higher or lower frequency from $U(x)$ respectively.

When processing the i-gram words, the j-gram words $(0<j<i)$ have been processed, so the computation complexity decreases, and the remaining words are more precise. Most Chinese words consist of two characters[10], so n in N-gram is often set to be 4.

Algorithm 1 Attribute selection algorithm based on interdependence

Input: D represents the texts with attributes to be selected; μ_U is the upper limit of interdependence; μ_L is the lower limit of interdependence; δ_U is the upper limit frequency of words; δ_L is the lower limit frequency of words; n is the value of N-gram.

Output: U_f is attribute set; W_f is the adjusted frequency set corresponding to U_f.

Steps:

① Segment words using the N-gram algorithm, then delete the words with higher frequency ($>\delta_U$) or with lower frequency ($<\delta_L$), finally get a word set U_f and a word frequency set W_f.

② For $I=n$ to 2 Step -1

Do

Pick an unprocessed I-gram word from U_f, and try to segment it into word x_i and word x_j existing in U_f. According to the frequency in W_f and Eq. (1), compute $I(x_i, x_j)$.

If $I(x_i, x_j)>\mu_U$ then

Subtract the frequency of I-gram word from the frequency of x_i and x_j of W_f respectively;

Else if $I(x_i, x_j)<\mu_L$ then

Delete the current I-gram word from

U_f, and delete the I-gram word's frequency from W_f.

End if

Until all of I-gram words in U_f are processed.

Next I;

③ Delete those words whose frequency is larger than δ_U or less than δ_L from U_f, and delete their frequency from W_f.

④ Use rule 1 to reduce the dimension, and yield an attribute set U_f and an frequency attribute W_f.

1.2 Classification model

Definition 2 (equivalent radius, ER) Let the number of training samples for category $\omega_i (i = 1, 2, \ldots, c)$ be n_i, and the dimension of vector space d. Project all the samples of ω_i on every dimension $d_j (j = 1, 2, \ldots, d)$ respectively, then get the projection range of ω_i on d_j, denoted as $R_{ij} = [R_{ij}^-, R_{ij}^+]$. Compute the center of gravity of ω_i on d_j, denoted as C_{ij}. Here, R_{ij}^- is the radius from C_{ij} to the origin O, and R_{ij}^+ is the radius backward O from C_{ij}. Generally, $R_{ij}^- \neq R_{ij}^+$. n_{ij}^- is the number of samples whose projection is located between O and C_{ij}. n_{ij}^+ is the number of samples which project within $[C_{ij}, R_{ij}^+]$. R_{ij} is an equivalent radius function of ω_i projected on d_j, expressed as

$$R_{ij} = a_{ij} R_{ij}^- + (1 - a_{ij}) R_{ij}^+ \tag{2}$$

where $1 \leqslant i \leqslant c$, $1 \leqslant j \leqslant d$, a_{ij} is the distribution coefficient.

Algorithm 2 Classification model

Input: d_{ti} is the training sample set, where $1 \leqslant t \leqslant n_i$, $1 \leqslant i \leqslant c$, n_i is the number of training samples for category $\omega_i (i = 1, 2, \ldots, c)$, and c is the number of categories.

Output: R_{ij} is the equivalent radius of the category i on the dimension j; C_{ij} is the center of gravity of the category i on the dimension j; a_{ij} is the distribution coefficient of the category i on the dimension j, $1 \leqslant i \leqslant c$, $1 \leqslant j \leqslant d$.

Phase 1 Compute $a_{ij}, R_{ij}^-, R_{ij}^+$

① Assign sequence numbers to categories and attributes respectively, where the category number ranges $[1, c]$, and the attribute dimension ranges $[1, d]$.

② Let $i = 1$.

③ For each category ω_i, do the following operations:

Normalize all the training sample vectors of the category ω_i.

Project all the training samples of ω_i on the dimension $d_j (j = 1, 2, \ldots, d)$, and get the projection range $R_{ij} = [R_{ij}^-, R_{ij}^+]$.

Compute $C_{ij}, n_{ij}^-, n_{ij}^+, R_{ij}^-, R_{ij}^+$.

④ If all the categories have been processed, then enter phase 2. Else, let $i = i + 1$, and go to step ③ to process the next category.

Phase 2 Determine the equivalent radius

① Let $i = 1$. For each category ω_i, do the following operations.

② Let $j = 1$.

③ For each dimension d_j, compute $a_{ij} = n_{ij}^- / (n_{ij}^- + n_{ij}^+)$, $R_{ij} = a_{ij} R_{ij}^- + (1 - a_{ij}) R_{ij}^+$.

④ If all the dimensions have been processed, then enter step ⑤; else, let $j = j + 1$, and go to step ③ to process the next dimension.

⑤ If all the categories have been processed, then stop. Else, let $i = i + 1$, and go to phase 2 to process the next category.

a_{ij} can be determined by Golden section or other methods. Algorithm 2 sets $a_{ij} = n_{ij}^- / (n_{ij}^- + n_{ij}^+)$ according to sample distribution. The reason is that, if the distribution of samples is denser in the area toward the origin O from C_{ij}, then $R_{ij}^- < R_{ij}^+$, $n_{ij}^+ < n_{ij}^-$, so R_{ij}^- should have a greater weight, and let $a_{ij} = n_{ij}^- / (n_{ij}^- + n_{ij}^+)$, $R_{ij} = a_{ij} R_{ij}^- + (1 - a_{ij}) R_{ij}^+$, and vice versa.

1.3 IER algorithm

First, construct a judging function depending on category ω_i. Where $(x_j - C_{ij})^2 / R_{ij}^2$ is defined as the relative distance between a training sample $x = \{x_1, x_2, \ldots, x_d\}$ and ω_i. Thus, the judging function is based on relative distance rather than absolute distance so as to avoid the misjudgment caused by a great disparity of different categories in the number of training samples or in the distribution area. According to Eq. (2), if the projection of x on d_j is equal to 0, then $R_{ij} = 0$, from which the division overflow arises. Therefore, a distance coefficient denoted as $1/\beta$ is introduced. Thus, the judging function is

$$g_i(x) = \sqrt{\sum_{j=1}^{k} \left(\frac{x_j - C_{ij}}{R_{ij}} \right)^2 + \sum_{j=k+1}^{d} \frac{x_j^2}{\beta^2}} \qquad i = 1, 2, \ldots, c \tag{3}$$

Among all the categories, if ω_i is the closest to the testing sample x in relative distance, i.e. $g_i(x)$ is the smallest, then we say x belongs to ω_i. So the rule is: if $g_j(x) = \min_i \{g_i(x)\}$, $i = 1, 2, \ldots, c$, then $x \in \omega_j$. Next, it will be shown that the classification result is not highly sensitive to $1/\beta^2$, because if a projection of one category on a dimension is equal to 0, then the dimension is generally a specific one for other categories. And a small change in $1/\beta^2$ cannot greatly influence the final results.

Algorithm 3 IER algorithm

Input: d_{ti} is the training sample set, where $1 \leqslant t \leqslant$

n_i, $1 \leqslant i \leqslant c$, n_i is the number of texts to be trained for category ω_i ($i = 1, 2, \ldots, c$), and c is the number of categories. t_j is testing sample set, where $1 \leqslant j \leqslant m$, and m is the number of texts to be tested.

Output: The category of the testing samples.

Steps:

① According to algorithm 1, extract attributes from the training sample set.

② According to algorithm 2, compute the center of gravity and the equivalent radius of classifier projections on every dimension.

③ From the testing sample set, pick a sample x, not yet classified.

④ Get the character value of x.

⑤ Compute $g_i(x)$ according to Eq. (3).

⑥ If $g_j(x) = \min_i \{g_i(x)\}$, $i = 1, 2, \ldots, c$, then $x \in \omega_j$.

⑦ If all the testing samples have been classified, then stop. Else, go to step③.

Notations are explained as follows: c is the number of categories, d is the number of dimension, n is the average number of testing texts for every category, and m is the average number of training texts for every category. Then, IER's performances are analyzed:

1) Classification efficiency From algorithm 3, we know that the worst-case complexity of IER is $\odot(cn(\log_2 c)!)$, while it is $\odot(c^2 nm + cn(\log_2(cm))!)$ for kNN. Generally, $n \ll m$, so the complexity of IER is less than that of kNN.

2) Updating performance IER is better at updating, because when adding or deleting training samples, just adjust the trained classifier into a new classifier according to the attributes of training samples to be added or deleted, rather than train all the samples again. Then, we introduce how to adjust the classifier in the case of adding a training sample. And it is done in the similar way when deleting. When adding $x = (x_1, x_2, \ldots, x_d)$, it is only required to adjust C_{ij}, the center of gravity for the category ω_i ($i = 1, 2, \ldots, c$), and R_{ij}.

The relationship between the adjusted center of gravity, written as $C_{ij}^{(1)}$, and the original center of gravity, written as $C_{ij}^{(0)}$, is described as[9]

$$C_{ij}^{(1)} = \frac{nC_{ij}^{(0)} + x_j}{n+1} \tag{4}$$

The relationship between the adjusted ER, written as $R_{ij}^{(1)}$, and the original ER, written as $R_{ij}^{(0)}$, is described as[9]

$$R_{ij}^{(1)} = R_{ij}^{(0)} + \frac{(n_{ij}^{+} - n_{ij}^{-})(C_{ij}^{(0)} - x_j)}{(n+1)^2} + \frac{|C_{ij}^{(1)} - x_j|}{n+1} \tag{5}$$

After the adjustment of C_{ij} and R_{ij}, n_{ij}^{+} and n_{ij}^{-} need to be revised according to algorithm 4. When more samples are added, repeat Eq. (5) and finally get a new classifier.

1.4 Updating performance

Algorithm 4 Updating IER classifier

Input: $C_{ij}^{(0)}$, $R_{ij}^{(0)}$, n_{ij}^{+}, n_{ij}^{-} (for existed classifier); U is the set of training samples to be added.

Output: $C_{ij}^{(1)}$, $R_{ij}^{(1)}$, n_{ij}^{+}, n_{ij}^{-} (for new classifier).

Steps:

① Pick a sample $x = (x_1, x_2, \ldots, x_d)$ from U.

② According to Eqs. (4) and (5), compute $C_{ij}^{(1)}$ and $R_{ij}^{(1)}$ on the category of x for the new classifier.

③ If $C_{ij}^{(1)} > x_j$ Then

{ $\Omega = [(C_{ij}^{(0)} - C_{ij}^{(1)}) \times n_{ij}^{-} / R_{ij}^{(0)}]$;

$n_{ij}^{-} = n_{ij}^{-} - \Omega + 1$;

$n_{ij}^{+} = n_{ij}^{+} + \Omega$; }

Else

{ $\Omega = [(C_{ij}^{(1)} - C_{ij}^{(0)}) \times n_{ij}^{+} / R_{ij}^{(0)}]$;

$n_{ij}^{+} = n_{ij}^{+} - \Omega + 1$;

$n_{ij}^{-} = n_{ij}^{-} + \Omega$; }

④ If all the training samples to be added have been processed in the above way, then the algorithm stops. Else, $C_{ij}^{(0)} = C_{ij}^{(1)}$, $R_{ij}^{(0)} = R_{ij}^{(1)}$, and go to step ①.

2 Experimental Results and Analysis

2.1 Experiment contents and performance

The experimental data, stored in two corpora G1 and G2, comes from the 2005-year's People Daily and the Sohu website, respectively, shown as Tab. 1 and Tab. 2.

Tab. 1 The testing corpora (G1)

Categories	Number of texts
Politics	1 243
Sports	421
Economy	593
Agriculture	186
Environment	404
Astronautics	575
Art	726
Education	321
Medicine	149
Transportation	352
Energy	142
Computer	251
Mining	368
History	702
Military	522

Tab. 2 The testing corpora (G2)

Categories	Number of texts
Mining	1 049
Military	921
Computer	746
Electronics	501
Communications	249
Energy	429
Philosophy	332
History	393
Law	344
Literature	199

Definition 3 (precision and recall) In classification, a is the number of texts belonging to the category ω_i and has been identified as ω_i successfully. b is the number of texts not belonging to ω_i, but is identified as ω_i mistakenly. c is the number of texts belonging to ω_i, but fails to be identified as ω_i. d is the number of the texts not belonging to ω_i, and has not been identified as ω_i. So the recall ratio is $a/(a+c)$, and the precision ratio is $a/(a+b)$.

Definition 4 (F-value) F-value is defined as a function depending on the precision ratio and the recall ratio to measure the classification performance. F-value is written as

$$F = \frac{2PR}{P+R} \tag{6}$$

The experimental process is as follows:

① Let the distance coefficient be 1, 6.5, 12.5, 25, 50, 100, 200, respectively. Taking different dimensions of the vector space, we use IER to test texts in G1 and G2 for the purpose of evaluating the influence of distance coefficients on classification results.

② Pick randomly 70%, 80%, 90% of the samples from G1 and G2 respectively for training, and the remaining 30%, 20%, 10% are used for testing. Taking different dimensions of the vector space, we compare IER with kNN and CCC (classification based on the center of classes) in classification performance for open tests.

③ Pick randomly 10%, 25%, 50% of the samples for every category in G1 and G2 to test. And 100% takes part in training. Taking different dimensions of the vector space, we compare IER with kNN and CCC in the classification performance of the closed test.

④ Use algorithm 4 to add 100 texts belonging to the 15 categories to G1. Then, use IER to test for the purpose of evaluating the learning ability in increments.

2.2 Experimental results

Figs. 1 and 2 show the F-value vs. the distance coefficient and the dimension. Not changing the dimension, F-value is improved greatly at the beginning of the increase of the distance coefficient. After reaching the maximum, F-value decreases slowly and trends to an invariableness (see Fig. 2). As mentioned above, the improvement of performance is not obvious after F-value's reaching the peak, so the classification result is not sensitive to the distance coefficient anymore. The reason is that, for a category, if the projection on a character is equal to 0, then generally the character is distinct from others. The distance coefficient emphasizes the distance between the training sample and the category. The emphasizing degree cannot affect the testing result. So a little change in the distance coefficient within a wider scope cannot greatly influence the classification performance. When the distance coefficient is larger than 40, the performance in different dimensions decreases smoothly. So in the experiment, $1/\beta^2$ is set to be 40. Fig. 1 shows the influence of dimension on classification performance. Too few dimensions would not be enough to express the characteristics of the dimension, while too many dimensions would cause some disturbance. So in the experiments of this work, when the dimension is equal to 300, the best performance is obtained.

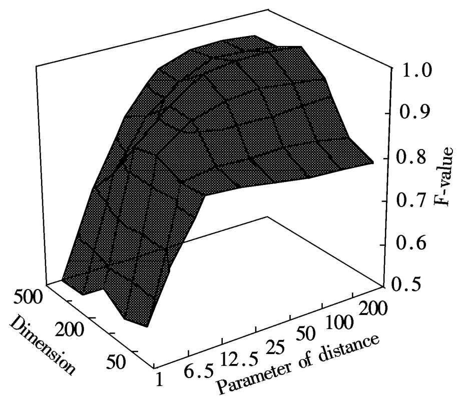

Fig. 1 F-value vs. distance coefficient and dimension

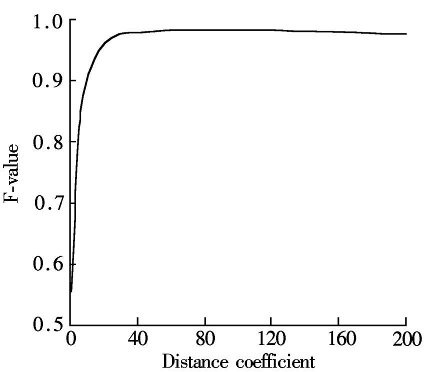

Fig. 2 F-value vs. distance coefficient when dimension is 300

When classifying a large scale of texts, the speed should be considered, as well as recall and precision. We take the average of F-value and the response time as factors in evaluating IER comprehensively. The larger F-value and the shorter the time, the higher the whole performance is. Figs. 3 to 6 illustrate the relationships of F-value and response time vs. dimensions for closed and open tests, respectively. The experiments show that IER is better than kNN and CCC in recall and precision. As for the response time, IER is better than kNN, and is equivalent to CCC. However, the F-value in CCC is far less then that in IER. So it can be concluded that IER outperforms kNN and CCC in the whole performance.

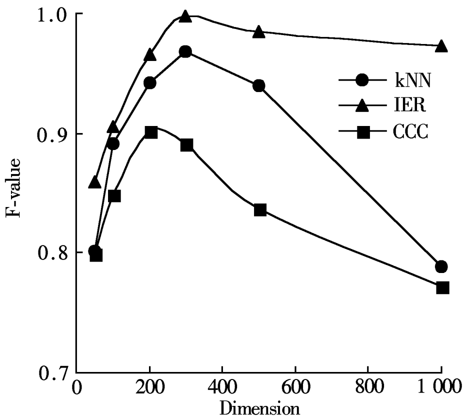

Fig. 3 F-value vs. dimension for closed test

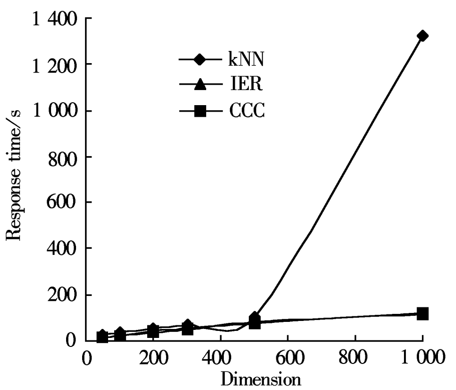

Fig. 4 Response time vs. dimension for closed test

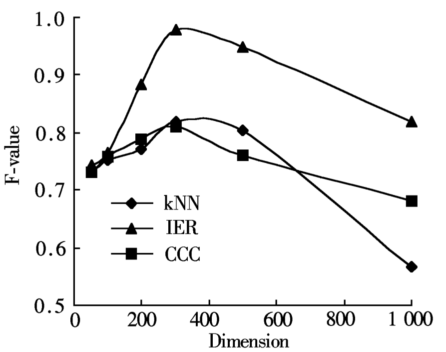

Fig. 5 F-value vs. dimension for open test

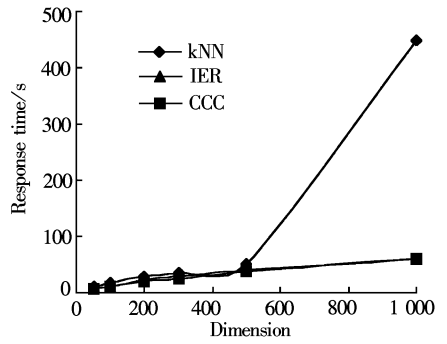

Fig. 6 Response time vs. dimension for open test

Moreover, after adding 100 texts to G1 by algorithm 4, both precision and recall increase 1%, so the performance of the classifier can still be guaranteed when updating corpora. Therefore, IER not only offers higher classification precision and speed, but also supports incremental learning.

3 Conclusion

In pattern recognition, most current classification algorithms are based on the vector space. Among them, kNN is a widely-used one. However, these algorithms do not adapt well to a large scale of texts because of the high complexity of computation. Moreover, when increasing the corpus of training samples in size, the classifier should be rebuilt. So they are poor in scalability. This paper presents two concepts, interdependence and equivalence radius, and proposes an algorithm based on the two concepts, which suits classifying a large text, and has good scalability. Compared with kNN and CCC, not only the precision and recall are increased, but the response speed is also improved.

Of course, there is still further work for IER. For example, it is by training and testing that the value of distance coefficient in the above experiments is determined. Though, the distance coefficient illustrates the same changing tendency as performance in G1 and G2, and experimental results are also satisfied, some questions still should be considered for a fully new application. It is uncertain whether the distance coefficient is an invariable just as it is in this paper. If not, how do we determine it efficiently? So the future work would focus on determination of a coefficient in a more precise way to optimize IER.

References

[1] Bian Zhaoqi, Zhang Xuegong. *Pattern recognition* [M]. Beijing: Tsinghua University Press, 2001: 9 − 43. (in Chinese)

[2] Burges C J C. A tutorial on support vector machines for

pattern recognition [J]. *Data Mining and Knowledge Discovery,* 1998, **2**(2): 955 − 974.

[3] Schapire R, Singer Y. BoosTexter: a boosting-based system for text categorization [J]. *Machine Learning*, 2000, **39**(2/3): 135 − 168.

[4] Dasarathy Y. Minimal consistent set (MCS) identification for optimal nearest neighbor decision system terms design [J]. *IEEE Trans Syst Man Cybern*, 1996, **24**(3): 511 − 517.

[5] Lam W, Ho C Y. Using a generalized instance set for automatic text categorization [C]//*Proc of the* 21*th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval*. Melbourne, Australia, 1998: 81 − 89.

[6] Zhou Shuigeng. The research on Chinese text database and Chinese text processing [D]. Shanghai: Fudan University, 2001. (in Chinese)

[7] Peng Fuchun, Schuurmans Dale. Self-supervised Chinese word segmentation [C]//*Proc of the* 4*th Int Symposium on Intelligent Data Analysis*. Cascais, Portugal, 2001: 238 − 247.

[8] Sproat R, Shih C L A stochastic finite-state word segmentation algorithm for Chinese [J]. *Computational Linguistics*, 1996, **22**(3): 377 − 404.

[9] Wang Jianhui, Hu Yunfa. An algorithm to classify documents based on equivalent radius. Technical report, No. 021011346 [R]. Shanghai: Fudan University, 2002. (in Chinese)

[10] Emerson Thomas. Segmenting Chinese in Unicode [C]//*Proc of the* 16*th Int Unicode Conference*. Amsterdam, Holland, 2000: 1 − 10.

基于互依赖和等效半径的文本分类方法

王洪伟[1]　伊　磊[2]　王建会[3]

([1] 同济大学经济与管理学院，上海 200092)
([2] 复旦大学数学科学学院，上海 200433)
([3] 复旦大学信息科学与工程学院，上海 200433)

摘要: 为了解决传统分类方法计算复杂度高及可扩展性差的问题，提出了互依赖和等效半径的概念，并将两者相结合，提出新的分类算法——基于互依赖和等效半径、易更新的分类算法 IER. IER 算法根据互依赖作为特征选择的量度，通过较长特征值的选择降低维度，通过重心和等效半径来建立分类模型. 算法分析显示 IER 计算复杂度较低，扩展性能较好，适用于大规模场合. 将 IER 算法应用于中文文本分类，并与 kNN 算法和类中心向量法进行比较，结果表明，在提高分类精度的同时，IER 还可以大幅度提高分类速度，有利于对大规模信息样本进行实时在线的自动分类.

关键词: 分类；等效半径；向量空间；互依赖；IER

中图分类号: TP139