Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 33 (2007) 1–5

Expert Systems with Applications

www.elsevier.com/locate/eswa

A novel feature selection algorithm for text categorization

Wenqian Shang [a,*], Houkuan Huang [a], Haibin Zhu [b], Yongmin Lin [a], Youli Qu [a], Zhihai Wang [a]

[a] *School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, PR China*

[b] *Department of Computer Science, Nipissing University, North Bay, Ont., Canada P1B 8L7*

Abstract

With the development of the web, large numbers of documents are available on the Internet. Digital libraries, news sources and inner data of companies surge more and more. Automatic text categorization becomes more and more important for dealing with massive data. However the major problem of text categorization is the high dimensionality of the feature space. At present there are many methods to deal with text feature selection. To improve the performance of text categorization, we present another method of dealing with text feature selection. Our study is based on Gini index theory and we design a novel Gini index algorithm to reduce the high dimensionality of the feature space. A new measure function of Gini index is constructed and made to fit text categorization. The results of experiments show that our improvements of Gini index behave better than other methods of feature selection.

Keywords: Text feature selection; Text categorization; Gini index; kNN classifier; Text preprocessing

1. Introduction

With the advance of WWW (world wide web), text categorization becomes a key technology to deal with and organize large numbers of documents. More and more methods based on statistical theory and machine learning has been applied to text categorization in recent years. For example, k-nearest neighbor (kNN) (Cover & Hart, 1967; Yang, 1997; Yang & Lin, 1999; Tan, 2005), Naive Bayes (Lewis, 1998), decision tree (Lewis & Ringuette, 1994), support vector machines (SVM) (Joachims, 1998), linear least squares fit, neural network, SWAP-1, and Rocchio are all such kinds of methods.

A major problem of text categorization is the high dimensionality of the feature space. For many learning algorithms, such high dimensionality is not permitted. Moreover most of these dimensions are not relative to text categorization; even some noise data hurt the precision of the classifier. Hence, we need to select some representative features from the original feature space (i.e., feature selection) to reduce the dimensionality of feature space and improve the efficiency and precision of classifier. At present the feature selection method is based on statistical theory and machine learning. Some well-known methods are information gain, expected cross entropy, the weight of evidence of text, odds ratio, term frequency, mutual information, CHI (Yang & Pedersen, 1997; Mladenic & Grobelnik, 2003; Mladenic & Grobelnik, 1999) and so on.

In this paper, we do not discuss these methods in detail. We present another new text feature selection method—Gini index. Gini index was early used in decision tree for splitting attributes and got better categorization precision. However, it is rarely used for feature selection in text categorization. Shankar and Karypis discuss how to use Gini index for text feature selection and weight-adjustment. They mainly pay attention on weight-adjustment. Their method only limits to centroid based classifier and their iterative method is time-consuming. Our method is very different from theirs. Through deeply analyzing the principles of Gini index and text feature, we construct a new

[*] Corresponding author.
E-mail addresses: shangwenqian@hotmail.com (W. Shang), haibinz@npissingu.ca (H. Zhu).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.04.001

measure function of Gini index and use it to select features in the original feature space. It not only fit centroid classifiers but also fit other classifiers. The experiments show that its quality is comparable with other text feature selection methods. However, its complexity of computing is lower and its speed is higher.

The rest of this paper is organized as follows. Section 2 describes the classical Gini index algorithm. Section 3 gives the improved Gini index algorithm. Section 4 discusses the classifiers using in the experiments to compare Gini index with the other text feature selection methods. Section 5 presents the experiments' results and their analysis. In the last section, we give the conclusion.

2. Classical Gini index algorithm

Gini index is a non-purity split method. It fits sorting, binary systems, continuous numerical values, etc. It was put forward by Breiman, Friedman, and Olshen (1984) and was widely used in decision tree algorithms of CART, SLIQ, SPRINT and intelligent miner. The main idea of Gini index algorithm is as follows:

Suppose S is the set of s samples. These samples have m different classes $(C_i, i = 1, \ldots, m)$. According to the differences of classes, we can divide S into m subset $(S_i, i = 1, \ldots, m)$. Suppose S_i is the sample set which belongs to class C_i, s_i is the sample number of set S_i, then the Gini index of set S is:

$$\mathrm{Gini}(S) = 1 - \sum_{i=1}^{m} P_i^2, \tag{1}$$

where P_i is the probability that any sample belongs to C_i and estimating with s_i/s. Gini(S)'s minimum is 0, that is, all the members in the set belong to the same class; this denotes it can get the maximum useful information. When all the samples in the set distribute equably for the class field, Gini(S) is maximum; this denotes it can get the minimum useful information. If the set is divided into n subset, then the Gini after splitting is:

$$\mathrm{Gini}_{\mathrm{split}}(S) = \sum_{j=1}^{n} \frac{s_j}{s} \mathrm{Gini}(S_j). \tag{2}$$

The minimum $\mathrm{Gini}_{\mathrm{split}}$ is selected for splitting attribute.

The main idea of Gini index is: for every attribute, after it traverses all possible segmentation methods, if it can provide the minimum Gini index then it is selected as the divisive criterion of this node no matter it is the root node or a sub node.

3. The improved Gini index algorithm

To apply the Gini index theory described above directly to the text feature selection, we can construct the new formula:

$$\mathrm{Gini}(W) = P(W)\left(1 - \sum_i P(C_i|W)^2\right) + P(\overline{W})\left(1 - \sum_i P(C_i|\overline{W})^2\right) \tag{3}$$

After we analyze and compare the merits and demerits of the existing text feature selection measure functions, we improve formula (3) to:

$$\mathrm{Gini\,Text}(W) = \sum P(W|C_i)^2 P(C_i|W)^2. \tag{4}$$

Why we amend formula (3) to formula (4)? The reasons include three aspects as follows:

(1) The original form of Gini index is used to measure the impurity of attributes towards categorization. Smaller the impurity is, better the attribute is. If we adopt the form $\mathrm{Gini}(S) = \sum_{i=1}^{m} P_i^2$, it is to measure the purity of attributes towards categorization. Bigger the value of purity is, better the attribute is. In this paper, we adopt the measure form of purity. This form is more adapt to text feature selection. In paper (Gupta, Somayajulu, Arora, & Vasudha, 1998; Shankar & Karypis), they all adopt the measure form of purity.

(2) In other authors' papers, they all emphasize that text feature selection inclines to high frequency words, namely, including the $P(W)$ factor in the formula. Experiments show that some words that do not appear have contributions to judge the class of text, but this contribution is far less significant than the effort to consider the words that do not appear, especially when the distribution of the class and feature values is highly unbalanced. Yang and Pedersen (1997) and Mladenic and Grobelnik (1999) compare and analyze synthetically the merits and demerits of many feature measure functions in their papers. Their experiments show that the demerits of information gain are to consider the word that does not appear. The demerits of mutual information are not to consider the affect of the $P(W)$ factor leading to select rare words. Expected cross entropy and weight of evidence of text overcome these demerits, hence their results are better. Therefore, when we construct the new measure function of Gini index, we get ride of the affection factor expressing words that do not appear.

(3) Iff W_1 appears in the documents of class C_1 and W_1 appears in every document of class C_1; Iff W_2 appears in the documents of class C_2 and W_2 appears in every documents of class C_2, then W_1 and W_2 is the same important feature. But due to $P(C_i) \neq P(C_j)$, from $\mathrm{Gini\,Text}(W) = P(W)\sum_i P(C_i|W)^2$ to compute out $\mathrm{Gini\,Text}(W_1) \neq \mathrm{Gini\,Text}(W_2)$, this is not consistent with domain knowledge. So we adopt $P(W|C_i)^2$ to replace $P(W)$, for considering the unbalanced class

distribution. In formula (4), iff W appears in the documents of class C_i and W appears in every document of class C_i, it can get the maximum $\text{Gini Text}(W)$, namely $\text{Gini Text}(W) = 1$. This is consistent with domain knowledge. If there is no term $P(W|C_i)^2$, according to the Bayes decision theory of minimum error rate, $P(C_i|W)^2$ is the posterior probability when feature W appears. When the documents distribute evenly where W appears, it gets the minimum $\text{Gini Text}(W)$. But text feature is special, it only gets two values: appearance in the documents or no appearance in the documents. Moreover, according to field knowledge, we omit the circumstance that a feature does not appear in the documents. The class in the training set is always unbalanced and it is opinionated to decide $\text{Gini Text}(W)$ is the minimum. Hence, when we construct the new measure function of Gini index, we consider feature W's condition probability, combining posterior probability and condition probability as the whole measure function to depress the affection when the class is unbalanced.

4. Classifiers in the experiments

In order to evaluate the new feature selection algorithm, we use three classifiers: SVM (support vector machine), kNN and fkNN to show that our new Gini index algorithm is effective in different classifiers. The algorithms of classifiers can be described as follows.

4.1. kNN classifier

The kNN algorithm is to search k documents (called neighbors) that have the maximal similarity (cosine similarity) in training sets. According to what classes these neighbors are affiliated with, it grades the test document's candidate classes. The similarity between the neighbor document and the test document is taken as this class weight of neighbor documents. The decision function can be defined as follows:

$$\mu_j(X) = \sum_{i=1}^{k} \mu_j(X_i)\,\text{sim}(X, X_i), \tag{5}$$

where $\mu_j(X_i) \in \{0,1\}$ shows whether X_i belongs to $\omega_j(\mu_j(X_i) = 1$ is true) or not ($\mu_j(X_i) = 0$ is false); $\text{sim}(X, X_i)$ denotes the similarity between training document and test document. Then the decision rule is: If $\mu_j(X) = \max_i \mu_i(X)$, then $X \in \omega_j$.

4.2. fkNN classifier

The kNN algorithm in 4.1 can not get better categorization performance, especially when the class is unbalanced. Hence, we adopt the fuzzy theory to improve the kNN algorithm as follows. The reasons of this improvement can consult (Shang, Huang, Zhu, & Lin, in press):

$$\mu_j(X) = \frac{\sum_{i=1}^{k} \mu_j(X_i)\,\text{sim}(X, X_i)\,\frac{1}{(1-\text{sim}(X, X_i))^{2/(b-1)}}}{\sum_{i=1}^{k} \frac{1}{(1-\text{sim}(X, X_i))^{2/(b-1)}}}, \tag{6}$$

where $j = 1, 2, \ldots, c$, $\mu_j(X_i)$ is the membership of known sample X to class j. If sample X belongs to class j then the value is 1, otherwise 0. From this formula, we can see that in reality the membership is using the different distance of every neighbor to the candidate classifying sample to weigh its effect. Parameter b is used to adjust the degree of a distance weight. In this paper we take b's value 2. Then fuzzy k-nearest neighbors' decision rule is: If $\mu_j(X) = \max_i \mu_i(X)$, then $X \in \omega_j$.

4.3. SVM classifier

SVM is put forward by Vapnik (1995). It is used to solve the problem of two-class categorization. Here we adopt the linear SVM, using the method of one-versus-rest to classify the documents. The detailed description can be referred to (Vapnik, 1995).

5. Experiments

5.1. Data collections

We use two corpora for this study: the Reuters-21578 and data set coming from the International Database Center, Department of Computing and Information Technology, Fudan University, China.

In Reuters-21578 data set, we adopt the top ten classes. 7053 documents in training set and 2726 documents in test set. The distribution of the class is unbalance. The maximum class has 2875 documents, occupying 40.762% of training set. The minimum class has 170 documents, occupying 2.41% of training set.

In the second data set, we use 3148 documents as training samples and 3522 documents as test samples. The training samples are divided into document sets A and B. In document set A, the class distribution is unbalance. In these documents, the political documents are 619 pieces, occupying 34.43% of the training document set A, the energy sources documents are only 59 pieces, occupying 3.28% of the training document set A. In training sample B, the class distribution is correspondingly balance. Every class is 150 pieces.

5.2. Experimental settings

For every classifier, in the phase of text preprocess we use information gain, expected cross entropy, the weight of evidence of text and CHI to compare with our improved Gini index algorithm. Every measure function can be described as follows:

Information gain:

$$\text{Inf Gain}(W) = P(W)\sum_{i}^{m} P(C_i|W)\log_2\frac{P(C_i|W)}{P(C_i)} + P(\overline{W})\sum_{i}^{m} P(C_i|\overline{W})\log_2\frac{P(C_i|\overline{W})}{P(C_i)} \tag{7}$$

Expected cross entropy:

$$\text{Cross Entropy}(W) = P(W)\sum_{i}^{m} P(C_i|W)\log_2\frac{P(C_i|W)}{P(C_i)} \tag{8}$$

CHI(χ^2):

$$\chi^2(W) = \sum_{i}^{m} P(C_i) * \frac{N(A_1A_4 - A_2A_3)^2}{(A_1+A_3)(A_2+A_4)(A_1+A_2)(A_3+A_4)} \tag{9}$$

Weight of evidence of text:

$$\text{Weight of Evid}(W) = P(W) \times \sum_{i=1}^{m} P(C_i)\left|\log\frac{P(C_i|W)(1-P(C_i))}{P(C_i)(1-P(C_i|W))}\right| \tag{10}$$

After selecting the feature subset using above measure functions, we use TF–IDF to weight the feature, the formula is as follows:

$$w_{ik} = \frac{tf_{ik} \times \log(N/n_i)}{\sqrt{\sum_{j=1}^{M}[tf_{ik} \times \log(N/n_i)]^2}} \tag{11}$$

In Reuters-21578, $k = 45$, in document set A $k = 10$, in document set B $k = 35$.

5.3. Performance measure

To evaluate the performance of a text classifier, we use $F1$ measure put forward by Rijsbergen (1979). This measure combines recall and precision as follows:

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

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

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

5.4. The experimental results and analysis

The experimental result in Reuters-21578 can be described as Table 1.

From this table, we can see that in SVM and fkNN, Gini index gets the best categorization performance. We can notice that five measure functions show better performance all. In SVM, the micro-F1 difference between the best and the worst is 0.366%, in kNN is 0.294%, in fkNN is 0.477%. In kNN, the Macro-F1 of Gini index is only inferior to information gain, the Micro-F1 of Gini index is only inferior to CHI.

The experimental result in the second data set can be described as Tables 2 and 3.

From Table 2, we can see that the categorization performance in SVM, Gini index is only inferior to CHI and exceed Information Gain, in kNN, the Macro-F1 of Gini index is only inferior to CHI, but the Micro-F1 of Gini index gets the best, in fkNN, Gini index is only inferior to weight of evidence of text.

Table 1
The performance of five feature selection measure functions on top 10 classes

Measure function	SVM		kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Gini index	**69.940**	**88.591**	66.584	85.620	**67.999**	**86.537**
Inf Gain	69.436	88.445	**66.860**	85.326	67.032	86.134
Cross Entroy	69.436	88.445	66.579	85.326	67.518	86.207
CHI	67.739	88.225	66.404	**85.761**	66.846	86.060
Weigh of Evid	68.731	88.481	66.766	85.180	67.509	86.280

Table 2
The performance of five feature selection measure functions on training set A

Measure function	SVM		kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Gini index	91.577	90.941	84.176	**83.043**	84.763	83.856
Inf Gain	91.531	90.708	83.318	81.301	84.346	82.811
Cross Entroy	91.481	90.708	83.318	81.301	84.216	82.578
CHI	**91.640**	**91.057**	**84.491**	82.811	85.256	84.008
Weigh of Evid	91.407	90.825	84.073	82.927	**85.867**	**85.017**

Table 3
The performance of five feature selection measure functions on training set B

Measure function	SVM		kNN		fkNN	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Gini index	91.421	91.222	86.272	**85.222**	87.006	**86.556**
InfGain	**91.799**	**91.556**	**86.326**	**85.222**	**87.305**	86.556
CrossEntroy	91.419	91.222	85.764	85.111	86.999	86.444
CHI	91.238	91.000	85.770	85.000	86.898	86.444
WeighofEvid	**91.799**	**91.556**	85.914	85.111	87.138	86.444

From Table 3, we can find that the categorization performance in SVM, Gini index is only inferior to information gain and weight of evidence of tex, in kNN, the Macro-F1 of Gini index is only inferior to information gain, but the Micro-F1 of Gini index gets the best, in fkNN, the Macro-F1 of Gini index is only inferior to information gain, but the Micro-F1 of Gini index gets the best.

In summary, in some data set, the categorization performance of our improved Gini index gets the best. In another data set, its performance is only inferior to other measure function. As a whole, Gini index shows better categorization performance. From formula (7)–(10), we can find that the computation of Gini index is simpler than other feature selection methods. Gini index has no logarithm computations and only has simple multiplication operations.

6. Conclusion

In this paper, we studied the text feature selection based on Gini index. We compare its performance with the other feature selection methods in text categorization. The experiments show that our improved Gini index has a better performance and simpler computation than the other feature selection methods. It is a promising method for text feature selection. In the future, we will improve this method further and will study how to select different feature selection methods at different data set.

Acknowledgement

This research is partly supported by Beijing Jiaotong University Science Foundation under the Grant 2004RC008.

References

Breiman, L., Friedman, J. H., Olshen, R. A., et al. (1984). *Classification and regression trees*. Montery, CA: Wadsworth International Group.

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

Gupta, S. K., Somayajulu, D. V. L. N., Arora, J. K., & Vasudha, B. (1998). Scalable classifiers with dynamic pruning. In *Proceedings of the 9th international workshop on database and expert systems applications* (pp. 246–251). Washington, DC, USA: IEEE Computer Society.

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

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

Lewis, D.D., Ringuette, M., 1994. Comparison of two learning algorithms for text categorization. In *Proceedings of the third annual symposium on document analysis and information retrieval*. Las Vegas, NV, USA, pp. 81–93.

Mladenic, D., Grobelnik, M., 1999. Feature selection for unbalanced class distribution and Naïve Bayes. In *Proceedings of 16th international conference on machine learning*, San Francisco 258–267.

Mladenic, D., & Grobelnik, M. (2003). Feature selection on hierarchy of web documents. *Decision Support Systems, 35*(1), 45–87.

Rijsbergen, V. (1979). *Information retrieval*. London: Butterworth.

Shang, W., Huang, H., Zhu, H., & Lin, Y. (2005). An improved kNN algorithm—Fuzzy kNN. In *Proceedings of international conference on computational intelligence and security* (pp. 741–746). China: Xi'an.

Shankar, S., Karypis, G. *A feature weight adjustment algorithm for document categorization*. Available from: http://www.cs.umm.edu/~karypis.

Tan, S. (2005). Neighbor-weighted K-nearest Neighbor for Unbalanced Text Corpus. *Expert System with Applications, 28*(4), 667–671.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer.

Yang, Y. (1997). An evaluation of statistical approaches to text categorization. *Information Retrieval, 1*(1), 76–88.

Yang, Y., Pedersen, J.O., 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th international conference on machine learning*, Nashville, USA, pp. 412–420.

Yang, Y., & Lin, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in the information retrieval* (pp. 42–49). New York: ACM Press.