Expert Systems with Applications 28 (2005) 667–671

Expert Systems with Applications

www.elsevier.com/locate/eswa

Neighbor-weighted K-nearest neighbor for unbalanced text corpus

Songbo Tan[a,b,*]

[a]Software Department, Institute of Computing Technology, Chinese Academy of Sciences, P.O. Box 2704, Beijing 100080, People's Republic of China
[b]Graduate School of the Chinese Academy of Sciences, Beijing, People's Republic of China

Abstract

Text categorization or classification is the automated assigning of text documents to pre-defined classes based on their contents. Many of classification algorithms usually assume that the training examples are evenly distributed among different classes. However, unbalanced data sets often appear in many practical applications. In order to deal with uneven text sets, we propose the neighbor-weighted K-nearest neighbor algorithm, i.e. NWKNN. The experimental results indicate that our algorithm NWKNN achieves significant classification performance improvement on imbalanced corpora.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Text classification; K-Nearest neighbor (KNN); Information retrieval; Data mining

1. Introduction

With the ever-increasing volume of text data from Internet, it is an important task to categorize these documents into manageable and easy to understand categories. Text categorization aims to automatically place the pre-defined labels on previously unseen documents. It is an active research area in information retrieval, machine learning and natural language processing. A number of machine learning algorithms have been introduced to deal with text classification, such as K-Nearest Neighbor (KNN) (Yang, 1999; Yang & Lin, 1999), Centroid-Based Classifier (Han & Karypis, 2000), Naive Bayes (Lewis, 1998), Decision Trees (Lewis & Ringuette, 1994) and Support Vector Machines (SVM) (Joachims, 1998).

Many of these standard classification algorithms usually assume that the training examples are evenly distributed among different classes. However, as indicated in (Japkowicz, 2000) unbalanced data sets often appear in many practical applications. In an unbalanced data set, the majority class is represented by a large portion of all the examples, while the other, the minority class has only a small percentage of all examples. When a text classifier encounters an unbalanced document corpus, the classification performance often decreases. Although Yang (1999) and Yang and Lin (1999) argued KNN offers top-notch performance for text categorization in most cases it also suffers from imbalance of text data.

In order to improve the performance of classification algorithms on unbalanced distribution text corpora, some researchers resort to sampling strategies (Singhal, Mitra, & Buckley, 1997; Zhang & Mani). However, the removal of training documents in large categories may lose some important information and always sacrifices the classification performance in some cases.

In this work, we propose Neighbor-Weighted K-Nearest Neighbor (NWKNN) for unbalanced text categorization problems. Instead of balancing the training data, our algorithm NWKNN assigns a big weight for neighbors from small class, and assigns a little weight for neighbors contained in large category. The experimental results indicate that our algorithm NWKNN achieves significant classification performance improvement on imbalanced corpora.

The rest of this paper is constructed as follows: Section 2 describes the traditional KNN classifier. Our algorithm NWKNN is introduced in Section 3. Experimental results are given in Section 4. Finally Section 5 concludes this paper.

* Tel.: +86 10 88449181 713; fax: +86 10 88455011.
E-mail address: tansongbo@software.ict.ac.cn.

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2004.12.023

2. KNN classifier

The KNN text classification approach is quite simple: given a test document d, the system finds the K-nearest neighbors among training documents, and uses the classes of the K-nearest neighbors to weight class candidates. The similarity score of each nearest neighbor document to the test document is used as the weight of the classes of the neighbor document. If several of K-nearest neighbors share a class, then the per-neighbor weights of that class are added together, and the resulting weighted sum is used as the likelihood score of that class with respect to the test document. By sorting the scores of candidate classes, a ranked list is obtained for the test document. Formally, the decision rule in KNN classification can be written as:

$$\mathrm{score}(d, c_i) = \sum_{d_j \in \mathrm{KNN}(d)} \mathrm{Sim}(d, d_j)\delta(d_j, c_i)$$

Above, KNN(d) indicates the set of K-nearest neighbors of document d. $\delta(d_j,c_i)$ is the classification for document d_j with respect to class c_i, that is,

$$\delta(d_j, c_i) = \begin{cases} 1 & d_j \in c_i \\ 0 & d_j \notin c_i \end{cases}$$

For test document d, it should be assigned the class that has the highest resulting weighted sum.

3. Proposed algorithm

3.1. The motivation

For unbalanced text corpora, the majority class tends to have more examples in the K-neighbor set for each test document. If we employ traditional KNN decision rule to classify the test document, the test document tends to be assigned the majority class label. As a result, the big category tends to have high classification accuracy, while the other the minority class tends to have low classification accuracy. Therefore the total performance of KNN will be inevitably harmed. Very intuitively, in order to alleviate the impact of imbalance of text data, we assign a small weight for the neighbors from majority class and relatively large weight for the neighbors contained in small category.

3.2. The NWKNN algorithm

In our experiments, the documents are represented using the vector space model (VSM). In this model, each document d is considered to be a vector in the term-space. The weight of each word is computed using TFIDF (Salton & Buckley, 1988).

For each test document d, we first select K neighbors among training documents contained in K^* categories $\{C_1^d, C_2^d, \ldots, C_{K^*}^d\}$. The weight is obtained by following formula:

$$\mathrm{Weight}_i = \frac{1}{(\mathrm{Num}(C_i^d)/\mathrm{Min}\{\mathrm{Num}(C_l^d) | l = 1, \ldots, K^*\})^{1/\mathrm{Exponent}}}$$

where Exponent > 1. The improved decision rule in NWKNN can be written as:

$$\mathrm{score}(d, c_i) = \sum_{d_j \in \mathrm{KNN}(d)} \mathrm{Weight}_i \, \mathrm{Sim}(d, d_j)\delta(d_j, c_i)$$

Above formula is equivalent to following formula:

$$\mathrm{score}(d, c_i) = \mathrm{Weight}_i \left(\sum_{d_j \in KNN(d)} \mathrm{Sim}(d, d_j)\delta(d_j, c_i) \right)$$

Above, KNN(d) indicates the set of K-nearest neighbors of document d. $\delta(d_j,c_i)$ is the classification for document d_j with respect to class c_i, that is,

$$\delta(d_j, c_i) = \begin{cases} 1 & d_j \in c_i \\ 0 & d_j \notin c_i \end{cases}$$

For test document d, as traditional KNN it should be assigned the class that has the highest resulting weighted sum.

4. Experiment results

4.1. The datasets

In our experiment, we use two corpora: Reuter (http://www.research.att.com/~lewis/reuters21578.html) and TDT2 (TDT2, 1998).

Reuter. The Reuters-21578 Text Categorization Test Collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. We used its subset: one consisting of 55 categories, which has approximately 10,324 documents (Table 1).

TDT2. TDT2 is the NIST Topic Detection and Tracking text corpus version 3.2 released in December 6, 1999 (TDT2, 1998). This corpus contains news data collected daily from nine news sources in two languages (American English and Mandarin Chinese), over a period of 6 months (January–June in 1998). We used only the Chinese news texts. The Chinese corpus is sampled daily from January through June 1998 and includes Voice of American's Mandarin News program, Xinhua newswire and news stories downloaded from Zaobao's web site (www.zaobao.com, www.asianone.com). The text data contains 3208 documents grouped under 20 categories (see Table 2).

Table 1
The distribution of all categories in Reuter

Category	Example	Category	Example	Category	Example	Category	Example	Category	Example
Acq	2186	Cotton	27	Lumber	13	Retail	19	Strategic-metal	19
Earn	3761	Cpi	75	Meal-feed	21	Rubber	40	Sugar	145
Money-fx	574	Bop	47	Carcass	29	Corn	8	Tea	9
Grain	489	Heat	16	Money-supply	113	Silver	16	Tin	32
Crude	483	Hog	16	Nat-gas	48	Jobs	50	Cocoa	59
Trade	441	Housing	16	Nickel	5	Lead	19	Veg-oil	93
Interest	263	Income	7	Oilseed	78	Lei	12	Wheat	21
Ship	204	Instal-debt	5	Orange	18	Livestock	57	Wpi	24
Coffee	124	Copper	62	Pet-chem	29	Alum	48	Yen	6
Gnp	117	Ipi	49	Potato	5	Fuel	13	Zinc	20
Gold	121	Iron–steel	51	Reserves	51	Gas	33	Dlr	37

Table 2
The distribution of all categories in TDT2

Category	Example	Category	Example	Category	Example	Category	Example	Category	Example
1	1789	5	142	9	47	13	11	17	10
2	427	6	62	10	39	14	7	18	16
3	242	7	55	11	20	15	5	19	14
4	239	8	53	12	6	16	5	20	19

4.2. The performance measure

To evaluate a text classification system, we use the $F1$ measure introduced by van Rijsbergen (1979). This measure combines recall and precision in the following way:

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

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

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

For ease of comparison, we summarize the $F1$ scores over the different categories using the macro-averages of $F1$ scores (Lewis, Schapire, Callan, & Papka, 1996):

Macro-F1 $=$ average of within-category $F1$ values

The Macro-$F1$ emphasizes the performance of the system on rare categories. Using Macro-$F1$, we can observe the effect of different kinds of data on a text classification system (Kian Ming Adam Chai, Hwee Tou Ng, & Hai Leong Chieu).

In the same way, we can obtain the Macro-Recall and Macro-Precision as follows:

Macro-Recall $=$ average of within-category Recall values

Macro-Precision

$=$ average of within-category Precision values

4.3. The experimental results

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. This is so called three-fold cross validation.

Figs. 1 and 4 display the performance curve for NWKNN and KNN on Reuter and TDT2 respectively after selecting features using Information Gain (Lewis & Ringuette, 1994). Note that *exponent* takes 3. From the two figures we can see that our algorithm NWKNN beats

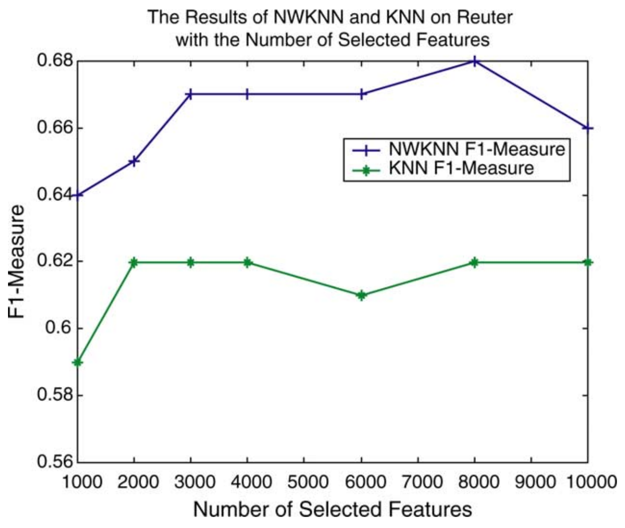

Fig. 1. The results of NWKNN and KNN on Reuter with the number of selected features.

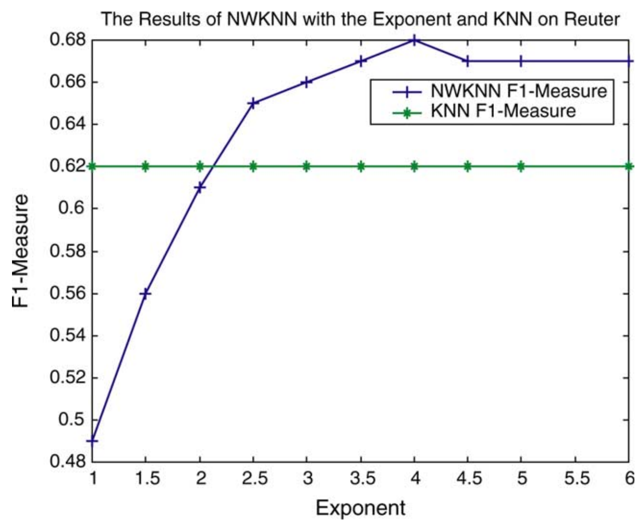

Fig. 2. The Results of NWKNN with the Exponent and KNN on Reuter.

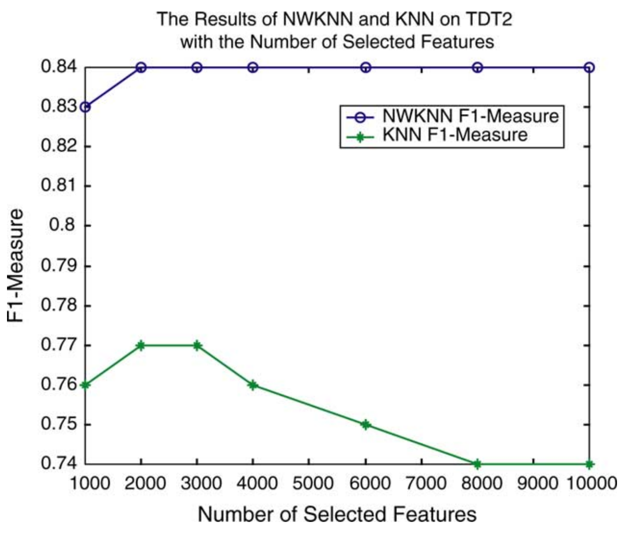

Fig. 4. The Results of NWKNN and KNN on TDT2 with the Number of Selected Features

the KNN under all feature numbers by about 5% on Reuter and about 8% on TDT2.

Figs. 2 and 5 illustrate the performance comparison between NWKNN using different *exponent* and KNN on Reuter and TDT2, respectively. Note that the feature number takes 10,000. From the two Figures, we can find a rule that with the increase of *exponent*, our algorithm NWKNN in the beginning performs better and afterward worse. When *exponent* takes 4.0, our algorithm NWKNN achieves the best results on two datasets and beats KNN by 10% on TDT2.

Figs. 3 and 6 report the Macro-Precision and Macro-Recall comparison between NWKNN using different *exponent* and KNN on Reuter and TDT2, respectively. Note that the feature number takes 10,000. From the two figures, we can discover a phenomenon that the bigger the *exponent*, the higher the Macro-Precision and the lower the Macro-Recall our algorithm NWKNN achieves on the two datasets.

5. Conclusion

In this paper we develop an improved KNN algorithm, i.e. NWKNN, for unbalanced text categorization problems. Comparison between our algorithm NWKNN and traditional KNN is conducted on Reuter and TDT2 corpora. The experimental results indicate that our algorithm NWKNN yields much better performance than KNN. Consequently, our algorithm NWKNN is an

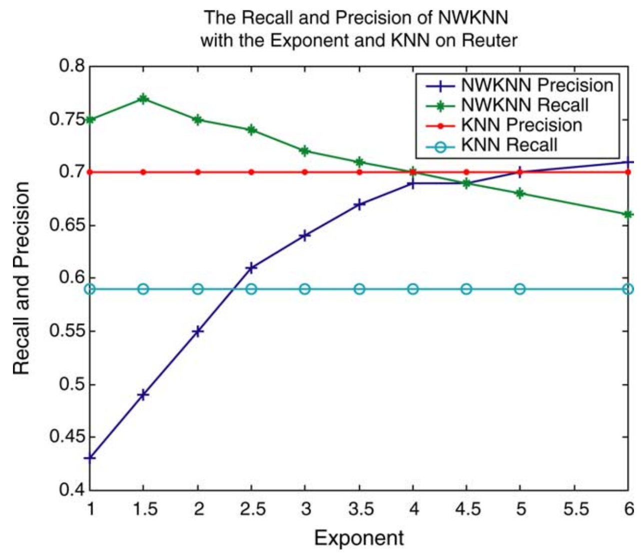

Fig. 3. The Recall and precision of NWKNN with the Exponent and KNN on Reuter.

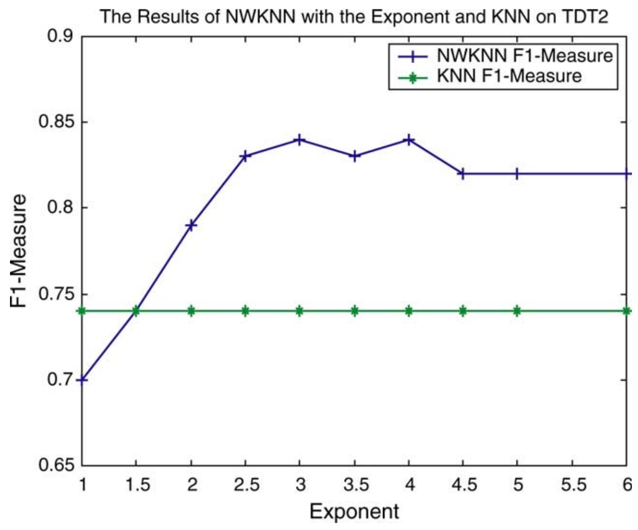

Fig. 5. The Results of NWKNN with the Exponent and KNN on TDT2.

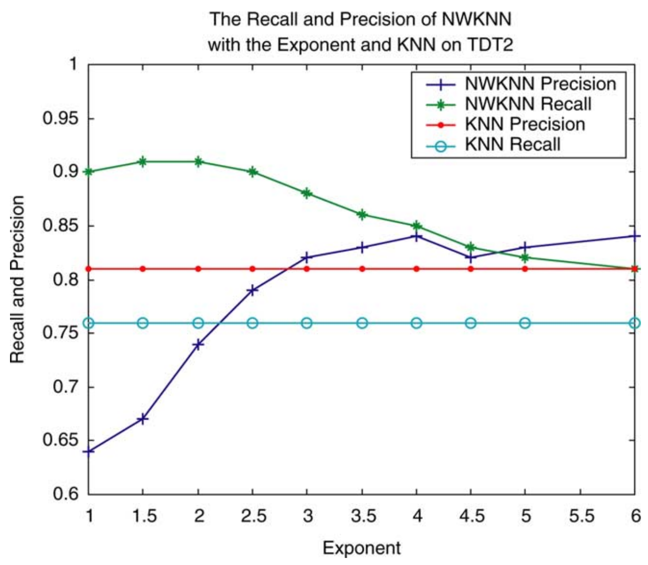

Fig. 6. The recall and precision of NWKNN with the exponent and KNN on TDT2.

effective algorithm for unbalanced text classification problems.

References

http://www.research.att.com/∼lewis/reuters21578.html

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

Japkowicz, N. (2000). Learning from imbalanced data sets: A comparison of various strategies. *Proceedings of Learning From Imbalanced Data Sets, AAAI Work Shop. Technical Report*.

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

Kian Ming Adam Chai, Hwee Tou Ng, & Hai Leong Chieu. Bayesian online classifiers for text classification and filtering.

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

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 (SDAIR'94)*.

Lewis, D. D., Schapire, R. E., Callan, J. P., & Papka, R. (1996). Training algorithms for linear text classiers. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 298–306.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*(5), 513–523.

Singhal, A., Mitra, M., & Buckley, C. (1997). Learning routing queries in a query zone. In *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval* (pp. 25–32) Philadelphia, US.

TDT2 (1998). *Nist topic detection and tracking corpus*www.nist.gove/speech/tests/tdt/tdt98/index.htm.

van Rijsbergen, C. (1979). *Information retrieval*. London: Butterworths.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, *1*(1), 76–88.

Yang, Y., & Lin, X. (1999). A re-examination of text categorization methods. In: *The 22nd Annual International ACM SIGIR Conference on Research and Development in the Information Retrieval*, New York: ACM Press.

Zhang, J., & Mani, I. kNN Approach to unbalanced data distributions: a case study involving information extraction.