Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

# CHINESE TEXT CATEGORIZATION STUDY BASED ON FEATURE WEIGHT LEARNING

**YAN ZHAN[1], HAO CHEN[1], SU-FANG ZHANG[2], MEI ZHENG[3]**

[1]Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding, 071002, China
[2]Teaching and research of section of mathematics, Hebei Information Engineering School, Baoding 071000, China
[3]College of international education, Yanshan University, Qinhuangdao 066004, China
EMAIL: zhanyan@cmc.hbu.cn, mczsf@126.com

**Abstract:**

**Text Categorization(TC) is an important component in many information organization and information management tasks. Two key issues in TC are feature coding and classifier design. The Euclidean distance is usually chosen as the similarity measure in K-nearest neighbor classification algorithm. All the features of each vector have different functions in describing samples. So we can decide different function of every feature by using feature weight learning. In this paper Text Categorization via K-nearest neighbor algorithm based on feature weight learning is described. The numerical experiments prove the validity of this learning algorithm.**

**Keywords:**

**Text Categorization; Feature weight; K-NN**

## 1. Introduction

The automated categorization of texts into topical categories has a long history, dating back at least to 1960. Until the late '80s, the dominant approach to the problem involved knowledge-engineering automatic categorizers, i.e. manually building a set of rules encoding expert knowledge on how to classify documents. In the '90s, with the booming production and availability of on-line documents, automated text categorizations has witnessed an increased and renewed interest [1].

Text Categorization (TC) is an important component in many information organization and information management tasks. Two key issues in TC are feature coding and classifier design.

Feature extraction, which is basically a method of document coding; automatically construct internal representations of documents. The basic principles in document coding are: Firstly, it should be amenable to interpretation by the classifier induction algorithms; Secondly, it should compactly capture the meaning of document and therefore is computationally flexible and feasible [2]. Mutual Information feature selection, filtering approach, and etc, are effective feature selection methods widely used in TC.

As for classifier design problem, a number of statistical classification and machine learning techniques have been applied in TC. These include multivariate regression models, probabilistic Bayesian models, nearest neighbor classifiers, decision trees, adaptive decision trees, neural networks, symbolic rule learning and Support Vector Machine Learning [3]. K-nearest neighbor (K-NN) and Support Vector Machine (SVM) have been reported as the top performing methods for text categorization [4].

This paper reports one statistical machine learning methods, namely K-NN to Chinese text categorization, moreover, we use feature weight learning to improve the traditional neighbor algorithm.

## 2. K-nearest neighbor algorithm

K-nearest neighbor is a classification method based on statistic theory [5]. It is a method frequently used in data mining classification algorithm. This algorithm assumes all instances correspond to points in the n-dimensional space $R^n$. The nearest neighbors of an instance are defined in terms of the standard Euclidean distance. More precisely, let an arbitrary instance x be described by the feature vector $<a_1(x), a_2(x)...a_n(x)>$, Where $a_r(x)$ denotes the value of the rth attribute of instance x. Then the distance between two instance $x_i$ and $x_j$ is defined to be $d(x_i, x_j)$, where

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \qquad (1)$$

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

In K-nearest neighbor learning the target function may be either discrete-valued or continuous-valued. The case of learning a discrete-valued target function of the form is Its basal idea is:

i). Initialize training data set, and partition it into the discrete-valued classes or the continuous-valued classes according to the classification standard that has already existed.

ii). Based on the classification of training data set, look for K nearest neighbors for each sample in testing data set using Euclidean distance as the similarity metrics. The sample that has the largest similarity value to the testing data is exactly the nearest neighbor. Generally the number of near neighbor is one or a few.

iii). When the class is continuous-valued, the final output of testing data is the average value of K nearest neighbors; when the class is discrete-valued, the final output of testing data is the class that has the most quantities of near neighbors.

One practical issue in applying K-NN algorithm is that the distance between instances is calculated based on all attributes of the instance (i.e., on all axes in the Euclidean space containing the instance). This lies in contrast to methods such as rule and decision tree learning systems that select only a subset of the instance attributes when forming the hypothesis. The distance will be dominated by a large number of irrelevant attributes.

This difficulty, which arises when many irrelevant attributes are present, is sometimes referred to as the *curse of dimensionality* [6]. K-NN algorithm is especially sensitive to this problem.

One interesting approach to overcoming this problem is to weight each attributes differently when calculating the distance between two instances [6]. This corresponds to stretching the axes in the Euclidean space, shortening the axes that corresponds to less relevant attributes, and lengthening the axes that corresponds to more relevant attributes.

## 3. Feature weight learning

Suppose that each object is identified by a collection of features $\{F_j \ (j=1,\ldots,n)\}$. Then the *i*-th object $e_i$ can be represented as a n-dimensional vector, i.e. $e_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ where $x_{ij}$ corresponds to the value of feature $F_j (1 \le j \le n) \ (i=1,\ldots,N)$.

We can consider the similarity measure associated with a weighted distance $d_{pq}^{(w)}$, which is defined as:

$$d_{pq}^{(w)} = d^{(w)}(e_p, e_q) = \left( \sum_{j=1}^{n} w_j^2 (x_{pj} - x_{qj})^2 \right)^{1/2} \quad (2)$$

Where $w = (w_1, w_2, \ldots, w_n)$ is called the feature weight vector. For each *j*, $w_j$ is nonnegative and is assigned to the *j*-th feature to indicate the importance of that feature. It is noted that the distance defined by equation (2) is just the usual Euclidean metric while all weights are equal to 1. Thus the weighted distance defined in (2) is a generalization of the Euclidean distance. The similarity measure is then defined by the following equation.

$$\rho_{pq}^{(w)} = \frac{1}{1 + \beta \cdot d_{pq}^{(w)}} \quad (3)$$

Where $\beta$ is a positive parameter.

Sanka Pal noted a simple function in [7]: $f(x,y) = x(1-y) + y(1-x) \quad (1 \le x, y \le 1)$. Since $\frac{\partial f}{\partial x} = 1 - 2y$, we can see $\frac{\partial f}{\partial x} > 0$ if $y < 0.5; \frac{\partial f}{\partial x} < 0$ if $y > 0.5$.

According to these characteristics we put forward the following evaluation function:

$$E(w) = \frac{2}{N(N-1)} \sum_{q<p} E_{pq}(w)$$

$$= \frac{2}{N(N-1)} \sum_{q<p} \frac{1}{2} \left( \rho_{pq}^{(w)} (1 - \rho_{pq}^{(1)}) + \rho_{pq}^{(1)} (1 - \rho_{pq}^{(w)}) \right) \quad (4)$$

in which $N$ is the number of objects, $w = (w_1, w_2, \ldots, w_n)$ represents the feature weight vector, $\rho_{pq}^{(w)}$ specified by equation (3) is the similarity between objects $e_p$ and $e_q$, and $\rho_{pq}^{(1)}$ is the value of $\rho_{pq}^{(w)}$ at $w = (1,1,\ldots,1)$. Let all feature weights be equal to 1, we can compute the similarity between two objects by equation (3), called "old similarity". Consider the feature weights as parameters, which vary on the interval $[0, \infty)$. The similarity between two objects can be computed by equation (3), called "new similarity" which depends on the selection of feature weights.

It is also true that by minimizing equation (4), we could improve the intra-similarity and inter-similarity. That is, the average similarity within the same class will increase and the average similarity among diverse class will decrease. In this way we can get more reasonable feature weight to improve classification and clustering result.

The gradient-descent technique [8] is used to minimize equation (4). The training algorithm repeats until

convergence, i.e., until the value of $E$ becomes less than or equal to a given threshold, or until the number of iterations exceeds a certain predefined number.

Since the similarity depends on feature weights, we can adjust (learn) the feature weights to change whether a pair of objects belongs to the same model.

## 4. Experiments and analysis

### 4.1. Experiments in UCI [9]

We apply the learning feature weights into K-NN algorithm, and do some experiments to compare the performance between original and learning feature weights. Then the distance between two instance $x_i$ and $x_j$ is defined to be $d_1(x_i, x_j)$, where

$$d_1(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (w_i * (a_r(x_i) - a_r(x_j)))^2} \quad (5)$$

By using this method, we can decrease the influence of many irrelevant attributes when using K-NN algorithm.

We select five databases from UCI machine learning repository [9], which are shown in Table 1. For these databases, the testing accuracy between after learning and before learning weight will be computed and compared. The result is shown in Table 1.

**Table 1. UCI data set**

| Data-base | Case number | Attributes-number | before learning | after learning |
|---|---|---|---|---|
| Iris | 150 | 4 | 0.98 | 0.99 |
| Rice | 105 | 5 | 0.88 | 0.92 |
| Pima | 768 | 8 | 0.65 | 0.65 |
| Ecoli | 336 | 7 | 0.88 | 0.96 |
| Glass | 214 | 9 | 0.85 | 0.94 |

We can observe the following experiment Analysis and explanation from Table 1:

1) In the numerical experiment above, the result of comparing the testing accuracy between learning feature weights and before shows that learning feature weights can improve performance of K-NN classification algorithm affirmatively.

2) Irrelevant attributes will have less effect on classification result by learning feature weights, even the feature weight can be learned to zero. From this point, learning feature weights is an extension of feature selection. It can not only improve performance of classification algorithm, but also decrease feature dimensions, and eliminate some unnecessary attributes.

3) The amount of improvement for testing accuracy depends on the specified database and the specified features, e.g. database Glass, Ecoli data have improved performance. Because there are obvious irrelevant attributes in those databases, eliminating these irrelevant attributes can make classification results clear and effective.

4) In a word, Table 1 shows that feature weight learning can improve classification performance than before certainly. This proves that it is a validity method to optimize K-NN algorithm.

### 4.2. Experiments in Chinese text Categorization

We also use K-NN algorithm to handle Chinese text categorization based on feature weight learning. We select 300 papers in People's Daily, which belong to computer programming, International News, profile interview, sports news etc. There are good recalls in these experiments. We do these experiments 100 times and have the average results. For these papers, the testing accuracy after weight learning is 88.2% and 80.5% before weight learning weight.

It verifies the validity and advantages of K-NN classification with feature weight learning. But it doesn't have distinct influence for all databases. That is to say, when all of the attributes of a data set are relevant, we may only reference to feature weights learning. "Attributes relevant" means that all the attributes have their effects on classification results and less differ between after feature weight learning and before learning.

Although the increase of feature weights learning is at the price of computation time, it is necessary to do for the sake of the improvement of classification performance, especially when there are many irrelevant attributes in database.

Feature weights learning is certain applicable when

traditional classification algorithm has a higher error rate. All the features of each vector have different functions in describing samples; each dimension has different contribution to classification too. Before we adopting feature weight learning all entire dimensions of weight are 1.

When all of the attributes have unequal functions, namely feature weights corresponding to every attribute are not all 1, it is applicable to using this feature weights learning method. In this way classification result will be improved evidently and the testing accuracy is precise after feature weight learning than before.

## 5. Conclusions

In this paper Text Categorization via K-nearest neighbor algorithm based on feature weight learning is described. Since K-NN algorithm is used extensively to a variety of areas, we can improve classification performance further and makes its widespread application in TC more valuable by optimizing this algorithm.

## Acknowledgements

This work was financially supported by the Science Foundation of Hebei University.

## References

[1] Fabrizio Sebastiani, "A Tutorial on Automated Text Categorisation", Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, pp. 7-35, Buenos Aires, AR

[2] Luigi Galavotti, Fabrizio Sebastiani, Maria Simi, "Feature Selection and Negative Evidence in Automated Text Categorization", Proceedings of the 4 th European Conference on Research and Advanced Technology for Digital Libraries, ECDL-00, 2000

[3] Yang, Y., Pedersen J.P. A, "Comparative Study on Feature Selectioin Text Categorization", Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), 1997, pp. 412-420.

[4] Yiming Yang, Xin Liu, "A re-examination of text categorization methods", Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp.42-49, 1999

[5] H.B. Mitchell, P.A. Schaefer, "A "soft" K-Nearest Neighbor Voting Scheme", International Journal of Intelligent Systems 2001, pp. 459-468

[6] T. M. Mitchell, Machine Learning, New York: McGraw-Hill Companies Inc., 1997. pp. 230~247

[7] J. Basak, R. K. De, S. K. Pal, "Unsupervised feature selection using a neuro-fuzzy approach", Pattern Recognition Letters. 1998. Vol.19, No.11, pp. 997-1006

[8] He Jian-Yong, Foundation of Operational Research, Tsinghua University Press, Beijing, 2000, pp. 301-306.

[9] UCI Repository of machine learning databases and domain theories. FTP address: ftp://ftp.ics.uci.edu/pub/machine-learning-databases