K-Nearest Neighbor Algorithm Optimization in Text Categorization

Shufeng Chen

Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu 611730, China

chenshufeng115@126.com

Abstract. K-Nearest Neighbor (KNN) classification algorithm is one of the simplest methods of data mining. It has been widely used in classification, regression and pattern recognition. The traditional KNN method has some shortcomings such as large amount of sample computation and strong dependence on the sample library capacity. In this paper, a method of representative sample optimization based on CURE algorithm is proposed. On the basis of this, presenting a quick algorithm QKNN (Quick k-nearest neighbor) to find the nearest k neighbor samples, which greatly reduces the similarity calculation. The experimental results show that this algorithm can effectively reduce the number of samples and speed up the search for the k nearest neighbor samples to improve the performance of the algorithm.

1. Introduction

With the rapid growth of the amount of text information, text classification has become the key technology in the fields of information retrieval, knowledge mining and management. At present, great progress has been made in this aspect and a series of classification methods have been proposed. Some famous text categorization methods are Support Vector Machine (SVM), K Nearest Neighbor (KNN), Neural Network, Linear Least Squares Estimator (LLSF), Bayesian and Decision Tree. Among these methods, KNN is a simple, efficient and non-parametric method [1].

The traditional KNN method has higher computational complexity and strong dependence on sample size [2]. In the KNN classification algorithm, it is necessary to calculate the similarity between the sample to be classified and all the samples in the training sample library so as to obtain the K nearest samples. It is well-known that text vector space has high dimensionality, so that for a text classification system with thousands of training samples, the huge amount of computation will seriously hinder the classification speed, it is difficult to meet the actual needs of users, and even cause KNN Algorithms lose usefulness in text classification. In this paper, we reduce the computational complexity of sample similarity and improve the classification speed of KNN by cutting the sample base to improve the practical value of KNN in text classification.

There are three main ways to improve the computational complexity of reducing sample similarity: Reducing the dimensionality of the feature space of the text sample to reduce the computational complexity of the sample similarity[3-4]; reduce the computation of sample similarity by using a small sample pool [5]; improving the speed of KNN algorithm to find k nearest neighbors, this kind of method is to find the nearest neighbor of the sample to be classified in a short time by the fast search algorithm.

In this paper, we first propose a method to optimize the sample selection, and on the basis of this, we quickly search for the K nearest neighbors by the fast search algorithm. In this method, the CURE clustering algorithm is firstly used to obtain a representative sample library S' for original sample S, and then on this superior sample S', determine a reference sample R, next, according to the distance from the reference sample R, all training samples are sorted, and create a corresponding index table. When a sample x of a category to be determined is given, finding k nearest neighbor samples based on the index table and the tree reduces the range and the number of disk starts, thus greatly accelerating the classification speed.

2. The acquisition of representative samples based on CURE clustering algorithm

The CURE algorithm combines the hierarchical method with the partitioning method. He overcomes the problem that most clustering algorithms prefer to find clusters with similar size and circular shape, and poor performance when dealing with abnormal data [6].

This section uses a simple CURE algorithm to obtain a representative sample library. In this method, the training sample library S is divided into a plurality of sample sets according to categories, and then the clusters are clustered according to the CURE algorithm. Based on the obtained clusters, representative samples of each cluster are calculated, and representative samples of all classes are composed new training sample S'. Then, the clustering result of each class is clustered, and the clusters clustered by different classes are recorded (If different classes come together, prove that this class belongs to the junction of different types of samples).

Acquisition procedure of representative sample library based on CURE algorithm:

(1). Suppose the samples in the training sample library S are divided into J classes and divide S into J sample sets S_1, S_2, ..., S_J according to the sample types.

(2). For each sample set Si(i=1,2,...,J) do the following:

①Assuming that $|S_i|$ denotes the number of samples in the sample set S_i, the sample S_i is then equally divided into$\lfloor |S_i|/NL \rfloor$sample set S_{ij} (j = 1, 2, ..., $\lfloor |S_i|/NL \rfloor$), where NL is the maximum allowable sample size for each sample set, $\lfloor |S_i|/NL \rfloor$means the smallest integer greater than or equal to $|S_i/NL|$. In this way, the number of samples in each sample set S_{ij} is less than or equal to NL.

Set to determine whether two clusters belong to the same cluster maximum distance d_{max};

②For each sample set Sij clustering, following are the specific steps:

Taking each sample of S_{ij} as an independent cluster, the representative point of each cluster is the sample itself, searching for the entire space of S_{ij}, combining the clusters whose distance between two clusters is less than d_{max} into one cluster, calculating the average value, and the average of the cluster as a representative point, and instead of the original cluster, then, each cluster has a representative point, that is, the average. The distance between clusters is the distance between the nearest representative sample points from the nearest two clusters. This section uses the weighted Euclidean distance to find the distance between samples, as shown in formula (1):

$$D(u,v) = \min_{x \in u.rep, y \in v.rep} \left\{ \sqrt{\sum_{i=1}^{m} w_i (x_i - y_i)^2} \right\} \tag{1}$$

Where, u and v are two different clusters, u.rep is the set of representative sample points of cluster u, m is the dimension of the sample feature space, x_i is the i th eigenvalue of the sample x, w_i is the weight of feature i, min{} means take the minimum value. During clustering, if a cluster grows too slowly, it is removed as an isolated point and recorded.

③Clusters and isolated point sets of each subset of Si obtained in step ② are clustered again to obtain a new cluster. Calculating the average value q.mean of the representative points of each new cluster q, let r be the representative point of the cluster before clustering contained in the new cluster q, then r will move to the cluster center and become new representative point r' according to the user-defined contraction factor β, calculated as formula(2). The contraction factor β is between (0, 1).

$$r' = r + \beta(r - q.mean) \tag{2}$$

If the number of samples in a cluster is less than the threshold d_{number}, it is removed as an isolated point.

(3). The representative samples obtained from the sample sets of each class are put together to form a new training sample library S'.

3. AKNN algorithm

The main idea of the proposed QKNN algorithm is to sort the samples and search k nearest neighbors in the ordered sample queue to reduce the search k nearest neighbors and further accelerate the classification speed. Therefore, the QKNN algorithm must firstly determine a reference point R, establish an ordered queue according to the distance from each sample to R, and establish an index table; then, given the sample x to be classified, first calculate the distance d_{xR} between x and R, then search the ordered sample queue index table for the range of sample q whose distance R is closest to d_{xR}; and then find q in this range. Taking the sample q as the center, k samples are taken as the k nearest neighbor initial values in the ordered queue of the training samples, then the samples before and after q in the ordered queue are searched, and the search is continuously replaced k nearest neighbor, search to the sample does not meet the conditions so far. At this point we find exactly k nearest neighbors of x.

3.1. Establish an Ordered Linear Space for Training Sample Base

The procedure for establishing an ordered linear space of a sample library with m training samples is as follows:

(1).Choose a random sample as a reference point $R(R_1, R_2, R_3, ..., R_n)$, n is the dimension of the sample feature vector.

(2).The distance d of each sample to R is calculated according to the formula (1), and an ordered queue queue is arranged by inserting and sorting. Each node includes the distance d of the corresponding sample to R, the category and the eigenvector.

(3).In order to consider the time cost of reading the disk when searching, an index table is constructed for the training sample ordered queue. In the index table, only record the 1,1 + L, 1 + 2L, ..., 1 + iL, ... (1 <= i <= [m / L]) position of the sample in the ordered queue and the distance to R. If you do not create an index table, directly operate the ordered queue, due to the large sample size, you need to start the disk to read the data several times, so the time cost is large. The contents of the index table is less and easy to read into memory quickly.

3.2. Search k nearest neighbor samples

Given the text sample feature vectors x $(x_1, x_2, x_3, ..., x_n)$ to be sorted, the steps of searching k nearest neighbor samples of x are as follows:

(1).Calculate the distance d_{xR} between x and R according to equation (1);

(2).The dichotomy is used to determine the sample range closest to R and dxR in the index table and then read these L samples from the disk. Find the sample q closest to R and dxR among these L samples, select k samples centered on q(Assuming that the s-th sample to the s + k-1th sample in the sample queue are selected and the ordered queue k-list is established according to the distance from each sample to the sample to be sorted x, each node in the queue includes the distance of corresponding sample to x and sample category).

(3).In the ordered queue, select k samples as the center and search forward and backward simultaneously to find the exact k nearest neighbor samples. Let d_{max} be the maximum distance of all the samples in queue k-list to x, the distance between sample i and R is d_{iR}, the distance between sample j and R is d_{jR}, and the algorithm for finding the nearest k nearest neighbor samples is as follows:

①According to the formula (1) to calculate the sample i and sample j to x distance d_{ix} and d_{jx};

```
②i←s-1; j←s+k;
while(|dxR-diR|≤dmax or |dxR-djR|≤dmax) and (i≥1 | j≤m) {
    if |dxR-djR|≤dmax and (i≥1) {
        if (dmax>dix) {
            Insert sample i into k-list ordered queue; delete the sample farthest from x in k-list;
        update $d_{max}$; }
        i←i-1; } /*if*/
    if (|dxR-djR|≤dmax) and j≤m {
        if (dmax > djx) {
        Insert sample j into k-list ordered queue; delete the sample farthest from x in k-list; update
    $d_{max}$; }
        j←j+1;
    }/*if*/
}/*while*/
```

③The k samples in k-list are the k nearest neighbors of x.

(4).According to the formula (3) to calculate the weight of each category. Compare the weights of the classes and assign the samples to be sorted to the category with the highest weight. Weight calculation formula:

$$p(x,C_j)=\sum_{i=1}^{k} Sim(a_i,x)Pa(a_i,C_j) \tag{3}$$

Where Sim (a_i, x) is the similarity between the k nearest neighbor samples a_i and x of x,

$$Pa\ (a_i,C_j) = \begin{cases} 1,\ a_i \text{ is a sample of category } \mathrm{C}_j. \\ 0,\ a_i \text{ is not a sample of category } \mathrm{C}_j. \end{cases}$$

4. Simulation

In order to verify the correctness and validity of the algorithm, this paper uses 6500 news essays from 8 categories of Sina websites to verify and test the improved algorithm. Of these, 5,500 were used as training samples and the remaining 1,000 were used as test samples.

Table 1 improved algorithm and KNN find the k nearest neighbor comparison

	Improved algorithm classification time	Traditional KNN classification time	The improved algorithm and the KNN classification results are the same?
K=6	12s	6min48s	same
K=9	14s	6min55s	same
K=18	21s	7min13s	same
K=27	36s	7min48s	same

In this paper, we use the VSM to represent the text features in the simulation experiments and use the improved CHI and feature aggregation in [7] to reduce the dimensionality. After the word segmentation and feature extraction, 5466 feature terms are selected for these short passages, and the 95-dimensional text feature vectors are obtained after dimension reduction. Under the same conditions, the proposed algorithm and the traditional KNN algorithm were used to classify the samples respectively. The processing time is shown in Table 1. It can be seen through practice that the improved algorithm greatly speeds up the classification time, and the k nearest neighbor found is consistent with the traditional KNN.

5. Conclusion

The simulation experiment shows that the improved algorithm greatly improves the speed of text classification. And the k nearest neighbor samples found by the improved algorithm are accurate, and the results of the traditional KNN algorithm are exactly the same, so all the advantages of KNN can be maintained.

The limitation of the improved algorithm in this paper is that it only improves the classification speed of the texts and does not improve the accuracy of the KNN classification. In the improvement of KNN classification performance, further research is needed.

References

[1] COVER T M, HART P E. Nearest neighbor pattern classification [J]. In Trans IEEE Inform Theory, 1967, IT-13:21-27.

[2] Yang Jianliang, Wang Yongcheng. Application of iterative nearest neighbor method based on KNN and automatic retrieval in automatic classification [J]. Intelligence Journal, 2004(2): 137-141.

[3] Vries A D, Mamoulis N, Nes N, et al. Efficient KNN search on vertically decomposed data//Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin. Madison: ACM Press, 2002: 322-333.

[4] Zhang Xiaohui, Li Ying, Wang Huayong, et al. Improved KNN Algorithm for Chinese Text Classification Using Feature Aggregation [J]. Journal of Northeastern University, 2003, 24(3): 229-232.

[5] Li Ronglu, Hu Yunfa. Training Sample Clipping Method Based on Density of KNN Text Classifier. Computer Research and Development, 2004, 41(4): 539-546.

[6] HAN J W, KAMBE M. Data Mining: Concepts and Techniques [M]. Fan Ming, Meng Xiaofeng, translated. Beijing: Mechanical Industry Press, 2001.

[7] Wang Yu, Wang Zhengou.Text Classification Rules Extraction Based on Fuzzy Decision Tree. Computer Applications, 2005, 25 (7): 634-637.