INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 01, January 2024, pp : 648-650

OPTIMIZING THE K-NEAREST NEIGHBOR ALGORITHM FOR TEXT CATEGORIZATION

R Angeeshwari[1]

[1]Dept. of CS, Fatima College, India.

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. Based on 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 exponential growth of textual information, text classification has emerged as a pivotal technology in the domains of information retrieval, knowledge mining, and management. Significant advancements have been achieved in this field, yielding a variety of classification methods, including Support Vector Machine (SVM), K Nearest Neighbor (KNN), Neural Network, Linear Least Squares Estimator (LLSF), Bayesian, and Decision Tree. Notably, KNN stands out as a straightforward, efficient, and non-parametric approach.

However, the conventional KNN method exhibits elevated computational complexity and a pronounced reliance on sample size. In the KNN classification algorithm, determining the K nearest samples necessitates calculating the similarity between the sample to be classified and all samples in the training sample library. Given the high dimensionality of text vector space, especially in text classification systems with numerous training samples, the substantial computational workload significantly impedes classification speed. This challenge renders KNN less practical for real-world user needs and may even render it ineffective in text classification.

The proposed method introduces an optimization in sample selection, utilizing the CURE clustering algorithm to obtain a representative sample library (S') from the original sample (S). Subsequently, a reference sample (R) is determined within this superior sample (S'). Based on the distance from the reference sample (R), all training samples are sorted, and an index table is created. When presented with a category-determining sample (x), finding K nearest neighbor samples is expedited by leveraging the index table and the tree structure, reducing the search range and the number of disk accesses and significantly accelerating the classification speed.

The acquisition of representative samples based on CURE clustering algorithm

The CURE algorithm effectively combines both hierarchical and partitioning methods, overcoming limitations observed in many clustering algorithms that tend to favor clusters with similar sizes and circular shapes, exhibiting suboptimal performance when confronted with anomalous data.

In this section, a simplified version of the CURE algorithm is employed to generate a representative sample library. The approach involves dividing the training sample library (S) into multiple sets based on categories and subsequently applying the CURE algorithm to cluster these sets. Utilizing the resulting clusters, representative samples are computed for each cluster, and a new training sample set (S') is constructed, incorporating representative samples from all classes. Subsequently, the clustering results for each class are further processed, and the clusters formed by different classes are systematically documented.

Acquisition Procedure for Generating a Representative Sample Library Using the CURE Algorithm:

1. Assume that the samples in the training sample library S are categorized into J classes, and subsequently, S is divided into J sample sets, denoted as S1, S2, ..., SJ, based on sample types.
2. For each sample set Si (where i = 1, 2, ..., J), the following steps are executed:

(a) Partition the sample set Si into |Si| / NL sample sets, denoted as Sij (where j = 1, 2, ..., ⌈|Si| / NL⌉), ensuring that the number of samples in each Sij is less than or equal to NL. Here, NL represents the maximum allowable sample size for each set.

(b) Establish a criterion to determine whether two clusters belong to the same cluster, with a maximum distance threshold, dmax.

(c) Perform clustering for each sample set Sij using the following steps:

- Treat each sample in Sij as an independent cluster, designating the sample itself as the representative point.
- Explore the entire space of Sij, combining clusters whose distance is less than dmax into a unified cluster.
- Calculate the average value for the cluster and designate it as the representative point, replacing the original cluster. As a result, each cluster has a representative point, which is the average.
- Measure the distance between clusters as the distance between the nearest representative sample points from the two closest clusters. The weighted Euclidean distance is employed for this purpose.

3. Clusters 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 centre 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).

If the number of samples in a cluster is less than the threshold dnumber, 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'.

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 dxR between x and R, then search the ordered sample queue index table for the range of sample q whose distance R is closest to dxR; 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.

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(R1, R2, R3, ..., Rn), 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.

2. **Search k nearest neighbor samples**

Given the text sample feature vectors x (x1, x2, x3, ..., xn) to be sorted, the steps of searching k nearest neighbor samples of x are as follows:

(1) Calculate the distance dxR 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.

Simulation

To assess the accuracy and effectiveness of the algorithm, this study conducts verification and testing using a dataset comprising 6,500 news articles sourced from 8 categories on Sina websites. Specifically, 5,500 articles are allocated for training purposes, while the remaining 1,000 articles serve as test samples.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 01, January 2024, pp : 648-650

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 study, the Vector Space Model (VSM) is employed to represent text features in simulation experiments, and the dimensionality is reduced using the enhanced CHI method and feature aggregation as outlined in [7]. Following word segmentation and feature extraction, 5,466 feature terms are chosen for these brief passages. Subsequently, dimension reduction yields 95-dimensional text feature vectors. In a comparative analysis under identical conditions, the proposed algorithm and the conventional KNN algorithm are applied for sample classification. The processing time is summarized in Table 1. Observations from practical implementation reveal that the enhanced algorithm significantly accelerates the classification process while maintaining consistency with the traditional KNN in identifying the k nearest neighbors.

2. CONCLUSION

The simulation experiment demonstrates a significant enhancement in the speed of text classification through the improved algorithm. The k nearest neighbor samples identified by the improved algorithm exhibit accuracy comparable to those obtained by the traditional KNN algorithm, thereby preserving all the advantages inherent to KNN. However, it is important to note that the improvement introduced in this paper is specifically focused on accelerating the text classification process and does not address the enhancement of KNN classification accuracy. Further research is required to delve into strategies for improving the overall performance and accuracy of KNN classification.

3. REFERENCES

[1] 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.

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

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

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