An Improved KNN Classification Algorithm based on Sampling

Zhiwei Cheng[1, a], Caisen Chen[1, b], Xuehuan Qiu[1, c] and Huan Xie[1, d]

[1] The Academy of Armored Forces Engineering, Beijing 100072, China;

[a]cheng.zw@mail.scut.edu.cn, [b]caisenchen@163.com,

[c]qiuxuehuan@139.com, [d]2387633126@qq.com

Keywords: KNN, classification algorithm, computational overhead, sampling.

Abstract. K nearest neighbor (KNN) algorithm has been widely used as a simple and effective classification algorithm. The traditional KNN classification algorithm will find k nearest neighbors, it is necessary to calculate the distance from the test sample to all training samples. When the training sample data is very large, it will produce a high computational overhead, resulting in a decline in classification speed. Therefore, we optimize the distance calculation of the KNN algorithm. Since KNN only considers the k samples of the shortest distance from the test sample to the nearest training sample point, the large distance training has no effect on the classification of the algorithm. The improved method is to sample the training data around the test data, which reduces the number of distance calculation of the test data to each training data, and reduces the time complexity of the algorithm. The experimental results show that the optimized KNN classification algorithm is superior to the traditional KNN algorithm.

1. Introduction

In the field of machine learning and data mining, classification is an important way of data analysis, and is an important way to predict and analyze data. Classification is one of the supervised learning methods. By analyzing the training sample data, a classification rule is obtained to predict the type of test sample data. Common classification algorithms are: decision tree, association rules, Bayesian, neural network, genetic algorithm, KNN algorithm [1]. This paper is a study of the traditional KNN algorithm.

KNN is an instance-based inert classification method [2] and have no training process [3]. Because of the simple realization and superior performance, it is widely used in the fields of machine learning and data mining. However, the traditional KNN algorithm has some drawbacks in the testing process of the sample. The traditional KNN algorithm need to calculate the distance of the new sample to all training samples in the classification process. When the test data is large, it will lead to the traditional KNN algorithm become slow, low performance in the forecast analysis. At present, many people have put forward some improved methods for KNN algorithm. Ugur et al. [4] proposed a density-based training sample cutting method. This method makes the sample distribution density evenly and reduces the calculation of KNN. Nikolay K et al. [5] proposed a method through establishing a classification model based on using a central document instead of original samples to reduce the number of samples that need to be similarly calculated by the KNN algorithm. So, it increases the classification speed. Wanita S et al. [6] proposed a reduction algorithm and a merging algorithm that reduces the computational complexity. In the case of not reducing the original accuracy rate, a fast classification algorithm is constructed to achieve the effect of fast classification.

A low performance problem for KNN in calculating the distance from the test sample to all training samples [7]. To solve this problem, in this paper, we propose a method to sample the training samples, which can improve the efficiency of KNN algorithm classification by reducing the time of distance calculation. During the test, training samples were sampled on each axis direction (d1, d2, d3... dn) around the value of every test data. And then we calculate the distance from the test sample to the training sample after sampled and select the-nearest-distance k training data to make the decision of the test data.

2. The Principle of KNN Algorithm

In all classification algorithms, K-nearest neighbor method (KNN) is the simplest and most understandable algorithm. The input of the algorithm is the eigenvector of the instance. By calculating the distance between the new data and the eigenvalues of the training data, we choose the nearest neighbor of K (K> = 1) to judge the classification. If K = 1, the new data is simply assigned to its neighboring class. In the KNN classification algorithm, there are three elements, namely: K value selection, distance measurement, classification decision rules.

(1)K value selection

The choice of K value will have a great influence on the result output of KNN algorithm. When K = 1, the KNN algorithm is called the nearest neighbor algorithm. If the value of K is small, it indicates that a smaller training instance is used to predict, which will increase the estimation error. On the other hand, if the K value is large, it indicates that more training examples are used to predict, which leads to an increase in the approximate error of learning. When K = N, all incoming new instances will be grouped into the same class.

(2)Distance measurement

The KNN algorithm requires that all features be quantifiable. If the data feature has a non-numeric type, the data must be quantified using quantization criteria. To prevent the parameters of the larger parameters overshadowed the smaller parameters, the sample parameters must be normalized. In the n-dimensional real vector space $\mathbb{R}$, the distance between the two instance points reflects the similarity of the two instance points. There are many distance calculation methods, but we generally use the European distance to calculate the distance between two instances. Assume that the general distance is L_p:

$$\begin{gathered} L_p(\vec{x}_i,\vec{x}_j)=(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^p)^{1/p} \\ \vec{x}_i,\vec{x}_j\in\mathbb{R}^n \\ \vec{x}_i=(x_i^{(1)},\ x_i^{(2)},\cdots,\ x_i^{(n)})^T \\ \vec{x}_i=(x_j^{(1)},\ x_j^{(2)},\cdots,\ x_j^{(n)})^T \\ p\geq 1 \end{gathered} \tag{1}$$

①when $p=1$, the distance L_p is the Manhattan distance:

$$L_1(\vec{x}_i,\vec{x}_j)=(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|) \tag{2}$$

②when $p=2$, the distance L_p is the Manhattan distance:

$$L_2(\vec{x}_i,\vec{x}_j)=(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^2)^{1/2} \tag{3}$$

(3)Classification decision rules

The classification decision of KNN algorithm usually adopts the majority voting method. In the selected K training data, the classifier classifies the new instance into the largest proportion of the label. In algorithmic decision making, we can rely on the distance from the new instance to the weighted vote. If the distance is smaller, the greater the weight; if the distance is larger, the smaller the weight. The purpose of classification decisions is to minimize misclassification. From the mathematical analysis, the smallest classification is equivalent to the minimum risk of experience. Set 0-1 loss function for the classification of the loss function, then the classification function is：

f: $\mathbb{R}^n \rightarrow \{c_1,c_2\cdots,c_k\}$ $c_1,c_2\cdots,c_k$ is the training data label, and the error probability is:

$$P(Y\neq f(X))=1-P(Y=f(X)) \tag{4}$$

In the set of the K nearest training points, the category label of the new instance is $c_j(c_j \in \{c_1, c_2 \cdots, c_k\})$, then the misclassification rate is:

$$\frac{1}{k}\sum_{\vec{x}_i \in N_k(\vec{x})} I(y_i \neq c_j) = 1 - \frac{1}{k}\sum_{\vec{x}_i \in N_k(\vec{x})} I(y_i = c_j) \tag{5}$$

In order to achieve the least risk of experience, we should make $\sum_{\vec{x}_i \in N_k(\vec{x})} I(y_i = c_j)$ be the largest, then we can get:

$$c_j = \arg\max_{c_j} \sum_{\vec{x}_i \in N_k(\vec{x})} I(y_i = c_j) \tag{6}$$

The three elements of the KNN algorithm determine the classification accuracy, efficiency, and speed and error rate of the algorithm. The distance calculation determines the time and space complexity of the KNN algorithm.

3. A KNN Optimization Algorithm Based on Sampling

Considering the distance calculation rules of KNN algorithm, the distance between these points will not cause any influence on the decision of the classification algorithm when the training point is very far from the test point. So, we can cut the uninfluential training points before calculating the distance. We propose a trimming method, which is mainly in the coordinate axis for parallel sampling. We take a dimension of the distance L as a sampling of the width of a dimension, the sampling in the two-dimensional coordinates is shown in Figure 1.

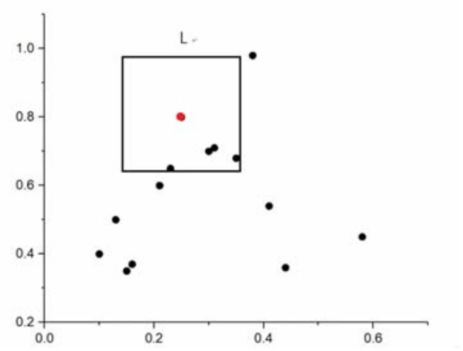

Fig.1 The sampling in the two-dimensional coordinates

We calculate the number of points in the square (excluding the points on the boundary), and we should ensure that the prediction accuracy after sampling does not change significantly. So, we must ensure that N> K. How to determine the width of the sampling is the difficulty of optimizing the KNN algorithm. In this paper, we take the maximum width L_{max} between training points as the upper limit of the cut width, and take the maximum width of 1/4 as the lower limit $L_{min}$of the width.

$$\mathrm{L}_{\max} = \max(|d^{\mathrm{i}}_{\max} - d^{\mathrm{i}}_{\min}|) \quad \text{(D is the value of the i-dimensional)} \tag{7}$$

$$\mathrm{L}_{\min} = \frac{1}{4}\mathrm{L}_{\max} \tag{8}$$

We take the new instance point E as the center, assuming its coordinates are $E(x_1, x_2, \cdots, x_n)$, then we need to calculate the distance of the training point range of values:

$$(E_1(x_1\text{-L}, x_2\text{-L}, \cdots, x_n\text{-L}), E_2(x_1 + \mathrm{L}, x_2 + \mathrm{L}, \cdots, x_n + \mathrm{L})) \tag{9}$$

When the width is taken L, we should guarantee N>K, N is the number of sampling. However, we are sampling in polygons. In fact, we sample points have invalid points, because when the new instance points as the center, the same distance forms a range of a circle. As shown in Figure 2 is a two-dimensional case of sampling, the outer side of the large polygon is V1, circle is V2, radius is L, the small polygon in the circle is V3. We call the point between V1 and V2 dirty point. If there are too many dirty points, we can not guarantee the effective sampling data Valid (N)> K. To ensure that the number of points in the circle must be greater than K, our approach is to ensure $N_1 > K$, N_1 is the number of training data in V3. From the mathematical relationship, we can see that the width of N_1 is $\sqrt{2}L$.

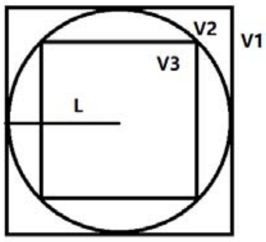

Fig. 2 The sampling in two-dimensional coordinates

To determine the data $N_1 > K$ in V, we take the new instance point E as the center. Assume its coordinates is $E(x_1, x_2, \cdots, x_n)$, then the coordinate range of E is:

$$(E_1(x_1\text{-}\frac{\sqrt{2}}{2}\mathrm{L}, x_2\text{-}\frac{\sqrt{2}}{2}\mathrm{L}, \cdots, x_n\text{-}\frac{\sqrt{2}}{2}\mathrm{L}), E_2(x_1+\frac{\sqrt{2}}{2}\mathrm{L}, x_2+\frac{\sqrt{2}}{2}\mathrm{L}, \cdots, x_n+\frac{\sqrt{2}}{2}\mathrm{L})) \quad (10)$$

N_2 is the number of calculation by optimized KNN algorithm.N_3 is the number of calculation by tradition KNN algorithm. In the number of calculation distance, we compare the optimized KNN with the traditional KNN algorithm, and we can get the following conclusions.

The best situation: $\mathrm{N}_2 \approx \frac{\mathrm{N}_3}{4^n}$ (n represents n dimensions) (11)

The poorest situation: $\mathrm{N}_2=\mathrm{N}_3$ (12)

Through the formula (11), we can get that the the times of distance calculation by the optimized KNN algorithm in the best situation is reduced exponentially. The optimized algorithm has reduced the time and space complexity, so we achieve the goal of algorithm optimization.

But the density of different data is different, which will cause the same cutting method in different data sets will be different efficiency. If the training points near the new instance are many, the sampling can meet the requirements; however, when the training points near the new instance are too few, the training points collected are too few to meet the requirements. In this case, we will increase the sampling width L to ensure that the optimized KNN algorithm can classify correctly. When the new instance is at the edge, the maximum width L cannot satisfy the judgment condition, the algorithm will automatically use the traditional KNN algorithm for classification.

4. Experiment verification and result analysis

The experiment uses the data set of the 2008 Sogou news corpus, which contains seven classes, namely history, culture, reading, military, society and law, entertainment and IT. The number of training for each category is 80, the rest for testing. Experimental run platform in the Intel core i5-4210U 1.7GHz, 4G memory Linux x64 bit system. In the experiment, the traditional KNN algorithm and the optimized KNN algorithm are used to classify the data and compare the running time. The first experiment is to change the value of K. When the training set and testing set in is fixed, we compare the running time of two algorithms, as shown in Figure 3. The second experiment is to compare the time of the data classification in different dimension cases. In the case of K = 9, we compare the running time of two algorithms, as shown in Figure 4.

As shown in Fig. 3, it can be seen from the experiment that the consumption time of the two algorithms increases with the K value. The time consuming of the optimized KNN algorithm is significantly lower than that of the traditional KNN algorithm. As shown in Fig. 4, the experiment shows that the KNN algorithm will increase the consumption time as the dimension increases. However, the time consuming of the traditional KNN algorithm increases rapidly with the dimension. The time consuming of the optimized KNN algorithm increases slowly. By comparing these two groups of times, it is proved that the optimized algorithm is superior to the traditional KNN algorithm.

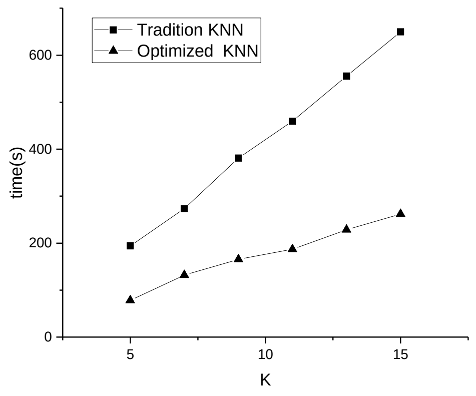

Fig. 3 Time consumption of changing the K value

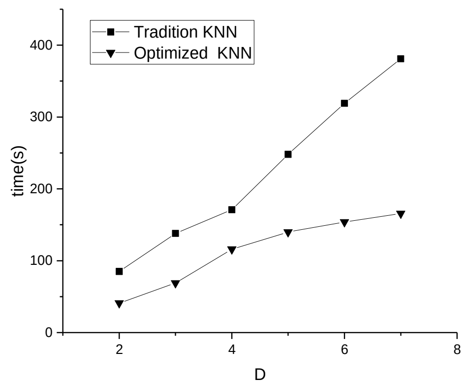

Fig.4 Time consumption of changing the dimension

5. Summary

In this paper, the KNN algorithm is optimized by reducing the number of distances calculated from the test point to the training point. The optimized algorithm achieves the purpose of reducing the running time of the algorithm, which is obviously superior to the traditional KNN algorithm, so it has general value. However, the algorithm is not ideal for edge data processing. If the sampling data cannot meet the requirements, the traditional KNN algorithm is used to classify automatically. How to deal with the edge data is also the direction of further research.

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments. This research was supported by the National Natural Science Foundation of China under Grant No. 61402528. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1]. WuX, Kumarv, Quinlan J R, et al. Top 10 algorithms in data mining[J]. Knowledge and Information Systems, 2008, 14(1):1-37.

[2]. Witten L H, Frank E.Data Mining: Practical Machine Learning Tools and Techniques[M].2nd Edition.New York, USA: Elsevier, 2005

[3]. Tang Jiliang, Gao Huiji, Hu Xia, et al. Exploiting homophily effect for trust prediction [C] //Proc of the 6th ACM International Conference on Wen Search and Data Mining. New York: ACM Press, 2013:53-62.

[4]. Ugur K, Golbeck J. Sunny: a new algorithm for trust inference in social networks using probabilistic confidence models[C]//Proc of National Conference on Artificial Intelligence. 2007:1377-1382.

[5]. Nikolay K, Thomo A. Trust prediction from user-item ratings [J]. Social Network Analysis and Mining, 2013, 3(3): 749-759.

[6]. Wanita S,Nepal S, Paris C. A survey of trust in social networks [J].ACM Computing Surveys, 2013, 45(4): Article 47.

[7]. Wang Aiping, Xu Xiaoyan, Guo Weiwei, et al. Text categorization method based on improved KNN algorithm [J]. Microcomputer & Its Applications, 2011, 30(18): 8-10. (In Chinese).