Hindawi
Complexity
Volume 2021, Article ID 5524388, 10 pages
https://doi.org/10.1155/2021/5524388

Research Article

Improved KNN Algorithm Based on Preprocessing of Center in Smart Cities

Haiyan Wang,[1] **Peidi Xu**,[2] **and Jinghua Zhao**[2]

[1]*College of Computer Science and Technology, Changchun University, Changchun 130022, China*
[2]*College of Computer, Jilin Normal University, Siping 136000, China*

Correspondence should be addressed to Haiyan Wang; wanghy80@ccu.edu.cn

Received 1 March 2021; Revised 26 March 2021; Accepted 27 March 2021; Published 7 April 2021

Academic Editor: Zhihan Lv

The KNN algorithm is one of the most famous algorithms in machine learning and data mining. It does not preprocess the data before classification, which leads to longer time and more errors. To solve the problems, this paper first proposes a PK-means++ algorithm, which can better ensure the stability of a random experiment. Then, based on it and spherical region division, an improved KNN^{PK+} is proposed. The algorithm can select the center of the spherical region appropriately and then construct an initial classifier for the training set to improve the accuracy and time of classification.

1. Introduction

Machine learning and data mining are two extremely important means to realize smart cities. The commonly used classification algorithms in data mining include the Support Vector Machines (SVM) algorithm, the ID3 (Naive Bayesian Classifier) algorithm, the Naive Bayesian Classifier (NBC) algorithm, and the *K*-nearest neighbor (KNN) algorithm [1]. Among them, the KNN algorithm is one of the most famous, simple, and basic algorithms. Because of its easy-to-understand and good classification effect, the KNN algorithm is widely applied in various fields. For example, it has a good classification effect in medical image processing, face recognition, text classification, multimedia communication, smart cities, and other fields [2]. It can be seen that the KNN algorithm has a strong learning ability and excellent application potential for data in different fields and with different characteristics [3].

KNN classification algorithm is a nonparametric learning method [4], which has the advantages of simple principle and few influencing factors. However, many problems are found in the comparison and analysis between KNN and other algorithms in the context of machine learning. First, the efficiency of KNN is low, and all the data should be calculated once for each classification, which takes a long time when the data is large. Secondly, when the sample size is very unbalanced, the prediction accuracy of KNN is low. Thirdly, it takes up a large amount of memory space, because it needs to receive all the data stored to calculate. Finally, the value of *K* is not easy to be selected quickly, and the optimal situation needs to be obtained by comparison. In the following part, we will improve the classification accuracy and efficiency of KNN with the help of the central preprocessing method.

K-means algorithm is an algorithm based on partition method in clustering analysis algorithm. It was put forward by the famous scholar Acqueen in 1967. This algorithm is the most common classical algorithm in cluster analysis. The algorithm is simple, fast, and easy to understand. Based on the *K*-means algorithm, many people make improvements. In 2011, Zhou et al. [5] improved the method of determining the initial clustering center based on the evaluation distance. The optimized algorithm has an obvious effect on the data with outliers.

Our research group improves the performance of the KNN algorithm by optimizing *K*-means under the guidance of local probability in the context of machine learning. With the help of the optimized *K*-means algorithm, the cluster region formed in the sample data set is transformed into multiple spherical regions, and the spherical center of the

spherical region is selected. Then, an initial classifier is constructed for the training set according to the center of the sphere and the corresponding radius. A new training set containing K-nearest neighbor training samples was determined by continuous calculation with the classifier. Finally, the KNN algorithm was optimized and improved on the new training set. The key to improve the algorithm is to add a preprocessing stage to make the final algorithm run with more efficient data and then improve the effect of classification. The experimental results show that the improved KNN algorithm improves the accuracy and efficiency of classification.

The rest of this article is organized as follows. Section 2 discusses related work, followed by the description and analysis of the research problem in Section 3. The optimized K-means algorithm PK-means++ under the guidance of local probability is discussed in Section 4. Section 5 gives the KNN algorithm KNN^{PK+} and the corresponding experimental results based on PK-means++ for optimization of spherical region division, and Section 6 summarizes this paper and the future research directions.

2. Related Work

For the KNN algorithm, researchers have made many improvements. In 2013, Zhu et al. [6] proposed an improvement method based on density. This method reduces the training data and the computational cost of the KNN algorithm by a way of merging. In this way, each class of sample data is clustered into several clusters, and noise sample data is reduced. Then, the sample files with high similarity in each cluster are merged. Subsequently, Saetern and Eiamkanitchat [7] proposed an integrated K-nearest neighbor classification method based on the neuro-fuzzy method. This method improves the KNN algorithm through the neural fuzzy method and the new classification paradigm and achieves good results. In 2015, Ma [8] proposed a parallel K-neighbor classification algorithm based on the Hadoop platform. This algorithm realizes the classification of network public opinion information according to the characteristics of a large amount of network public opinion information data and dispersed content. The next year, Tian [9] proposed an improved weighted KNN algorithm, incorporating the idea of variance into the KNN algorithm and assigning different weight values to feature items with different distributions. The improved algorithm would take a longer operation time, but its classification performance was significantly improved. In the same year, Hu [10] proposed an improved KNN algorithm using the supersphere region and cuboid region division method, which improved the accuracy and efficiency of classification. In recent years, more and more improved KNN tends to the application field. For example, a secure KNN classifier for cloud encrypted data in smart cities is proposed in 2020, which can ensure data privacy, customer information query, and data arrival design [11]. In the same year, a kind of KNN applied to an intrusion detection system was proposed [12]. Meantime, Fauzi et al. applied KNN to autonomous ground vehicle technology and effectively obtained accurate classification results according to the most discriminative features. [13].

For the K-means algorithm, many people are also working hard to improve its performance. In 2011, Bao [14] proposed a hybrid clustering algorithm that embedded the genetic algorithm into the K-means algorithm, aiming at the influence of the initial clustering center in the data clustering analysis of the traditional K-means clustering algorithm. In 2015, Cheng and Lu [15] selected the initial cluster center based on the maximum and minimum distance between data instances and chose the cluster division with the sparsest relative to it based on the sum of the squared errors (SSEs). Then, the number of clusters is determined automatically by stopping cluster splitting according to the trend of SSE variation. In 2016, Gu [16] et al. used the subtraction clustering algorithm to determine the initial clustering center. In 2018, Jiang and Xue [17] proposed an improved K-means clustering algorithm, which firstly determined the number of K needed to be clustered according to the clustered index, and then adopted the idea based on density.

Experiments show that the improved algorithm is more accurate than the original K-means clustering algorithm. Among all the improved algorithms, the improved K-means++ algorithm proposed by Arthur and Vassilvitskii [18] is very important. It randomly selects a variant of the initial cluster center from the data points. Then, the data points are weighted according to the square of the distance between the data points and the selected nearest clustering center, so as to make the selection of the clustering center more clear. Obviously, among the many improved methods, the improvement of the process to obtain the cluster center is still a widely studied method. The acquisition of cluster centers plays an important role in further KNN classification.

3. Problem Description and Analysis

3.1. KNN Classification Algorithm. KNN classification algorithm [3], namely, the K-nearest neighbor algorithm, is one of the most commonly used classification algorithms with the best classification effect. The basic idea is as follows: when entering new data of unknown category to be classified, the category of the data to be classified needs to be determined according to the category of other samples. Firstly, the characteristics of the data to be classified should be extracted and compared with the characteristics of each known category data in the test set. Then, the K-nearest neighbor data were extracted from the test set and the categories in which most of the K data were counted. Finally, the data to be classified is grouped into this category.

KNN classification algorithm, with N training samples $A = \{x_1, x_2, \ldots, x_n\}$, was distributed in S categories W_1, W_2, $\ldots$, W_S. In each category, there is N_i $(i = 1, 2, \ldots, S)$ as training samples. Find K of the nearest samples K_1, K_2, $\ldots$, K_S. The discriminant function is $g_i(x) = k_i$, $i = 1, 2, \ldots, S$, and the category of sample X to be classified is determined by $g_i(x) = \mathrm{Max}(K_i)$. The implementation processes of the KNN classification algorithm are $g_i(x)$as follows:

Step 1. The data were divided into a training sample set and a test sample set. The training sample set was A,

$A = \{a_1, a_2, \ldots, a_n\}$, the category of the sample is expressed as S, $S = \{W_1, W_2, \ldots, W_s\}$, and the test sample set is X, $X = \{x_j | j = 1, 2, \ldots, n\}$.

Step 2. Set the initial k value as the initial nearest neighbor to X.

Step 3. Calculate the distance between the test sample points and all other training sample points.

Step 4. Sort the obtained distance in ascending order and select the appropriate k value.

Step 5. Select the closest k known samples.

Step 6. The category with the highest probability among k known samples was counted.

Step 7. Determine the category of test sample points as the category obtained by Statistics of Step 6.

Although the KNN classification algorithm has many advantages such as easy to understand and good classification effect, it also has many shortcomings. One of them is that the time and space overhead of the algorithm is very high. As the KNN classification algorithm is a kind of lazy algorithm, it will choose to receive all data without any processing before classification. Therefore, each sample data should be taken into account in the calculation, which results in a long calculation time.

In order to illustrate the problems existing in KNN, based on the same sets of data, this paper compares the KNN algorithm and SVM algorithm to study their classification effect and makes a comparative analysis of the experimental results. This will be used as the comparison data of part 5.

3.2. Comparison and Analysis between KNN and SVM. UCI is a commonly used standard test data set library. The two algorithms were tested on six data sets selected from the UCI database [19], Hayes Roth, Iris, Seeds, Pima Indians, Page Blocks, and Shuttle, respectively. The comparison results of classification time and accuracy between the KNN classification algorithm and SVM classification algorithm are studied. The experimental results are as follows.

Table 1 shows the comparison experimental results of classification accuracy, and Table 2 shows the comparison experimental results of classification time. It can be easily seen from the two tables that after the same data are classified by the KNN algorithm and SVM algorithm, the classification time of the SVM algorithm is significantly lower than that of the KNN algorithm. The classification accuracy in the first five data sets is also higher than that of the KNN algorithm, and the accuracy in the sixth data set with a large amount of data is basically the same as the KNN algorithm.

The reason is that the SVM algorithm will train the sample data and then conduct classification prediction after the training. However, the KNN algorithm has no training process for sample data. But SVM algorithm needs to find a suitable hyperplane for classification, and the determination of hyperplane is very complicated. Moreover, when the number of sample data is large, the accuracy of the SVM algorithm will fluctuate greatly due to the selection of hyperplane, resulting in the fluctuation of accuracy. Therefore, inspired by the segmentation of data samples in SVM algorithm classification, this paper improves the classical KNN classification algorithm by tailoring the data samples.

Table 1: Accuracy comparison of classical KNN and SVM.

	Classical KNN	SVM
Hayes Roth	0.93	0.958
Iris	0.941	0.968
Seeds	0.856	0.887
Pima Indians	0.837	0.912
Page blocks	0.857	0.863
Shuttle	0.836	0.814

Table 2: Comparison of classification time between classical KNN and SVM (unit: seconds).

	Classical KNN (s)	SVM (s)
Hayes Roth	0.039	0.028
Iris	0.057	0.042
Seeds	0.263	0.212
Pima Indians	1.189	0.853
Page blocks	11.419	8.159
Shuttle	14,526.970	2,317.946

3.3. KNN Text Classification Experiment. On the other hand, when using the KNN algorithm for text classification, we find that data preprocessing has a great impact on classification accuracy. To illustrate this effect, we did a simple experiment with high-frequency words. The purpose of the experiment is to prove that the removal of a certain number of high-frequency words that are useless to the classification of text data will have a favorable effect on the final classification results in text preprocessing. The experimental results are determined by the relationship between the number of high-frequency words removed and the classification accuracy.

Figures 1–4represent the relationship of the deleted numbers of high-frequency words (deleteNS) and classification accuracy (test_accuracy) in curves. Each curve has several relatively smooth and long segments.

The line of 420–520 in Figure 1 represents that when removing 420–520 high-frequency words, the accuracy curve of the KNN algorithm is relatively stable and the accuracy is close to 0.7. The other curves are interpreted the same way. In order to make the experimental results more broad-spectrum, the stability of classification accuracy should select the part of the line segments in different accuracy curves within the same interval that is both stable and represent higher accuracy.

After many experiments, by observing the comparison of the stationary line segment in the figure, it can be concluded that the accuracy curve is relatively stable and higher when 400–500 high-frequency words are removed; that is, the classification effect is the best. To get more accurate results, we experiment with values every 10 times out of 400–500 and test

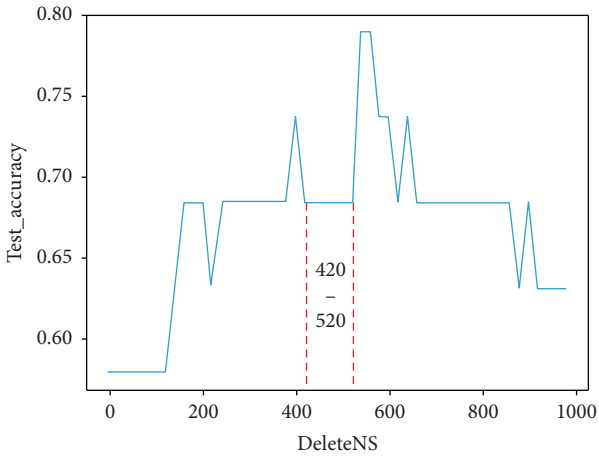

Figure 1: Curve 1 of deleteNS and classification accuracy.

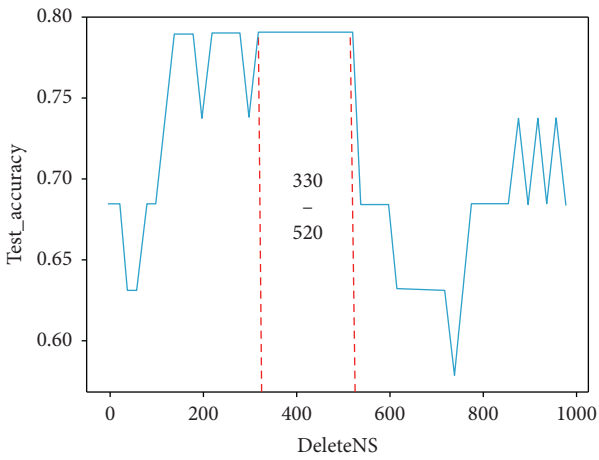

Figure 2: Curve 2 of deleteNS and classification accuracy.

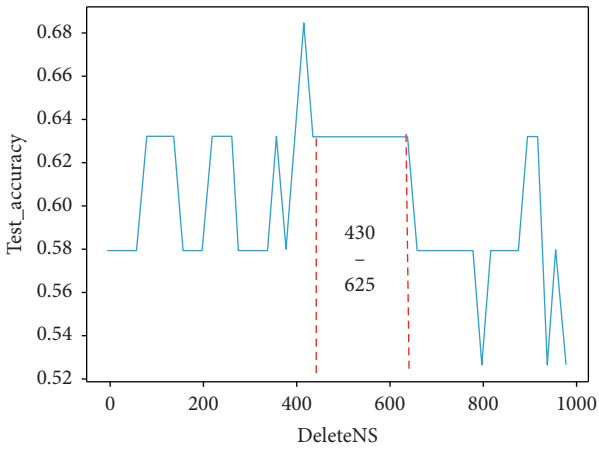

Figure 3: Curve 3 of deleteNS and classification accuracy.

each value 100 times. According to the experimental results in Table 3, when 450 high-frequency words are removed, the number with a classification accuracy of less than 50% is the least; that is, the classification effect is the best.

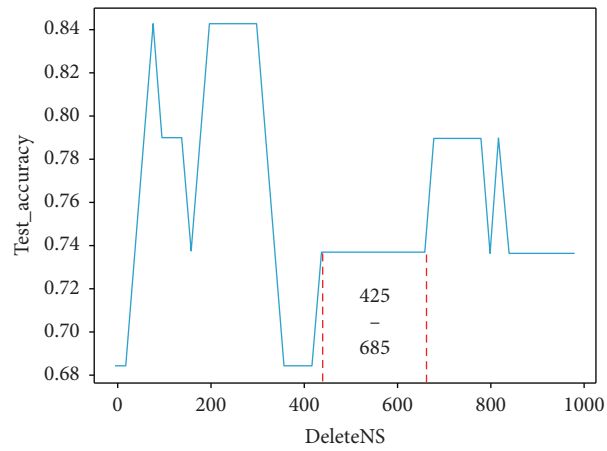

Figure 4: Curve 4 of deleteNS and classification accuracy.

Table 3: Test results of classification accuracy.

Number of deletions	Number of tests	Number of classification accuracies higher than 50%	Number of classification accuracies lower than 50%
400	100	80	20
410	100	87	13
420	100	72	28
430	100	92	8
440	100	75	25
450	100	96	4
460	100	89	11
470	100	87	13
480	100	91	9
490	100	89	11
500	100	77	23

According to the above KNN text classification experiments, it can be seen that the improvement in the data pretreatment stage of KNN algorithm is of great help to the improvement of algorithm accuracy and classification efficiency.

3.4. K-Means++ Algorithm. Clustering algorithm [20] is a kind of unsupervised learning in machine learning, among which the *K*-means algorithm is the simplest and most basic. *K*-means algorithm belongs to the division clustering algorithm. The basic idea is as follows: randomly select *K* samples from *n* data samples as the initial centers, and then calculate the distance between the other samples and the *K* centers. According to the calculated distance, each sample is divided into the set closest to the center; that is, *K* clusters are formed. Then, calculate the center of the newly formed cluster, divide the data according to the new center, and iterate until the center of the cluster no longer changes. Although the principle of the *K*-means algorithm is simple and easy to implement, there are also problems. The initial clustering center needs to be selected artificially, and a different initial clustering may lead to different clustering results. *K*-means++ clustering is an optimization algorithm proposed by Arthur and Vassilvitskii [18] on the basis of the

K-means algorithm. It randomly selects a variant of the initial clustering center from the data points and weights the data points according to the square logarithm of the distance between the data points and the selected nearest clustering center to make the selection of the clustering center more clear. Generally speaking, the K-means++ algorithm has better precision and speed than the K-means algorithm.

Suppose that the data set $X = \{x_1, x_2, x_3, \ldots, x_{n-1}, x_n\}$, the number of clustering is K, and $D(x)$ represents the shortest distance from the data point to the nearest clustering center that has been selected. The workflow of the K-means++ algorithm is as follows.

Step 1. Randomly select a point from data set X as the first clustering center C_1

Step 2. Select X as the next clustering center C_i from data set X in a certain way

Step 3. Repeat Step 2 until K cluster centers are selected

Step 4. Continue to use the standard K-means algorithm for the next calculation

In the process of K-means++ research, there are many specific ways to select the initial clustering center in Step 2 of the workflow, and the most classic ones are as follows:

(1) The vector corresponding to the maximum value of formula (1) is used as the new cluster center [21]:

$$P\left(x_j\right) = \frac{D\left(x_j\right)^2}{\sum x_{j\in}\vartheta_x^{\text{center}} D\left(x_j\right)^2}. \quad (1)$$

(2) Calculate the density of each data sample, sort by the density, take the midpoint of the data sample point with the highest density and its closest point as the initial clustering center, and finally, use the circular domain to divide [22]

(3) Select a seed point, then calculate the distance $D(x_i, y_i)$ between the detection node and the nearest seed node, calculate sum$(D(x_i, y_i))$, and then take a random value that can fall in sum$(D(x_i, y_i))$, calculate random$-=D(x_i, y_i)$, until random<0, then the point is the new cluster center point, and repeat the operation until all K seed nodes have been selected [23]

From the above analysis, we consider that if the characteristics of the clustering algorithm are used to introduce the local probability guidance strategy for the K-means++ algorithm for preprocessing optimization, the clustering effect may be improved.

Therefore, the latter part will operate on the experimental data set with the help of PK-means++ and preprocessing operation. As PK-means++ introduces the local probability guidance strategy on the basis of K-means++, after its improvement, the algorithm can cut out the data more suitable for the KNN classification algorithm experiment, so as to improve the accuracy and efficiency of classification.

4. PK-Means++ Algorithm

4.1. Description. The locally probabilistic PK-means++ (probability K-means++ [24]) algorithm calculates the probability interval occupied by each sample by using the K-means++ algorithm. The farther the point is, the greater the proportion in (0, 1) is, and the higher the probability of randomly picking this interval will be. The steps of algorithm PK-means++ are as follows.

Step 1. Randomly select a point in the array as the center point of the first cluster

Step 2. Iterate over all points in set D, calculate the distance from all points to the center of the nearest cluster, and record the data into the distance array, denoted as $D[1], D[2], \ldots, D[n]$

Step 3. Add up all $D[i]$ ($i = 1, 2, 3, \ldots, n$, $D[i]$ represents the distance between the ith point and the center of the nearest cluster) to get the distance and Sum $(D[n])$, calculate the probability of $D[i]$ in its Sum $(D[n])$ respectively, which is denoted as $P[i]$, express the probability $P[i]$ in (0, 1) through the form of probability segment, and store the starting point of the probability segment in the array PK

Step 4. Take the point in the interval of a random number $rP\,(0 < r < P < 1)$ as the next clustering center point

Step 5. Repeat Step 2 to Step 4 until all the initial centers of K clusters are selected

Step 6. Continue to use the standard K-means algorithm for the next calculation

Take the first cluster with an initial cluster center subscript of 4. The probability of the distance from each data point to the first cluster center is expressed on the interval of (0, 1). The probability segment of the distance from each point to the first initial clustering center is stored in the data group P and $P[4] = 0$. Store the actual point data in the probability segment (0, 1) in the array PK. If the randomly selected point can be found in the interval (PK$[n-1]$, PK$[n]$), then the nth data point will be selected in the next clustering center.

4.2. Experimental Test

4.2.1. Data Set Acquisition. In order to verify the advantages of the algorithm PK-means++ in the SSE, the research team locked the data set on the scattered data set. To ensure relatively dispersed data sets, the experimental team randomly selected 20 two-dimensional data points in a square as data set I (the x-coordinate $x \in (1, 5)$, the y-coordinate $y \in (1, 5)$). The visualization effect of data points is shown in Figure 5. Then, 20 two-dimensional data points in a square are randomly selected as data set II. The visualization effect of data points is shown in Figure 6. Finally, 50 two-dimensional data points in a square are randomly selected as data set III. The visualization effect of data points is shown in Figure 7. As can be seen from the three figures, the data selected in the study are very scattered.

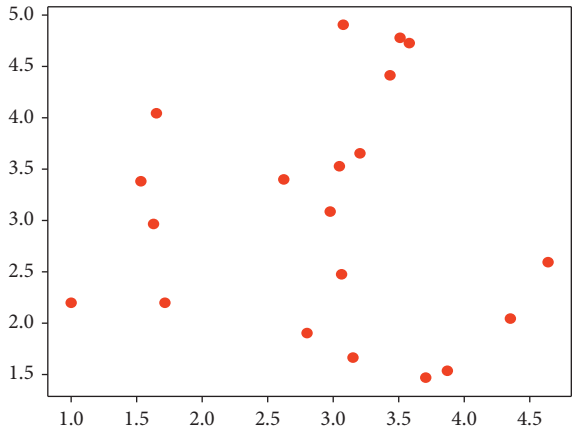

Figure 5: Random points graph of zone (1, 5).

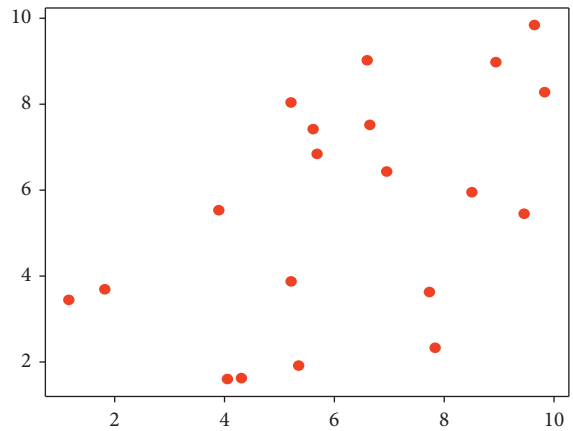

Figure 6: Random points graph of zone (1, 10).

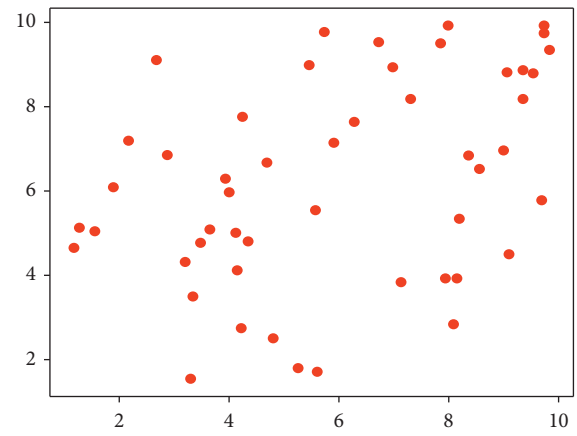

Figure 7: 50 random points graph of zone (1, 10).

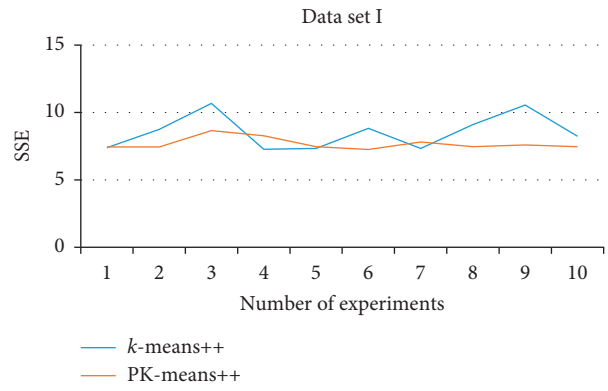

Figure 8: Comparison of the SSE on data set I.

4.2.2. Experimental Analysis. On the basis of the above selected scattered data, in order to fully illustrate the advantages of the PK-means++ algorithm, the *K*-means++ algorithm and PK-means++ algorithm were, respectively, compared and clustered many times to reduce the impact of the random experiment on the experimental results. In order to find the dynamic curve of the error sum of squares, we record the value of the SSE.

The experiment was based on the following machine environment, Intel(R) Core™ i5-7200 processor, with the main frequency of 2.50 GHz and memory of 8.00 GB. The research team conducted 10 experiments, respectively, recorded the SSE 10 times, and drew a line chart for the comparison of the two for different data sets.

The experiment was first performed on data set I. The research team will mark in the figure the SSE calculated by the clustering algorithm of *K*-means++ and PK-means++, respectively. Then, you get a line diagram as shown in Figure 8. It can be clearly seen from the line graph that the SSE calculated by the PK-means++ algorithm changes steadily, while the SSE calculated by the original *K*-means++ algorithm fluctuates relatively more.

This is because *K*-means++ first randomly selects a number less than the distance sum and then takes the random number as the subtractive to do the distance subtraction operation in turn. Finally, the point when the difference is less than 0 is taken as the next initial clustering point. The calculation method of the PK-means++ algorithm is to take points within the distance probability (0, 1). These two algorithms have similar effects on the SSE in the data sets with obvious clustering. For more dispersed data sets, the advantages of the PK-means++ algorithm are highlighted. As the distance between data points is relatively average and the distance difference is small, the PK-means++ algorithm has a smaller random number range than the *K*-means++ algorithm, and the small fluctuation of the number leads to the small fluctuation of the point. In this way, the results of each experiment are close to each other, so as to ensure that the fluctuation of the SSE will not be too obvious, and then present a stable state.

To further demonstrate the advantages of PK-means++, the research team expanded the experimental scale to data sets II and III. The SSEs calculated by *K*-means++ and PK-means++ were observed, respectively, as shown in Figures 9 and 10. Obviously, the PK-means++ algorithm still has an absolute advantage in the level of smoothness. However, the *K*-means++ algorithm still has a large fluctuation range, and the optimal actual data may not be obtained by random values.

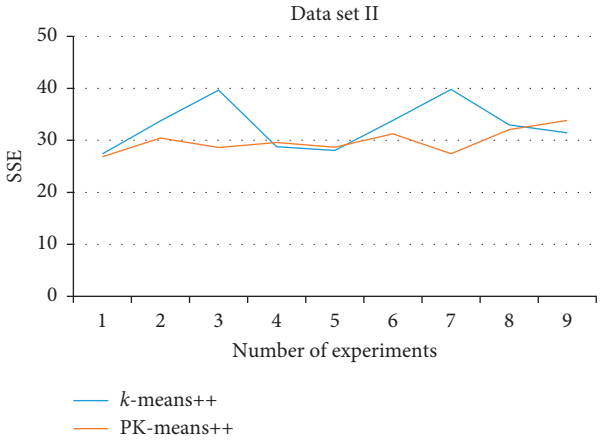

Figure 9: Comparison of the SSE on data set II.

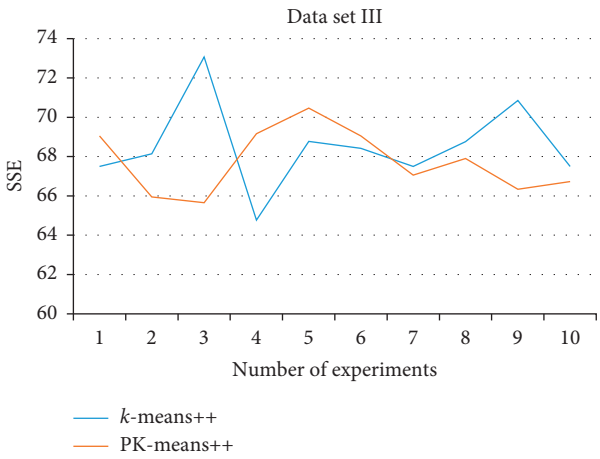

Figure 10: Comparison of the SSE on data set III.

In order to clearly prove the superiority of the PK-means++ algorithm, our team set the experiment on the Watermelon Data Set and locked the number of experiments to 10. Figure 11 is the comparison diagram of error squares and broken lines obtained from 10 experiments of the *K*-means++ algorithm and PK-means++ algorithm for Watermelon Data Set. As shown in the figures, the sum of the SSE calculated by the PK-means++ algorithm fluctuates less than that by the *K*-means++ algorithm, and the results are relatively average. This also fully proves the advantages of the PK-means++ algorithm in the calculation of the SSE, especially for dispersed data.

5. Improved Spherical KNN Algorithm KNN^{PK+}

In order to improve the accuracy, the data of KNN classification are cut by using the method of spherical region division. However, the spherical center is random, so the optimal spherical center is selected by means of the PK-means++ algorithm. This allows it to avoid misclipping in cases where the edges of the valid data are not in the sphere.

5.1. Determination of Initial Classifier. Based on the good performance of PK-means++, our research team improved

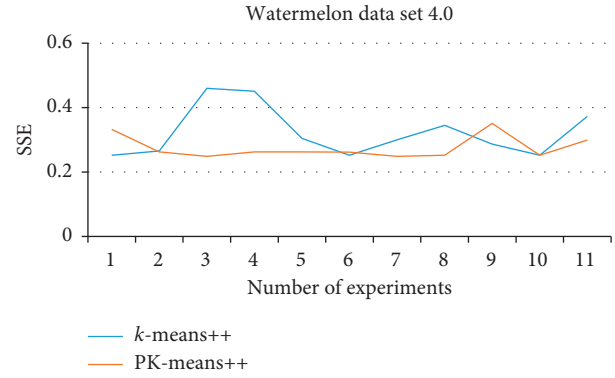

Figure 11: Comparison of the SSE on Watermelon Data Set 4.0.

the KNN algorithm. Then, randomly select some data from the UCI database as the data source. The PK-means++ algorithm is a clustering algorithm, which aims to divide the samples into the sample data set into several clusters. However, the shape of the cluster it forms is not regular, as shown in Figure 12. Since the shape of the region formed by the clustering division algorithm is similar to the sphere, it is more convenient to transform the cluster region formed in the sample data set into a sphere region. The determination processes of the initial classifier are as follows.

Step 1. The centroid vectors of each region in the sample data are calculated by the PK-means++ algorithm, and the appropriate initial center is selected

Step 2. Calculate the distance from all training samples in the data set to each center and put them into the cluster with the closest distance to them

Step 3. Training samples are constantly increasing, and the center point of the cluster is updated timely

Step 4. Calculate the SSE. When the SSE no longer decreases and the samples contained in the cluster basically do not change, the sample in the cluster is terminated to update

Step 5. Take the centroid vector of each cluster as the centroid of the spherical region, calculate the distance from other samples to the centroid, and take the farthest distance as the radius of the spherical region

Step 6. The samples contained in the formed spherical region are saved and used as the initial classifier

5.2. Steps of KNN^{PK+}. First, the center of a spherical region is selected by using the PK-means++ algorithm, and then an initial classifier is constructed for the training set according to the center and corresponding radius. A new training set containing *K*-nearest neighbor training samples is determined by the classifier. Finally, the KNN algorithm is used in the new training set. As the improvement of this KNN algorithm is based on PK-means++, it is named the KNN^{PK+} algorithm. The steps of KNN^{PK+} are as follows.

Step 1. The center point of the spherical region is obtained by using the PK-means++ algorithm.

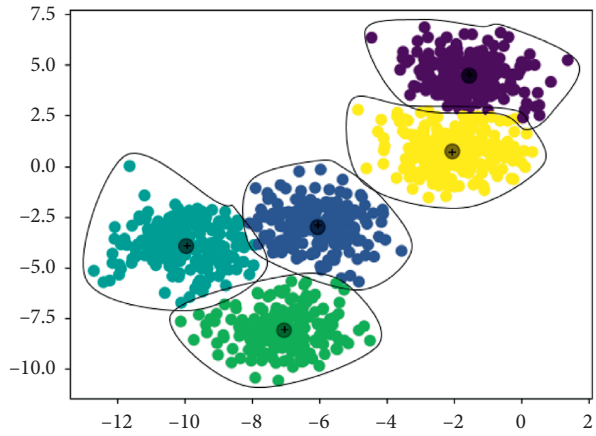

Figure 12: Cluster partitioning map.

Step 2. Calculate the distance between the center point of each spherical region and other samples, and store them in array D. All the values in D are arranged in descending order, and the farthest distance is taken as the radius of the spherical region to form the initial classifier.

Step 3. Calculate the distance between the sample to be tested and each spherical region, and record the maximum distance value.

Step 4. The new training set S is initially empty. If the distance is less than 0 in the calculation process, all the samples in the region will be added to the new training set.

Step 5. Add all samples contained in the closest spherical region into the new training set S.

Step 6. If the distance between the sample to be tested and the adjacent K samples is less than the distance between it and the spherical region without the addition of the new training set, the calculation is terminated; otherwise, go to Step 1.

Step 7. KNN algorithm is used in S to classify test samples.

5.3. The Experiments

5.3.1. The Data Set. In this paper, the improved KNN classification algorithm still experiments on six data sets selected from the UCI database. They are Hayes Roth, Iris, Seeds, Pima Indians, Page blocks, and Shuttle, respectively. The basic information of these six data sets is shown in Table 4.

For the above six data sets, this paper will extract 20% of the data from each data set as test samples, and the remaining 80% of the data will be used as training samples. As the number of data samples of each category is different, the proportion of experimental data selected from each category will try to be close to the proportion of this category in the overall sample number. Thus, the excessive number of samples in a certain category in the selection process will reduce the situation that affects the classification results.

Table 4: Information of six data sets.

Data set	Total number of samples	Number of attributes	Number of categories
Ayes Roth	133	6	3
Iris	150	5	3
Seeds	210	8	3
Pima Indians	769	9	2
Page blocks	5,473	11	5
Shuttle	58,000	10	7

5.3.2. Analysis of Experimental Results. This part of the experiment aims at ensuring the classification efficiency of the KNN classification algorithm and improving the classification accuracy of the algorithm. Therefore, the running time and classification accuracy of the algorithm are analyzed and compared, and the final conclusion is drawn. Classification experiments were carried out on the six data sets in UCI. The initial K value was set as 1, and then, it was increased by 1 each time. Classification calculation was continued, and classification accuracy was recorded. If the K value is still increasing but the accuracy is no longer changing significantly; then, select the K value. The experimental results are listed in Table 5.

Table 5 shows the experimental results of the KNN^{PK+} classification algorithm. To further see the difference between the classical KNN algorithm, SVM algorithm, and KNN^{PK+} algorithm, Table 6 and Figure 13 are made, respectively. Table 6 records the comparison results of the classification accuracy of the three algorithms in the classification of six data sets, and Figure 13 documents the classification time.

Through the above experimental results, it can be observed that the classification accuracy of the KNN^{PK+} algorithm is significantly higher than that of the classical KNN algorithm. The classification time is also reduced, but the reduction is not very large. Compared with the SVM algorithm, the classification time of the KNN^{PK+} algorithm is reduced and the classification accuracy is improved. But in the Pima Indians data set, KNN^{PK+} algorithm accuracy is slightly less than the SVM algorithm, because the content of the data set is Pima medical records, as well as in the past five

Table 5: The classification results of KNN^{PK+}.

	K value	Accuracy (%)	Classification time (s)
Hayes roth	2	98.1	0.03
Iris	2	982	0.034
Seeds	3	89.7	0.246
Pima Indians	3	90.1	0.987
Page blocks	3	91.7	8.489
Shuttle	6	89.6	977.46

Table 6: omparison of classification accuracy of three algorithms

	Classical KNN algorithm (%)	SVM algorithm (%)	KNN^{PK+} algorithm (%)
Hayes Roth	93	95.8	98.1
Iris	94.1	96.8	98.2
Seeds	85.6	88.7	89.7
Pima Indians	83.7	91.2	90.1
Page blocks	85.7	86.3	91.7
Shuttle	83.6	81.4	89.6

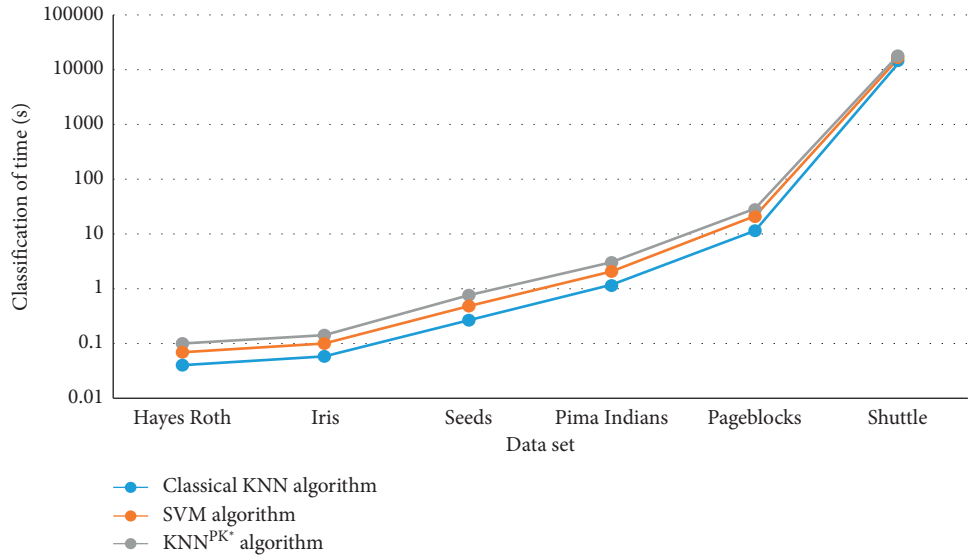

Figure 13: Comparison of the classification time log coordinates of three algorithms.

years, it has diabetes. This is a binary classification problem, and the SVM algorithm itself is a binary classification model, so the classification of the SVM algorithm effect will be better. Therefore, the use of the PK-means++ algorithm to select the classifier with the spherical center formation in the spherical region can effectively avoid the error cut of the effective data in the data set. That is, the KNN^{PK+} algorithm can effectively improve classification accuracy and classification efficiency.

6. Conclusion

In brief, in view of the problem that the KNN classification algorithm does not preprocess data samples, which leads to a long classification time and a decrease in classification accuracy, an improved algorithm KNN^{PK+} for spherical region division is put forward, which is based on PK-means++. SVM algorithm, classical KNN classification algorithm, and KNN^{PK+} algorithm were, respectively, applied to the same data sets, and the classification accuracy and classification time of each algorithm were compared. It is clearly evident from the experiments that our proposed KNN^{PK+} algorithm can effectively improve the accuracy of classification, and the time required for classification is also reduced although the reduction is small. That is, the KNN^{PK+} algorithm has a better classification effect than the classical KNN algorithm and SVM algorithm. The algorithm has some limitations, such as the overlap of intervals. So, the next improvement direction is to adopt multiple methods to select the radius, so as to reduce the interval overlap as much as possible. In the future, we will conduct a more detailed study on the parameters of spherical region division and KNN optimization and further apply these theories to the sensitive issues [25] of smart cities.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All authors contributed equally to this manuscript.

Acknowledgments

This work was supported by the "13th Five-Year" Scientific Planning Project of the Education Department of Jilin Province (no. JJKH20191000K) and the Postgraduate Scientific Research Innovation Plan of Jilin Normal University (no. 201947).

References

[1] L. Cui, "Imbalanced K-NN classification method based on clustering," *Modern Computer*, vol. 33, pp. 6–9, 2020.

[2] X. Wu, S. Wang, and Y. Zhang, "Review of K nearest neighbor algorithm theory and application," *Computer Engineering and Application*, vol. 53, no. 21, pp. 1–7, 2017.

[3] X. Huang, "*An improved KNN algorithm and its application in real-time car-sharing prediction*," M.S. thesis, Dalian University of Technology, Daian, China, 2018.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transaction on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[5] A. Zhou, D. Cui, and Y. Pan, "An optimization initial clustering center of K-means clustering algorithm," *Microcomputer & Its Applications*, vol. 30, no. 13, pp. 1–3, 2011.

[6] Y. Jing, H. Gou, and Y. Zhu, "An improved density-based method for reducing training data in KNN," in *Proceedings of the International Conference on Computational and Information Sciences*, pp. 972–975, Shiyang, China, June 2013.

[7] K. Saetern and N. Eiamkanitchat, "An ensemble *K*-nearest neighbor with neuro-fuzzy method for classification," *Advances in Intelligent Systems and Computing*, vol. 265, pp. 43–51, 2014.

[8] B. Ma, "Research on an improved parallel *k*-neighbor network public opinion classification algorithm," *Microelectronics & Computer*, vol. 32, no. 6, pp. 63–66, 2015.

[9] L. Tian, "*Research on KNN text classification algorithm*," M.S. thesis, Xi'an University of Technology, Xi'an, China, 2016.

[10] J. Hu, "*Improved KNN classification algorithm based on region division*," M.S. thesis, Qingdao University, Qingdao, China, 2016.

[11] P. Vinaybhushan and T. Hirwarkar, "Privacy-perserving KNN classification protocol over encrpted relational data in the cloud," *Advances in Mathematics: Scientific Journal*, vol. 9, no. 7, pp. 4589–4596, 2020.

[12] A. Pathak and S. Pathak, "Study on decision tree and KNN algorithm for intrusion detection system," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, pp. 376–381, 2020.

[13] A. A. Fauzi, F. Utaminingru, and F. Ramdani, "Road surface classification based on LBP and GLCM features using KNN classifier," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1446–1453, 2020.

[14] L. Bao, "*Applied research of hybrid genetic clustering algorithm in customer segmentation*," M.S. thesis, Jinan university., Guangzhou, China, 2011.

[15] W. Cheng and Y. Lu, "Adaptive clustering algorithm based on maximum and minimum distances and SSE," *Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition)*, vol. 35, no. 2, pp. 102–107, 2015.

[16] L. Gu, "A novel locality sensitive k-means clustering algorithm based on subtractive clustering," in *Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 836–839, Beijing, China, August 2016.

[17] L. Jiang and S. Xue, "A k-means algorithm based on optimizing the initial clustering center and determining the *k* value," *Computer & Digital Engineering*, vol. 46, no. 1, pp. 21–24, 2018.

[18] D. Arthur and S. Vassilvitskii, "*k*-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, New Orleans, LA, USA, January 2007.

[19] X. Li and Q. Zhu, "Research on improved BP neural network prediction by adaboost algorithm," *Computer Engineering and Science*, vol. 35, no. 8, pp. 96–102, 2013.

[20] X. Chen, "Analysis and research of clustering algorithm in data mining," *Digital Technology & Application*, vol. 2017, no. 4, pp. 151-152, 2017.

[21] Y. Zhang and Z. Yu, "Video summarization generation algorithm based on k-means++ clustering," *Industrial Control Computer*, vol. 30, no. 7, pp. 129-130, 2017.

[22] W. Chen, D. Xu, and J. Zhang, "Intrusion detection method for industrial control system with optimized support vector machine and k-means++," *Journal of Computer Applications*, vol. 39, no. 4, pp. 1089–1094, 2019.

[23] X. Yu, D. Liu, and J. Yang, "Research on wireless sensor networks clustering algorithm based on k-means++," *Application Research of Computers*, vol. 34, no. 1, pp. 181–185, 2017.

[24] H. Wang, W. Cui, P. Xu, and C. Li, "An optimized *K*-means++ algorithm guided by local probability," *Journal of Jilin University (Science Edition)*, vol. 57, no. 6, pp. 1431–1436, 2019.

[25] Z. Liu, L. Lang, B. Hu, L. Shi, B. Huang, and Y. Zhao, "Emission reduction decision of agricultural supply chain considering carbon tax and investment cooperation," *Journal of Cleaner Production*, vol. 294, no. 4, Article ID 126305, 2021.