Novel KNN Algorithms for Spherical Regions Based on Clustering and Region Division

Haiyan Wang Haiyan Wang
College of Engineering and Technology
Peidi Xu Peidi Xu
Jilin Normal University
Jinghua Zhao Jinghua Zhao (zjh@jlnu.edu.cn)
Jilin Normal University

Research Article

Keywords: artificial intelligence, equal radius spherical region division, KNN algorithm, tabu search algorithm

Posted Date: March 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-355688/v1

Novel KNN Algorithms for Spherical Regions Based on Clustering and Region Division

Haiyan Wang[1], Peidi Xu[2], AND Jinghua Zhao[2]

[1]College of Computer Science and Technology, Jilin Provincial Key Laboratory of Human Health Status Identification and Function Enhancement, Changchun University, Changchun 130022, China

[2]College of Computer, Jilin Normal University, Siping 136000, China

Correspondence: Jinghua Zhao (e-mail: zjh@jlnu.edu.cn).

This work was supported by the National Natural Science Foundation of China under Grant 21773082, and the "13th Five-Year" Scientific Planning Project of the Education Department of Jilin Province under Grant JJKH20191000K, and the Postgraduate Scientific Research Innovation Plan of Jilin Normal University under Grant 201947).

Abstract: The KNN classification algorithm is one of the most commonly used algorithm in the AI field. But classical KNN classification algorithm does not preprocess data before classification calculation, which results in a long time required for classification and a decrease in classification accuracy. To solve the above problems, this paper proposes two improved algorithms, namely KNN^{TS}, and $KNN^{TS\text{-}PK+}$. The two improved algorithms are based on KNN^{PK+} algorithm, which uses PK-Means ++ algorithm to select the center of the spherical region, and sets the radius of the region to form a sphere to divide the data set in the space. The KNN^{PK+} algorithm improves the classification accuracy on the premise of stabilizing the classification efficiency of KNN classification algorithm. In order to improve the classification efficiency of KNN algorithm on the premise that the accuracy of KNN classification algorithm remains unchanged, KNN^{TS} algorithm is proposed. It uses tabu search algorithm to select the radius of spherical region, and uses spherical region division method with equal radius to divide the data set in space. On the basis of the first two improved algorithms, $KNN^{TS\text{-}PK+}$ algorithm combines them to divide the data sets in space. After preprocessing the data by two methods, experiments are carried out on the new data set and the classification results were obtained. Results revealed show that the two improved algorithms can effectively improve the classification accuracy and efficiency after the data samples are cut reasonably.

Keywords: artificial intelligence; equal radius spherical region division; KNN algorithm; tabu search algorithm;

1. Introduction

KNN(K-nearest neighbor algorithm) classification algorithm is a non-parametric learning method [1]. The advantages of the algorithm are its simple principle and few influencing factors, but it also has many shortcomings, such as too much time consuming and space overhead and difficulty in choosing *K* value. That is because KNN classification algorithm does not preprocess the data before classification, but includes all the data. Therefore, many researchers are still exploring it. Wang Zhihua et al. proposed an improved K-modes KNN algorithm, it calculated the distance from the sample to the center of the cluster through string kernel function iteration, and constantly modified the center of the cluster. After the improved K-modes algorithm was used to cluster the data sets, the KNN classification model was established [2].Wang Yanfei et al. proposed an improved KNN algorithm based on clustering and density clipping, which obtained evenly distributed data through the density clipping of data, and then carried out clustering, and then divided into several clusters, transforming globular clusters into super spheres, and finally forming a new training sample set for classification [3].Saetern et al. proposed an integrated K-nearest neighbor classification method based on the neural fuzzy method, improved the KNN algorithm through the neural fuzzy method and the new classification paradigm, and achieved good results [4]. F Lu et al. 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 longer operation time, but its classification performance was significantly improved [5].

KNN classification algorithm [6], namely K-nearest neighbor algorithm, is one of the most commonly used classification algorithms in the AI field. Its basic idea is: when entering new data of unknown category to be classified, the category of the data to be classified should 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 nearest neighbor data of *K* should be taken from the test set to count the categories in which most of the data are located. Finally, the data to be classified should be classified into this category.

KNN classification algorithm is set with N training samples $A= \{x_1, x_2 ..., x_n\}$, distributed in *S* categories w_1, $w_2...w_s$, N_i (i =1,2...s) training samples. Find *K* nearest samples k_1, $k_2...k_s$ out of all the samples, the discriminant function is $g_i(x)=k_i$, i =1,2...*s,*

the category of sample X to be classified is determined by $g_i(x)$= Max (k_i). The specific implementation process of KNN classification algorithm is as follows:

Step 1. The data were divided into training sample set and test sample set. The training sample set was A, A= {a_1, a_2 ..., a_n}, the category of the sample is expressed as S, S= {w_1, w_2 ... w_s}, the test sample set is X, X= {x_j|j=1,2...n}.

Step 2. Set the initial k value as the initial neighbor of X.

Step 3. Calculate the distance between 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 in Step 6statistics.

The rest of this article is organized as follows. Section 2 discusses related work, followed by the description and the experimental analysis of the improved KNN algorithms being researched in Section 3. Section 4 reports the corresponding experimental results and data analysis. Finally, Section 5 concludes the proposed approach, and presents the future research directions.

2. Related Works

2.1. PK-means++ Algorithm

Clustering algorithm [7] is a kind of unsupervised learning in machine learning, among which the simplest and most basic method is the K-means algorithm in partitioning clustering algorithm. The basic idea of K-means algorithm [8] is: Among n data samples, K samples were randomly selected as the initial centers, and the distance between the other samples and the K centers was calculated. Then, according to the calculated distance, each sample is divided into the set closest to the center, that is, K clusters are formed. After that, the center of the newly formed cluster is calculated, and then the data is divided according to the new center, and the data is iterated until the center of the cluster is no longer changed. Since the k-means algorithm has the problem that the initial clustering center needs to be artificially selected, and different initial clustering may lead to different clustering results, this paper chooses the PK-Means ++ algorithm(probability K-Means ++) [9] that optimizes this problem, which is guided by local probability. It can improve the KNN classification algorithm effectively.

PK-means++ algorithm calculates the probability interval occupied by each sample by using K-means ++ algorithm. The interval here refers to the weight of the distance/total distance value in 1, the farther the distance, the greater the weight. That is the farther the point is, the greater the proportion in (0,1), the higher the probability of randomly picking this interval will be. The algorithm steps 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 nearest cluster center, and record the data into the distance array, denoted as: D[1], D[2]...D [n].

Step 3. Put all D[i] (i =1,2,3...n, D[i] refers to the distance from the ith point to the nearest cluster center). Add the distance and Sum (D[n]), and calculate the probability of D[i] in Sum (D[n]), denoted as: P[i]. Then, the probability P[i] is expressed in (0,1) as a probability segment, and the starting point of the probability segment is stored in the array PK.

Step 4. Take the point in the interval of a random number rP(0<rP<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 whose initial cluster center index is 4 as an example, the distance probability of each data point from the first cluster center is expressed on the interval of (0,1), as shown in Figure 1. Where, the probability segment of the distance from each point to the first initial clustering center is stored in the array P and P[4] =0.The actual point data in the probability segment (0,1) is stored in the array PK. If the randomly selected point can be found in the interval (PK[n-1], PK[n]), then the next data point is selected in the next clustering center.

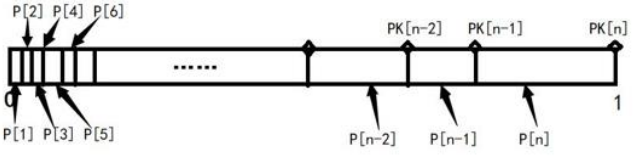

Figure 1. Schematic of array P and array *PK*

2.2. Spherical Region Division Algorithm with Equal Radius

The main method of equal radius spherical region division [10] is to divide the training set sample points in the space into multiple spherical regions with multiple spherical shapes of equal radius. First, set the radius of each spherical shape to the same *r*. Secondly, the distance between all sample points in the spherical region and the center of the sphere is set to be less than or equal to *r*. Each spherical region is then set to contain at least one sample point.

Set the training sample set as *M*, the radius of the spherical region as r, the sample set contained in the spherical region as *N*, and the number of spherical shapes formed in the space as *i* (the initial value of *i* is 1). The specific steps of the equal-radius spherical region division algorithm are as follows:

Step 1. Randomly select a training sample from *M* and make it the center of the *i*th spherical body, then delete it from *M* and add it to *N*.

Step 2. Calculate the distance between the remaining training samples in *M* and the center of the sphere. If the distance is less than the given r, delete it from *M* and add it to *N*; if the distance is greater than *r*, it stays in *M*.

Step 3. *i* can be used as the number of iterations, and $i = i + 1$.

Step 4. Loop Step 1 to Step 3 until *M* is empty and the algorithm ends.

2.3. Tabu Search Algorithm

Tabu Search (TS) [11,12] is characterized by simulating the memory process of human beings and adopting Tabu technology [13,14], that is, the previous work is prohibited to avoid the local optimal situation in local neighborhood Search [15]. The idea of tabu search algorithm is to select the appropriate candidate set in the initial solution neighborhood with the given initial solution and neighborhood. So first initialize the parameters of the model, and set the tabu table to null, then determine the initial solution. If there is a case in the candidate set that the target value corresponding to the candidate solution is better than the current solution, let it replace the current solution and add the corresponding object to the tabu table for modification; if there is no qualified candidate solution, the non-tabu optimal solution is selected as the new current solution, and the corresponding object is added to the tabu table for modification. Repeat the above search steps until the termination principle (the amnesty rules) are met and then stop to get the optimal result. The flow chart of tabu search algorithm is shown in Figure 2.

2.4. Experiments Settings

In order to prove the effect of improvement, the classical KNN algorithm, KNN^{PK+} algorithm, KNN^{TS} algorithm and KNN^{TS-PK+} algorithm were compared in this paper. The experiments are based on six data sets selected from the common UCI standard test database [16]. These data sets are Hayes Roth, Iris, Seeds, Pima Indians, Page Blocks and Shuttle respectively. In addition, the number of samples increased in turn. The basic informations of these six data sets are shown in Table 1 below.

Table 1 Information of six data sets

Data set	The total number of samples	Number of attributes	Number of categories
Hayes Roth	133	6	3
Iris	150	5	3
Seeds	210	8	3
Pima Indians	769	9	2
Page Blocks	5473	11	5
Nursery	12960	8	3
Census Income	48842	14	2
Shuttle	58000	10	7

In these six data sets, this paper will extract 20% data from each data set as test samples, and the remaining 80% data as training samples. As the number of data samples of each category is different, the proportion of the selected experimental data from each category should be close to the proportion of the category in the overall sample number, so as to reduce the incidence of the influence of the classification results caused by the excessive number of samples of a category in the selection process.

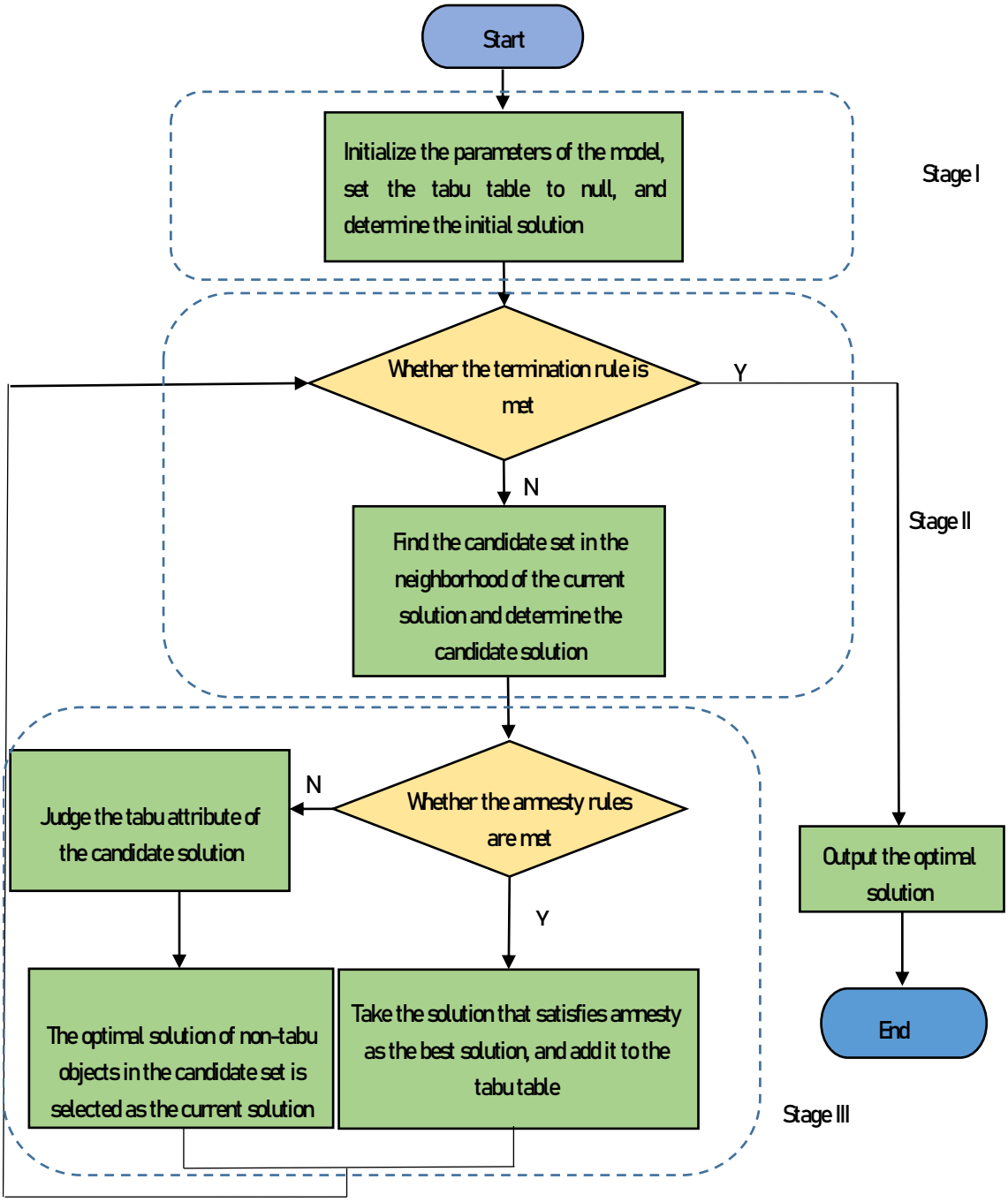

Figure 2. Flow chart of tabu search algorithm

3. Methodology

The calculation method of KNN classification algorithm is to calculate the distance between the test sample and each training sample in the sample set, and then determine the K nearest neighbors t of the test sample category according to the distance value obtained. However, since KNN algorithm does not process any sample data before classification, when the number of samples is too large, the classification efficiency and accuracy will be lower. Therefore, this paper improves the data pretreatment part of KNN classification algorithm.

3.1. Improved KNN^{PK+} Algorithm Based on PK-Means ++ Spherical Region Division

3.1.1. Determination of Initial Classifier

PK-means ++ algorithm is a clustering algorithm, which aims to divide the samples in the sample data set into several clusters, but the shape of the clusters formed by it is not regular. Since the shape of the region formed by the clustering division

algorithm is similar to the sphere, in order to facilitate the calculation, this part converts the region of the cluster formed in the sample data set into multiple spherical regions. The determination process of the initial classifier is as follows:
Step 1. Calculate the centroid vectors of each region in the sample data through PK-means++ algorithm, and select the appropriate initial center.
Step 2. Calculate the distance of all training samples in the data set to each center, and put them into the cluster with the closest distance.
Step 3. Increase the training samples constantly, and update the center point of the cluster timely.
Step 4. Calculate the sum of squared errors. When the sum of squared errors is no longer reduced and the samples contained in the cluster are basically unchanged, the updating of samples in the cluster is ended.
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. Save the samples contained in the formed spherical region and use them as the initial classifier.

3.1.2. Steps of KNN^{PK+} Algorithm

First, PK-Means ++ algorithm is used to select the center of the spherical region, then an initial classifier is constructed for the training set according to the center and corresponding radius, and then a new training set containing *K* nearest neighbor training samples is determined through continuous calculation of the classifier. Finally, KNN algorithm is used in the new training set.The steps of KNN^{PK+} algorithm are as follows:
Step 1. The center point of the spherical region is obtained by using PK-Means ++ algorithm.
Step 2. Calculate the distance between the center point of each spherical region and other samples, and store it in array *D*. Then arrange all the values in *D* in descending order, and take the farthest distance as the radius of the spherical region to form the initial classifier.
Step 3. Calculate the distance of the sample to be tested to each spherical region and record the maximum distance value.
Step 4. The new training set *S* is initially set to be empty. If the distance is less than 0 in the calculation process, all samples in the region will be added to the new training set.
Step 5. Add all samples contained in the closest spherical region to the new training set *S*.
Step 6. Decide whether to continue or not. 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 adding the new training set, then the calculation is terminated, otherwise, go to Step 1.
Step 7. The test samples were classified by KNN algorithm in *S*.

3.1.3. The Experimental Results of Algorithm KNN^{PK+}

The aim of this part is to improve the classification accuracy of KNN on the premise of stabilizing the classification efficiency. Therefore, this section will analyze and compare the running time and classification accuracy of the algorithm and draw the final conclusion. Classification experiments were carried out on the six data sets in UCI. The initial value of *K* is set to 1, and then it increases by 1 each time. Continuously perform classification calculations and then record classification accuracy. If the *K* value is still increasing but the accuracy is no longer changing significantly, select the *K* value. The experimental results are as follows:

Table 2 Comparison of classification accuracy of the two algorithms (%)

	Classical KNN algorithm	KNN^{PK+} algorithm
Hayes Roth	93.0	98.1
Iris	94.1	98.2
Seeds	85.6	89.7
Pima Indians	83.7	90.1
Pageblocks	85.7	91.7
Shuttle	83.6	89.6

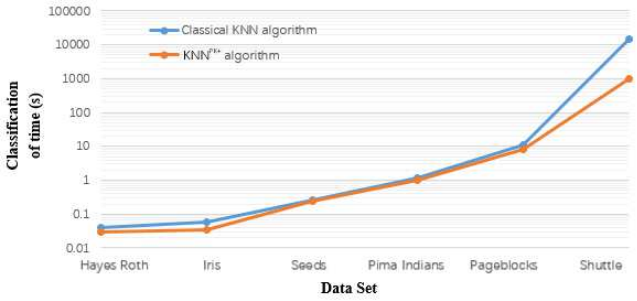

Figure 3. Classification time comparison of the two algorithms

It can be seen from Table 2 and Figure 3 that the classification accuracy of KNN^{PK+} algorithm is significantly higher than that of classical KNN classification algorithm. There has also been a decrease in classification time, but not by much. The reason is that the classical KNN algorithm does not perform any preprocessing on the received data, so some useless or fuzzy data may affect the classification accuracy. By determining the center of the sphere used to divide the spherical region, algorithm KNN^{PK+} processed the data and then obtained a new data set more suitable for classification, which can effectively avoid the error clipping of the effective data in the data set. Therefore, KNN^{PK+} algorithm can effectively improve the accuracy of classification.

Algorithm KNN^{PK+} can remove useless samples and thus reduce the number of training samples.Then, the center of the spherical region is found and all the selected centers are made optimal, which can avoid deleting the effective data of the edge during sample clipping, thus effectively improving the classification accuracy. However, when region division is carried out by Algorithm KNN^{PK+}, the radius of the region selects the farthest distance within the region, which leads to serious overlap between regions. Repeated calculation of many data will increase the amount of calculation. Therefore, although the classification time required by the algorithm is reduced, the classification time is still not optimal.

3.2. Improved KNN^{TS} Algorithm Based on TS-equal Radius Spherical Region Division

In the previous part, although algorithm KNN^{PK+} can effectively improve the accuracy of classification, it does not greatly improve the classification time. Therefore, on the basis of KNN^{PK+} algorithm, the research team considers another way to improve the KNN algorithm. That is, on the premise that the accuracy of KNN classification algorithm is not changed, the classification efficiency of the algorithm is improved.

Through the analysis of algorithm KNN^{PK+} in the previous part, it can be seen that there is too much overlap between multiple spherical regions in the space when the data set is partitioned. If the classification is used directly, there will be a lot of redundant data, thus increasing the computation. Therefore, this section will improve the problem that there are many overlapping parts between regions. By using the spherical region division method with equal radius and the tabu search algorithm to divide the data set, the classification efficiency of KNN algorithm will be improved.

3.2.1. Tabu Search Method to Solve the Radius R Value

After the simple division of the training samples, an initial classifier should be constructed. The initial classifier is determined by the initial radius. In all spherical regions in space, the center of the spherical region is found and the distance from the center of all sample points is calculated. The radius is set as the farthest distance calculated, that is, the maximum distance to the center in the spherical region is kept as the initial radius and all training samples contained therein as the initial classifier. It is very important to select the radius of the spherical region. Too large or too small radius may directly affect the overlap between spherical regions in the space, as well as the number of samples cropped. Therefore, in the calculation after the initial classifier is determined, the radius R value should be selected as an appropriate value to ensure that the number of spherical objects in the space is reasonable.

To determine the radius R value of a spherical region of equal radius, it is necessary to first find the optimal number of spherical regions in space [17].If the number of training samples in the original training set is n and the number of spherical regions is s, then the average number of samples contained in each spherical region is n/s. Calculate the distance between the sample to be tested and the center of each spherical region. Calculate s times in total. After calculating the distance, the number of spherical regions that can be added to the new training set is determined. Assuming that the data in m spherical regions can be added to the new training set, the number of samples in the new training set is mn/s. Since the processed new training set samples are used in the classification of KNN algorithm, the new training set samples need to calculate the mn/s sub-distance. If the time required to calculate a distance is 1s, then the calculation time of KNN classification algorithm is as follows:

$$f(s) = s + mn/s \qquad (1)$$

If the optimal number s of spherical body in the region is required, $f(s)$ needs to be minimized. Therefore, the derivative of $f(s)$ can be obtained as follows:

$$f(s) = 1 - mn/s^2 \quad (2)$$

It is calculated that when $S = \sqrt{mn}$, the minimum value of $f(s)$ is 0, and the optimal number of spherical regions in the region is $\sqrt{mn}$. Tabu search algorithm is used to find the optimal r value of the selected spherical regions with equal radius. The objective function is the absolute value of the difference between square root $\sqrt{mn}$ and the actual number of spherical regions.The steps to solve the radius r value using tabu search algorithm are as follows:

Step 1. Set the training set as Y, the number of training samples in the training set as n, the number of iterations as i, the initial feasible solution as R_{now}, and the initial solution as the current solution (namely the current optimal solution R_{best}); initialize tabu T and empty it. The objective function is abs (R_{now}).

Step 2. If the termination rule is satisfied, stop calculating and go to Step 8; otherwise, go to Step 3.

Step 3. In the neighborhood of the initial solution R_{now}, a 2-OPT operation is performed to select an untabu candidate solution or a tabu solution that satisfies the amnesty rule, and the evaluation value of the solution is better than that of the current solution R_{next}, so that $R_{now}=R_{next}$, and the tabu table is updated.

Step 4. In the neighborhood of the initial solution R_{now}, a 3-opt operation is performed to select an untabu candidate solution or a tabu solution that satisfies the amnesty rule, and the evaluation value of the solution is better than that of the current solution R_{next}, making $R_{now}=R_{next}$, and updating the tabu table.

Step 5. Repeat Step 3 and Step 4 until all solutions in the R_{now} neighborhood are taboo and cannot be forgiven.

Step 6. Search for the best solution that is not taboo and is superior to R_{next} in the neighborhood of R_{now}. If the objective function is less than 0 and abs (R_{now}) is superior to abs (R_{best}), make $R_{best}=R_{now}$ and turn to Step 7.

Step 7. Update tabu T to make the number of iterations i = i +1, and then go to Step 2.

Step 8. Output the optimal solution R_{best} and the algorithm ends.

After the algorithm is finished, the output R_{best} is the optimal radius R value. The contents of Table 1 are the optimal number of spherical regions solved by each data set in the experiment, the radius R value solved by tabu search algorithm, and the error value of the number of spherical regions actually formed and the number of spherical regions during region division.The calculation formula of the error value of the number of spherical regions is as follows:

$$\text{The error value} = \frac{\text{The optimal number of spherical bodies} - \text{The actual number of spherical bodies}}{\text{The optimal number of spherical bodies}} \quad (3)$$

According to the error value of the number of spherical regions in Table 3, it can be seen that the actual number of spherical regions differs little from the optimal number. Number is ideally in solving the optimal number of spherical region formed in the space of not considering the distribution of actual data, so the actual division may be due to data centralized data more dispersed and the actual number is more than the optimal number, or data aggregation and number less than the optimal number of actual condition. Although there is a little error in the number of spherical regions, the radius R value of spherical regions is the local optimal value, so the sample data obtained after the actual division is relatively accurate.

Table 3 Number of spherical regions and r value of radius

	The optimal number of spherical regions	The actual number of spherical regions	Radius value R	Error value of number of spherical regions
Hayes Roth	16	15	0.27	0.0625
Iris	16	16	0.243	0
Seeds	24	24	0.43	0
Pima Indians	46	48	0.47	-0.042
Pageblocks	114	113	0.166	0.009
Shuttle	427	420	0.021	0.016

3.2.2. Steps of KNN^{TS} Algorithm

First, an initial classifier is constructed for the training set by using the spherical region division method with equal radius, and then a new training set with redundant data removed is calculated continuously. Then, KNN algorithm is used in the new training set. Suppose the original training set is S_1, the result training set is S_2, the new training set is S_3, the number of samples is n, the number of iterations is i, and the distance matrix is *Dis*. The steps of the algorithm are as follows:

Step 1. Tabu search algorithm is used to obtain the optimal radius R value;

Step 2. Store the sample from the divided spherical region of equal radius in S_2, and set i =0;

Step 3. After obtaining the spherical region divided according to the radius R value, a point in the spherical region is randomly selected as the center, and the sample distance R_{new} in the ith spherical region is calculated, and R_{new} is taken as the new radius of the spherical region, and $i = i + 1$;

Step 4. repeat Step 3. If $i > n$, then stop the cycle, and store the data sample in the adjusted spherical region in S_2 to form the initial classifier;

Step 5. Calculate the distance between the sample to be tested and each spherical region, store the distance in *Dis*, and then arrange the data in *Dis* in ascending order.

Step 6. Determine the spherical region of K near-training samples and add the training samples contained in the region to S_3.

Step 7. In the new data set S_3, KNN algorithm is used to classify the test samples.

The main task of Step 6 and Step 7 above is to determine the training samples of the new training set. The main process is shown in Figure 4.

3.2.3. Experimental Results of KNN^{TS} Algorithm

Based on our assumptions, the purpose of this section is to improve the classification efficiency of KNN algorithm on the premise that the accuracy of KNN classification algorithm remains unchanged. Experiments are still carried out on the six data sets selected from the UCI database. This part discusses the differences in classification time and accuracy between the classical KNN algorithm, the common spherical region division KNN algorithm with equal radius and the KNN^{TS} algorithm. In the course of the experiment, the Classical KNN algorithm receives all the data without any processing, calculates all the data, and then obtains the classification result. Classical KNN algorithm for spherical region division with equal radius performs simple region division processing in advance, and randomly selects the radius and center of spherical region. KNN^{TS} algorithm also performs pre-region division processing on data, but in addition to randomly selecting the center of the sphere, the regional radius is set through the local optimal value determined by tabu search algorithm. After many experiments, the results of classification time and classification accuracy are as follows:

Table 4 Classification time comparison of three algorithms (Unit: millisecond, ms)

	Hayes Roth	Iris	Seeds	Pima Indians	Pageblocks	Shuttle
Classical KNN	39	57	263	1189	11419	14526970
Common spherical regions division KNN	20	41	176	512	8548	8645635
KNN^{TS}	17	35	128	455	3359	784630

Figure 5 shows the comparison of classification accuracy of the three algorithms, and Table 4 shows the comparison of classification time. In order to illustrate the advantages of the improved algorithms, Figure 6 is specially made to show the comparison of the improvement rates of classification time. Each section of these figures respectively records the percentage improvement of the classification time of the common spherical region division KNN algorithm with equal radius compared with that of classical KNN algorithm(left part) and the percentage improvement of the classification time of KNN^{TS} algorithm compared with that of classical KNN algorithm(right part).

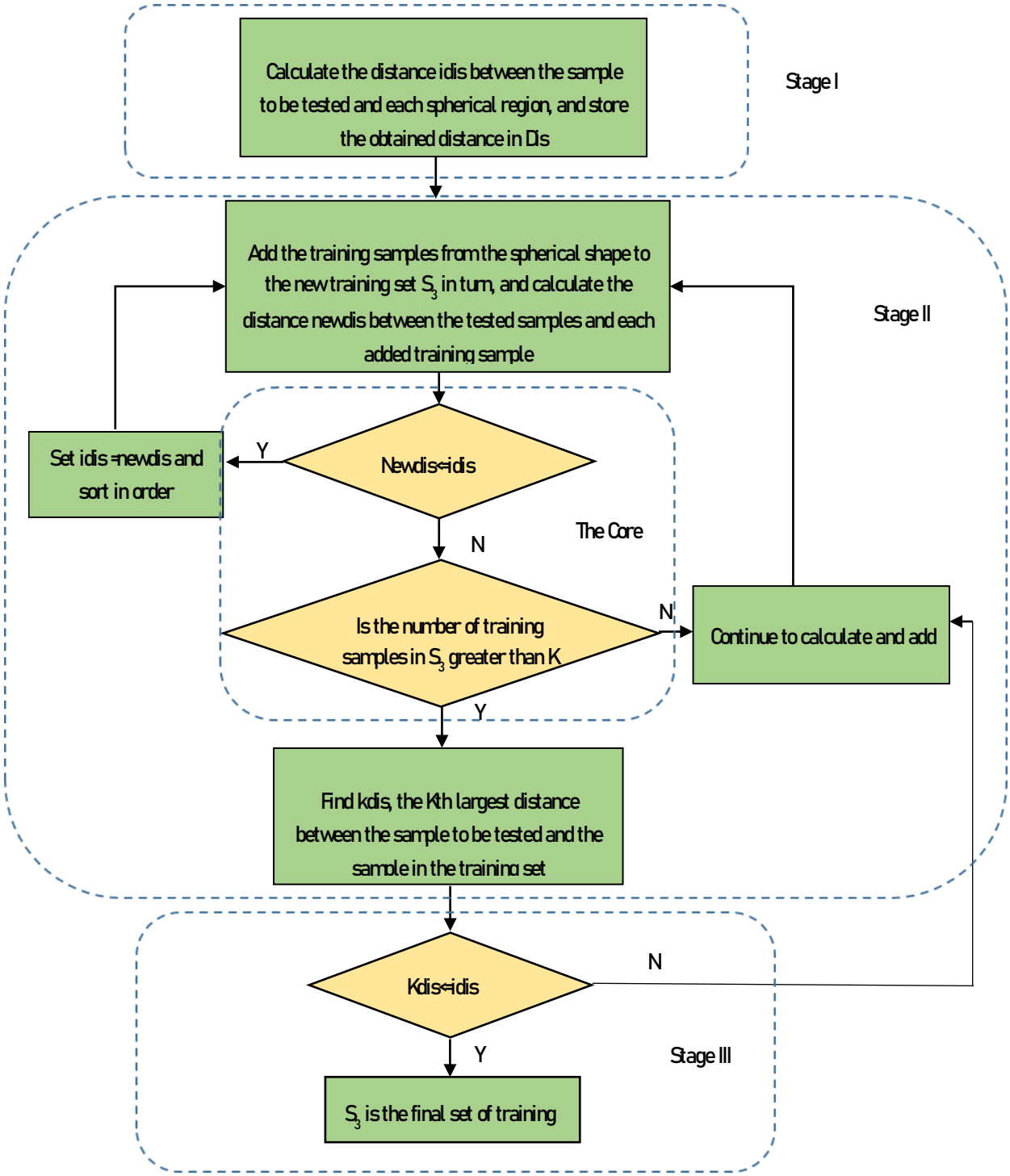

Figure 4. Flow chart of identify new training sets

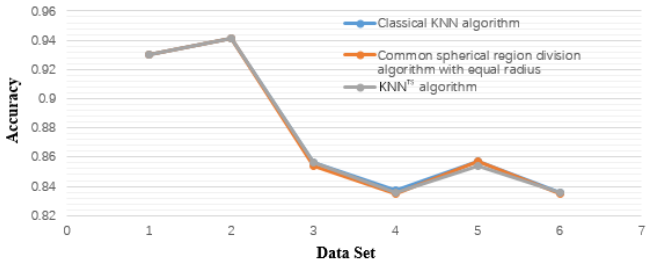

Figure 5. Classification accuracy comparison of the three algorithms

It can be clearly seen from Figure 5 that the classification accuracy of the common spherical region division KNN algorithm with equal radius and KNN^{TS} algorithm is basically unchanged after improvement, which is basically the same as that of classical KNN algorithm. By Table 4, you can see that after using the improved algorithm, the classification times of the common spherical region division KNN algorithm with equal radius and KNN^{TS} algorithm compared with the classical KNN algorithm have greatly improved. Moreover, the percentages of improvement rate are marked as the blue part of Figure 6. From these values, it can be further seen that KNN^{TS} algorithm can save more time than the common spherical region division KNN algorithm with equal radius. Especially when the number of samples keeps increasing, the time required for classification is significantly reduced and the improvement rate gradually increases.

The reason is that the classical KNN algorithm does not do any processing on the data and computations on all the received data, which results in too long classification time. Although KNN algorithm for spherical regions with equal radius can reduce the classification time, it still takes a long time because it cannot determine the appropriate radius and needs to carry out distance calculation for many times. The KNN^{TS} algorithm avoids the above problems and reduces the overlapping between regions, thus reducing the required time. Therefore, KNN^{TS} algorithm can effectively improve the classification efficiency while ensuring the classification accuracy is basically unchanged.

KNN^{TS} algorithm can calculate the appropriate radius required in the classified spherical region, solve the problem of partial data duplication caused by excessive overlap between regions, and effectively improve the classification efficiency of the algorithm without reducing the classification accuracy. However, KNN^{TS} algorithm selects the center of the spherical region randomly in the regional division, which may lead to errors in the effective sample data and affect the accuracy of classification.

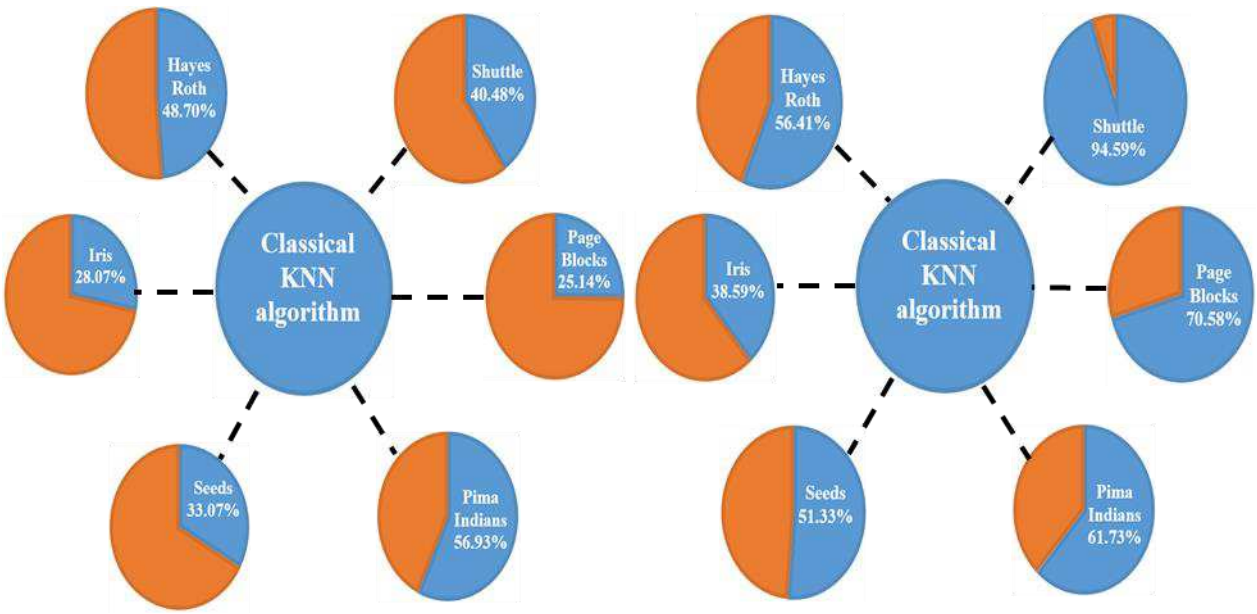

Figure 6. Comparison of classification time improvement rates of the two algorithms

3.3. Improved KNN Algorithm $KNN^{TS\text{-}PK+}$ Based on Clustering and Regional Division

As can be seen from the above two parts, KNN^{PK+} algorithm can remove useless samples and reduce the number of training samples, find the center of the spherical region and make the center selected optimal, and effectively improve the classification accuracy. However, the disadvantage is that the radius of the region selects the farthest distance within the region, leading to a lot of overlap between regions, and repeated calculation of many data will increase the amount of calculation. KNN^{TS} algorithm can calculate the appropriate radius required in the classified spherical region, solve the problem of partial data duplication caused by excessive overlap between regions, and effectively improve the classification efficiency of the algorithm without reducing the classification accuracy. However, the disadvantage is that the center of the spherical region is randomly selected, and some valid sample data may be misclipped, thus affecting the accuracy of classification.

Aiming at the above problems, this paper combines the two methods and proposes an improved KNN algorithm $KNN^{TS\text{-}PK+}$ based on clustering and regional division. First, the equal radius spherical region algorithm is used to divide the sample data in the space into several spherical regions, then PK-means++ algorithm is used to find the center of the spherical region, and then tabu search algorithm is used to find the radius of the spherical region to form an appropriate spherical region. Then, the original sample data set is clipped with iterative operation to form a new sample data set suitable for KNN algorithm. Finally, KNN algorithm is used to classify the processed new sample data set, so as to improve the accuracy of classification and the efficiency of classification.

3.3.1. Steps of $KNN^{TS\text{-}PK+}$ Algorithm

Assuming that the original training set is S, the number of samples is N, the number of iterations is i, and the new training set is H. Store the samples in the divided spherical region with equal radius in H, and let i =0. The specific steps of the algorithm are as follows:

Step 1. PK-Means ++ algorithm is used to determine the initial center point of the spherical region.

Step 2. Tabu search algorithm was used to calculate the optimal radius value of the spherical region.

Step 3. The R value obtained in Step 2 and the center point obtained in Step 1 were respectively taken as the radius and center of the spherical region division method with equal radius, and the sample data was divided into several regions, and the training samples in the regions were saved in train[i].

Step 4. repeat Step 1 to Step 1. If $i > n$, stop the cycle and stores the data samples in the adjusted spherical region in H to form the initial classifier.

Step 5. Calculate the distance between the samples to be tested and each spherical region, store the distance in D, and arrange the data in D in ascending order.

Step 6. Determine the spherical regions adjacent to the training samples, and add the training samples in these K regions to the new training set.

Step 7. In H, KNN algorithm is used to classify test samples.

3.3.2. Experimental Results for $KNN^{TS\text{-}PK+}$ Algorithm

In order to prove the superiority of the improved KNN algorithm, 10 experiments were conducted on six sample data sets, and the classification time and classification accuracy of each experiment were recorded. Take the average of these 10 accuracy rates for the final comparative analysis. The experimental results are as follows:

Table 5 The 10 accuracy rates and average results of KNNTS-PK+ algorithm (%)

	1	2	3	4	5	6
Hayes Roth	97.8	97.7	97.5	96.5	97.9	97.0
Iris	97.3	96.8	96.6	97.0	96.1	96.3
Seeds	86.9	87.0	86.9	88.0	87.2	86.4
Pima Indians	89.2	88.6	89.3	89.0	88.8	89.2
Page Blocks	87.5	87.4	87.1	87.3	87.7	87.4
Shuttle	85.0	84.4	84.3	84.9	84.7	84.6

	7	8	9	10	Average
Hayes Roth	96.8	98.1	97.7	98.0	97.5
Iris	96.9	97.1	96.7	96.2	96.7
Seeds	86.7	87.3	87.5	87.1	87.1
Pima Indians	88.6	88.2	89.1	89.0	88.9
Page Blocks	87.3	87.2	86.9	87.2	87.3
Shuttle	85.0	84.8	84.9	84.4	84.7

As can be seen from Table 5, $KNN^{TS\text{-}PK+}$ algorithm has a high accuracy in classification calculation. Although the accuracy varies from one experiment to another in the ten experiments, the overall fluctuation range is not large, that is, the classification accuracy is relatively stable.

Table 6 Comparison of classification accuracy (%)

	Classical KNN algorithm	$KNN^{TS\text{-}PK+}$ algorithm
Hayes Roth	93.0	97.5
Iris	94.1	96.7
Seeds	85.6	87.1
Pima Indians	83.7	88.9
Pageblocks	85.7	87.3
Shuttle	83.6	84.7

Table 7 Classification time comparison of the two algorithms

	Classical KNN algorithm	The proportion of $KNN^{TS\text{-}PK+}$ algorithm
Hayes Roth	1	53.85%
Iris	1	70.18%
Seeds	1	74.14%
Pima Indians	1	56.09%
Pageblocks	1	57.59%
Shuttle	1	5.88%

Table 6 lists the comparison of classification accuracy between classical KNN algorithm and $KNN^{TS\text{-}PK+}$ algorithm, and Table 7 shows the classification time proportion of $KNN^{TS\text{-}PK+}$ algorithm and classical KNN algorithm. As can be seen from Table 6, compared with the classical KNN algorithm, the classification accuracy of $KNN^{TS\text{-}PK+}$ algorithm is significantly improved. In the case of a small number of data samples, the classification accuracy of $KNN^{TS\text{-}PK+}$ algorithm is relatively high, while on Pageblocks and Shuttle data set with a large number of data samples, the classification accuracy of $KNN^{TS\text{-}PK+}$ algorithm is higher than that of classical KNN algorithm, but the scale of the improvement is less obvious. It can be seen from Table 7 that the classification time of $KNN^{TS\text{-}PK+}$ algorithm is about more than half of that of the classical KNN algorithm when the data samples are relatively small, but only about one tenth of it when the data samples are relatively large.

The classical KNN algorithm does not discriminate the data to be classified before classification calculation, but calculates all the data, which leads to excessively long classification time. In this way, the received interference data directly leads to the decrease of classification accuracy. $KNN^{TS\text{-}PK+}$ divides and cuts out all the data before calculation, which can effectively remove invalid data.This makes it more likely to select and retain valid data suitable for calculation, so classification accuracy and efficiency are greatly improved.

4. Experimental Results And Analysis

In this paper, the experimental results of the classical KNN algorithm,the KNN^{PK+} algorithm, the KNN^{TS} algorithm, and the KNN^{TS-PK+} algorithm are collated and compared. The basic information of these eight data sets is shown in Table 8 below. Classification accuracy and classification efficiency are still taken as measurement standards. Figure 7 and Figure 8 can more intuitively show the comparison effect of classification accuracy and classification time of these four algorithms.

Table 8 Information of eight data sets.

Data set	The total number of samples	Number of attributes	Number of categories
Hayes Roth	133	6	3
Iris	150	5	3
Seeds	210	8	3
Pima Indians	769	9	2
Page Blocks	5473	11	5
Nursery	12960	8	3
Census Income	48842	14	2
Shuttle	58000	10	7

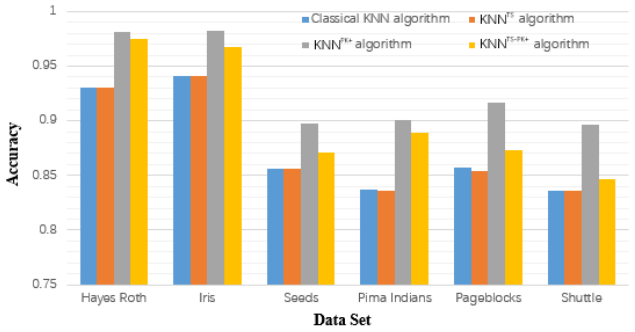

Figure 7. Compares the classification accuracy of the four algorithms for different data sets

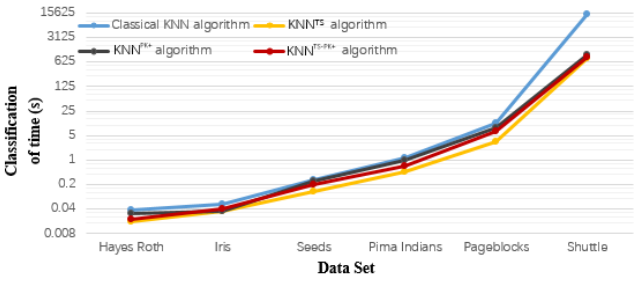

Figure 8. Time log coordinate comparison diagram of classification of different data sets by four algorithms

Figure 7 compares the classification accuracy of the novel improved algorithms and the classical KNN algorithm on the same data set. In the previous experimental results, it can be seen that the classification accuracy of KNN^{PK+} algorithm is higher than that of classical KNN algorithm, and the classification accuracy of KNN^{TS} algorithm is basically the same as that of classical KNN algorithm. After combining the two ideas, the KNN^{TS-PK+} algorithm changes slightly. Compared with the classical KNN algorithm and KNN^{TS} algorithm, its classification accuracy has been further improved, but compared with KNN^{PK+} algorithm, its classification accuracy is slightly lower. On the other hand, starting from the comparative results of classification time, the experimental results are consistent with the previous ones. As can be seen from Figure 8, compared with the classical KNN algorithm, the classification time of KNN^{PK+} algorithm is reduced, but not to a great extent. Compared with the classical KNN algorithm, the classification time of KNN^{TS} algorithm is significantly reduced. Compared with the classical KNN algorithm and

KNN^{PK+} algorithm, the combined KNN^{TS-PK+} algorithm has significantly shortened the classification time, but compared with KNN^{TS}, the classification time is slightly longer.

Based on the analysis of the above experimental results, it can be concluded that the classical KNN algorithm does not pre-process the received data, so it cannot accurately process some fuzzy defined data, which affects the classification accuracy and classification time. KNN^{PK+} algorithm makes more accurate processing of data by determining the center used to divide the spherical region, and then obtains new data sets more suitable for classification, which can effectively avoid the misclipping of useful data in the data set, so the classification accuracy is improved. KNN^{TS} algorithm reduces the overlap problem between regions when dividing and processing data, and avoids the problem of distance calculation for multiple times due to the uncertainty of appropriate radius, thus significantly shortening the classification time. After KNN^{TS-PK+} determines the center of spherical region, the number of sample clipping may be less than that of KNN^{TS} algorithm, so the classification time is longer, but the probability of removing useful data is greatly reduced. After determining the radius of the spherical region, the new sample data set may have more fuzzy data than the samples determined by KNN^{PK+} algorithm, so even if the accuracy is not improved greatly, the problem of overlapping spherical regions is greatly reduced. The overall classification accuracy and efficiency of KNN^{TS-PK+} algorithm is higher than that of classical KNN algorithm.

5. Conclusions And Future Work

To solve the problem that KNN classification algorithm does not pre-process data samples, which leads to a long classification time and a decrease in classification accuracy, two improved algorithms KNN^{TS} and KNN^{TS-PK+} are proposed in this paper. KNN^{PK+} algorithm can greatly improve classification accuracy and effectively reduce classification time. KNN^{TS} algorithm can greatly reduce the classification time while the classification accuracy is basically unchanged. Although the classification accuracy of KNN^{TS-PK+} algorithm is slightly lower than KNN^{PK+} algorithm, its classification time is effectively higher than KNN^{TS} algorithm. In a word, the improved algorithms proposed in this paper all improve the overall classification effect, and they effectively improve the classification accuracy and efficiency. As a relatively perfect classification method, KNN is required for everything from the recognition of numbers to the recognition of faces. In the future, more research emphasis will be placed on the application of optimization algorithms. In particular, KNN can be used in data prediction and analysis, such as disease prediction.

Declarations: Funding: No Funding
Conflicts of interest/Competing interests: Not Applicable
Availability of data and material: Not Applicable
Code availability: Not Applicable
Authors' contributions: Not Applicable

References

[1] Cover, T; Hart, P. “Nearest neighbor Pattern Classification,” IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21-27, 1953.
[2] Wang, Z. H.; Liu, S. T.; Luo, Q. “KNN classification algorithm based on improved k-modes clustering algorithm,” Computer engineering and design, vol. 40, no. 8, pp. 2228-2234, 2019.
[3] Wang, Y. F.; Hao, W. J.; Fan, Z. J. “Improved KNN algorithm based on clustering and density clipping,” Journal of Qingdao university (natural science edition), vol. 30, no. 2, pp. 62-68, 2017.
[4] Saetern, K.; Eiamkanitchat, N. “An Ensemble K-nearest Neighbor with neuro-fuzzy Method for Classification,” Advances in Intelligent Systems & Computing, vol. 265, pp. 43-51, 2014.
[5] Tian, L. “Research on KNN Text Classification Algorithm,” M.S. thesis, Xi'an University of Technology. , Xi’an, China, 2016.
[6] Huang, X. Y. “An Improved KNN algorithm and its application in real-time car-sharing prediction,” M.S. thesis, Dalian University of Technology. , Daian, China, 2018.
[7] Chen, X. D. “Analysis and Research of common clustering algorithm in Data Mining,” Digital Technology and Application, vol. 2017, no. 4, pp. 151-152 , 2017.
[8] Jiang, L.; Xue, S. L. “K-means Algorithm for Optimizing initial clustering center and Determining K value,” Computer and Digital Engineering, vol. 46, no. 1, pp. 21-24, 2008.
[9] Wang, H. Y.; Cui, W. C.; Xu, P. D.; Li, C. “An optimized k-means ++ algorithm guided by local probability,” Journal of Jilin university (science edition), vol. 57 no. 6, pp. 1431-1436, 2019.
[10] Hu, Y. “Research on KNN Text Fast Classification Algorithm based on Regional Division,” M.S. thesis, Shandong University. , Jinan, China, 2012.
[11] Wang, Y.; Tai, Y. H. “Research on the location of beer distributors based on tabu search algorithm,” Logistics technology, vol. 42, no. 11, pp. 13-16, 2019.
[12] Glover, F. “Artificial Intelligence, Heuristic Frameworks and Tabu Search,” Managerial and Decision Economics, vol. 11, no. 5, pp. 365-375, 1990.
[13] Manikandan, S & Chinnadurai, M 2019, ‘Intelligent and Deep Learning Approach OT Measure E-Learning Content in Online Distance Education’, The Online Journal of Distance Education and e-Learning, vol.7, issue 3, July 2019, ISSN: 2147-6454
[14] Garcia, F.; Guijarro, F.; Oliver, J. “Index Tracking Optimization with cardinality constraints: a performance comparison of genetic algorithms and Tabu Search heuristics,” Neural Computing and Applications, vol. 30, no. 8, 2018.
[15] Jin, Q. M.; Li, F. F. “Application of BFD mixed tabu search in one-dimensional packing problem,” Qinghai traffic science and technology, vol. 32, no. 1, pp. 34-38, 2020.
[16] Manikandan S, Chinnadurai M, Thiruvenkatasuresh M.P, Sivakumar M. (2020). "Prediction of Human Motion Detection in Video Surveillance Environment Using Tensor Flow", International Journal of Advanced Science and Technology, 29(05), 2791 - 2798

[17] Hu, J. W. “Improved KNN classification Algorithm based on Regional Division,” M.S. thesis, Qingdao University. , Qingdao, China, 2016.

Figures

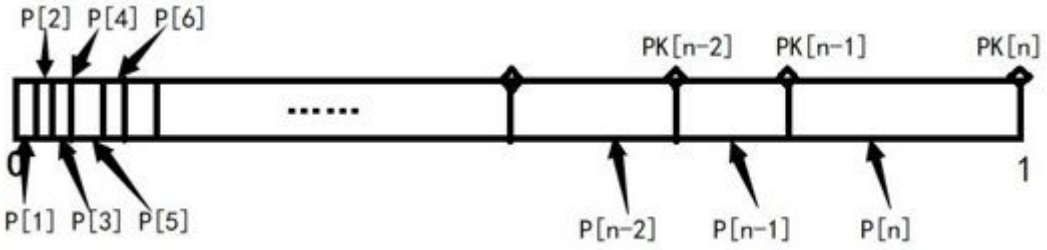

Figure 1

Schematic of array P and array PK

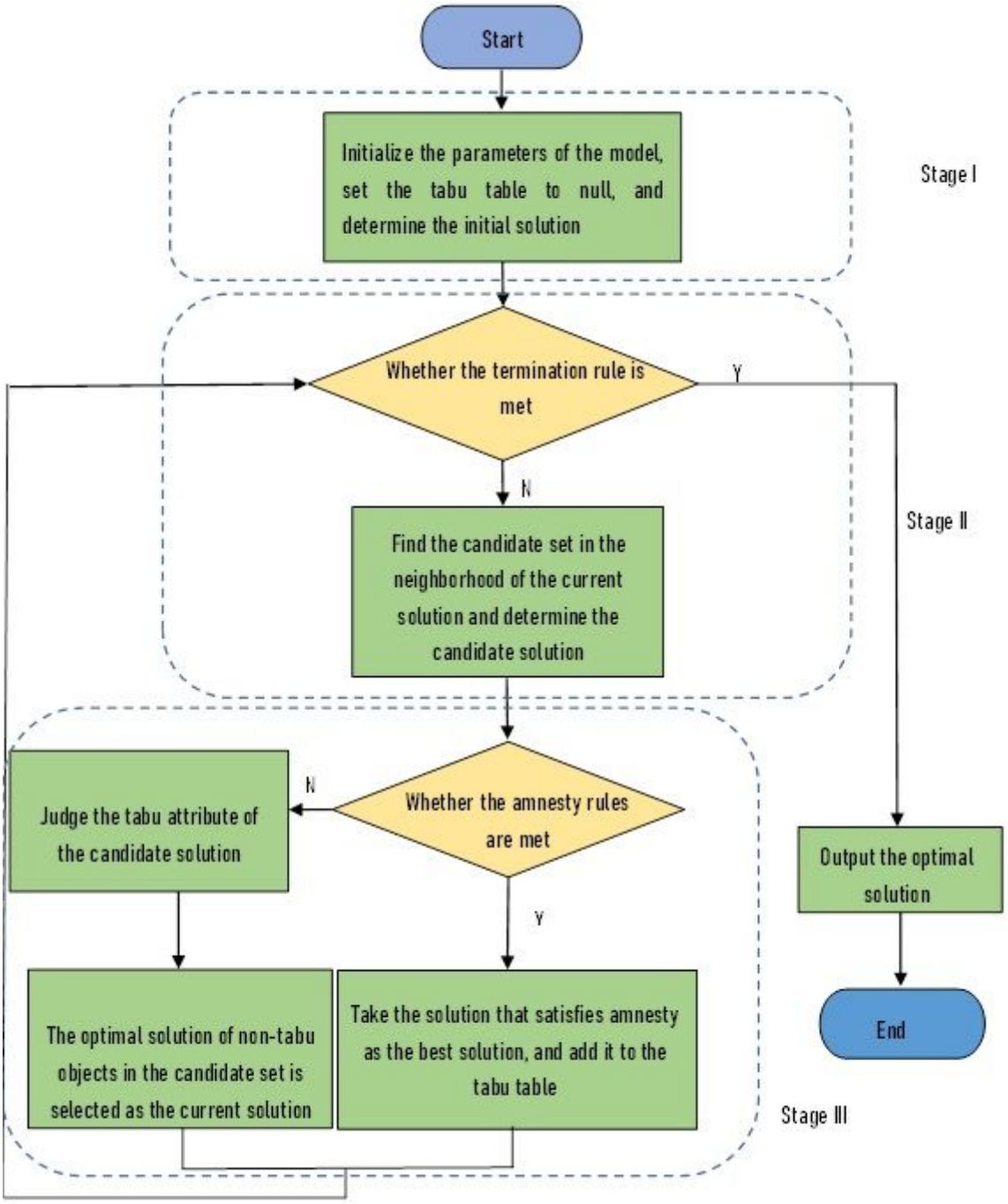

Figure 2

Flow chart of tabu search algorithm

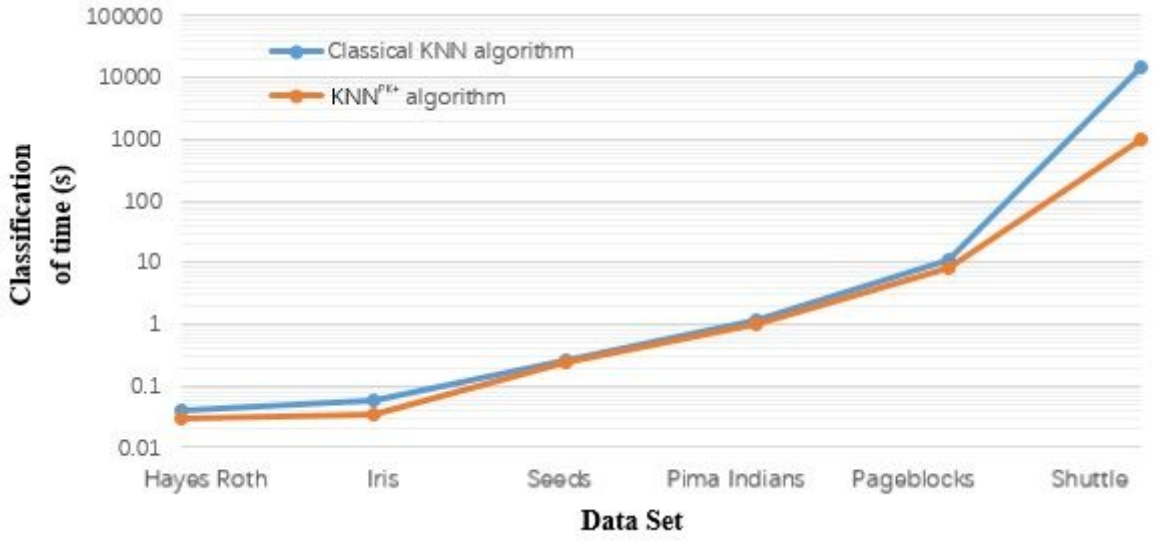

Figure 3

Classification time comparison of the two algorithms

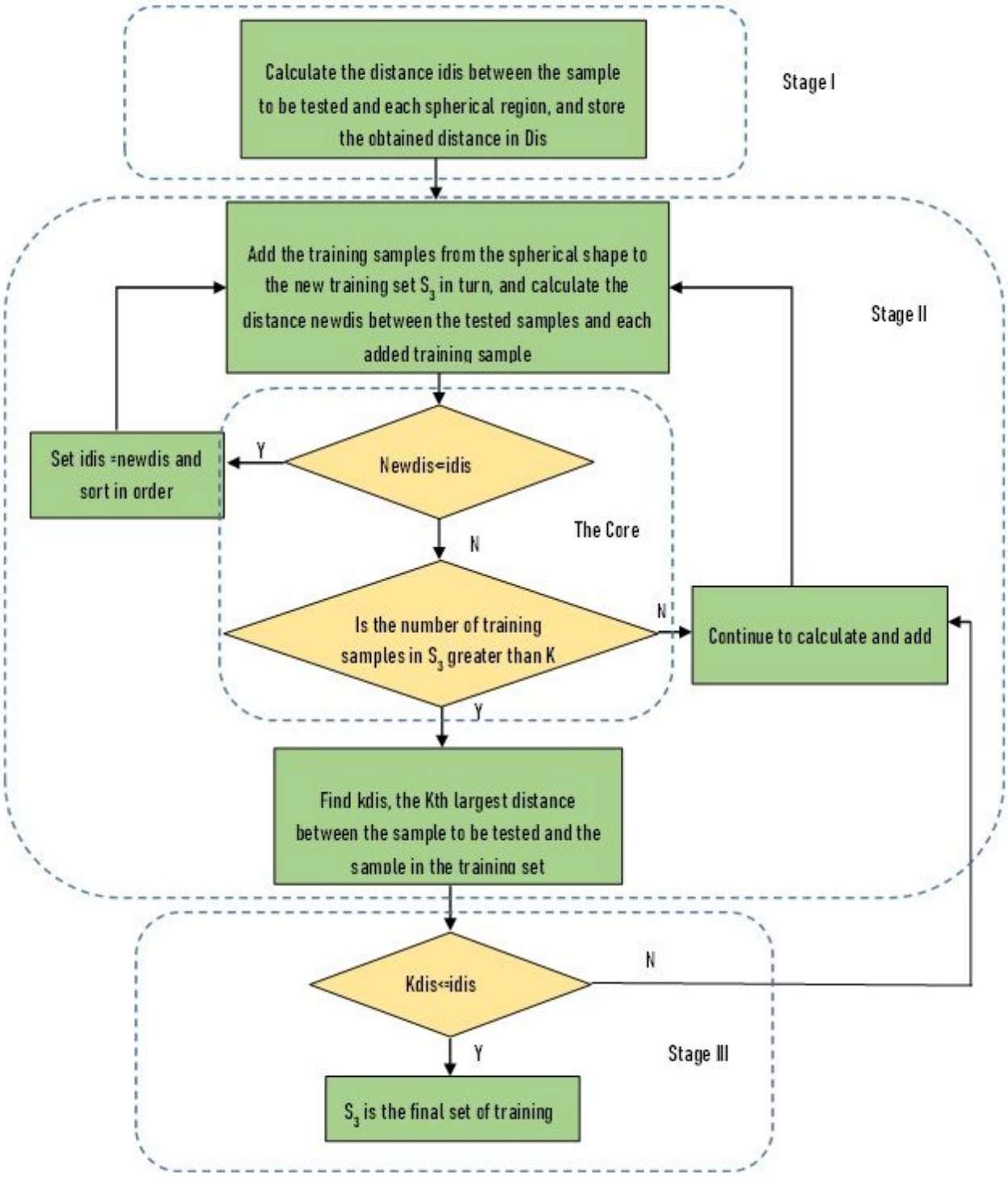

Figure 4

Flow chart of identify new training sets

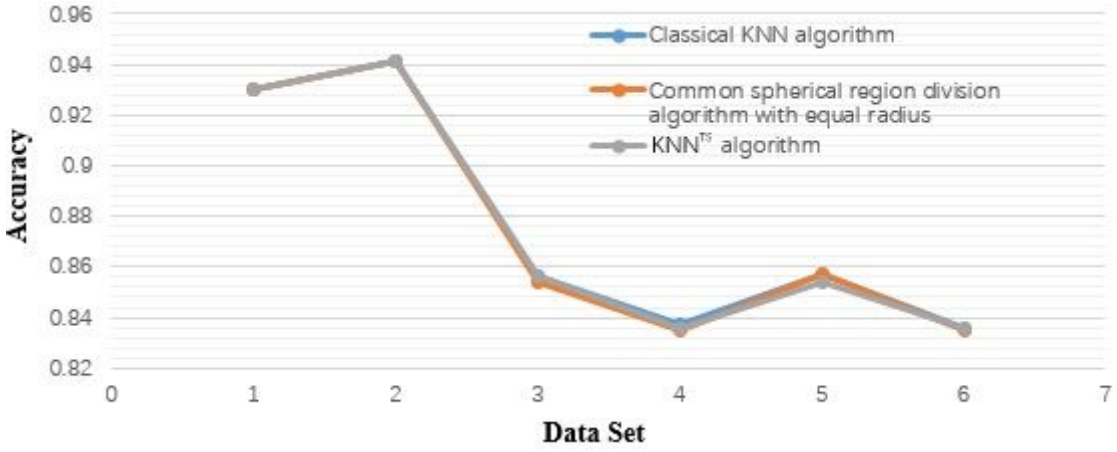

Figure 5

Classification accuracy comparison of the three algorithms

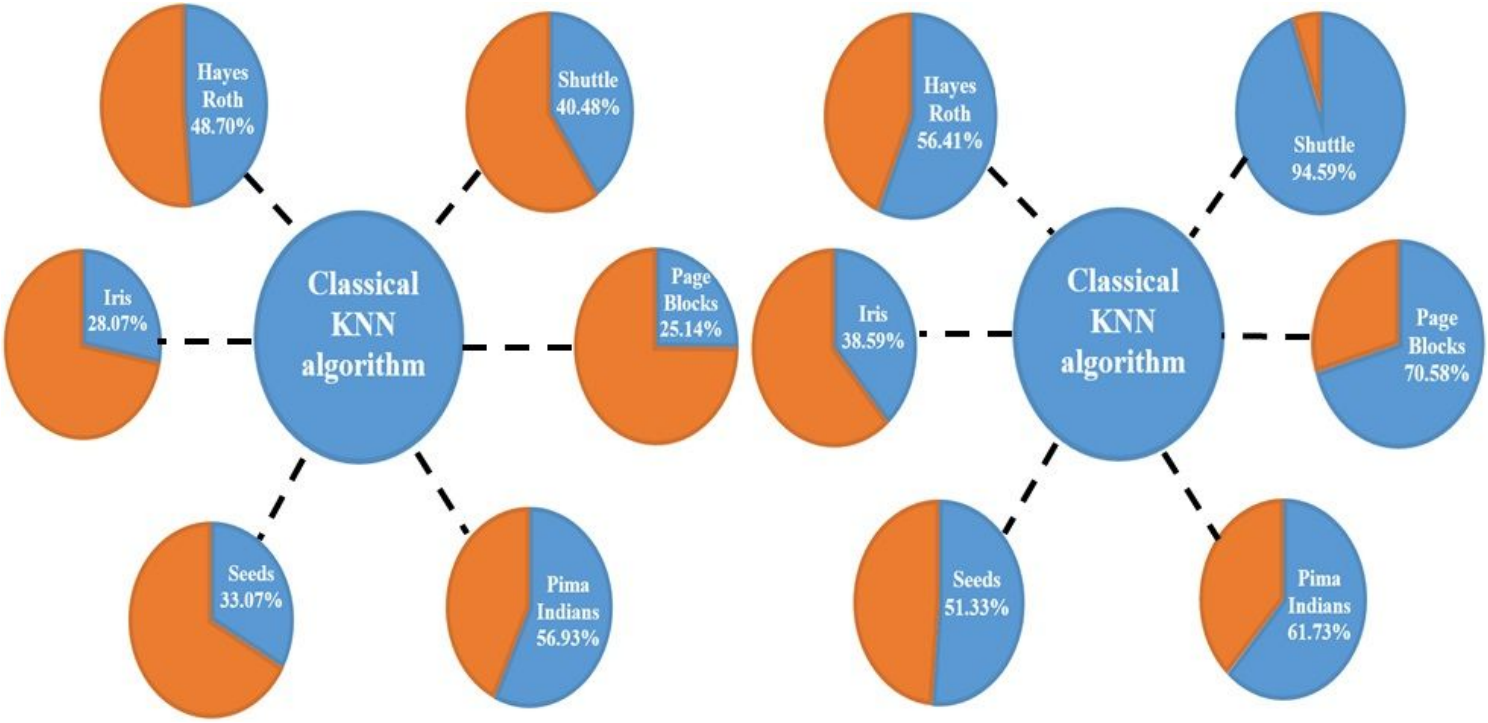

Figure 6

Comparison of classification time improvement rates of the two algorithms

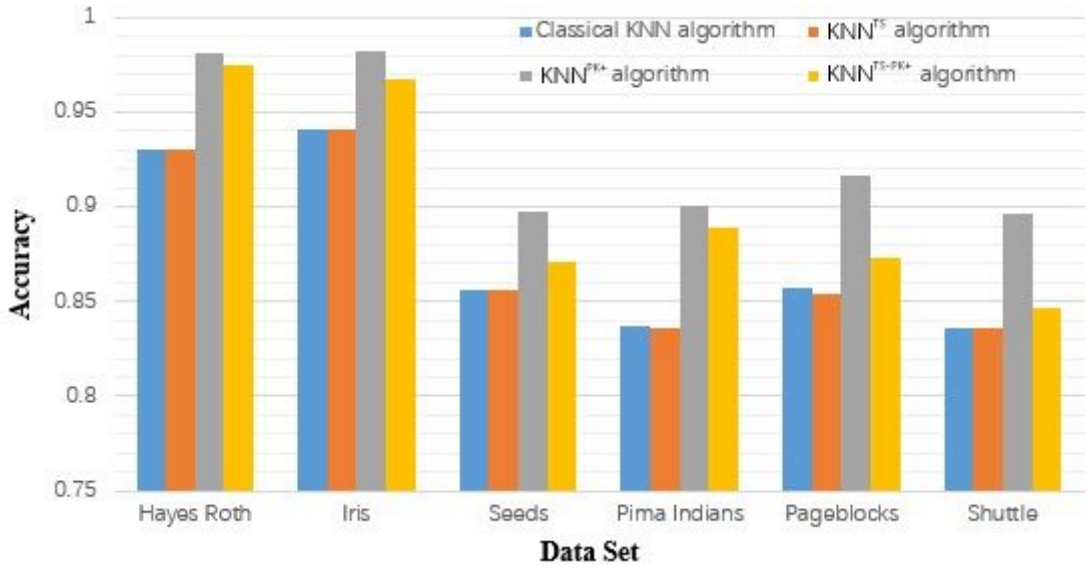

Figure 7

Compares the classification accuracy of the four algorithms for different data sets

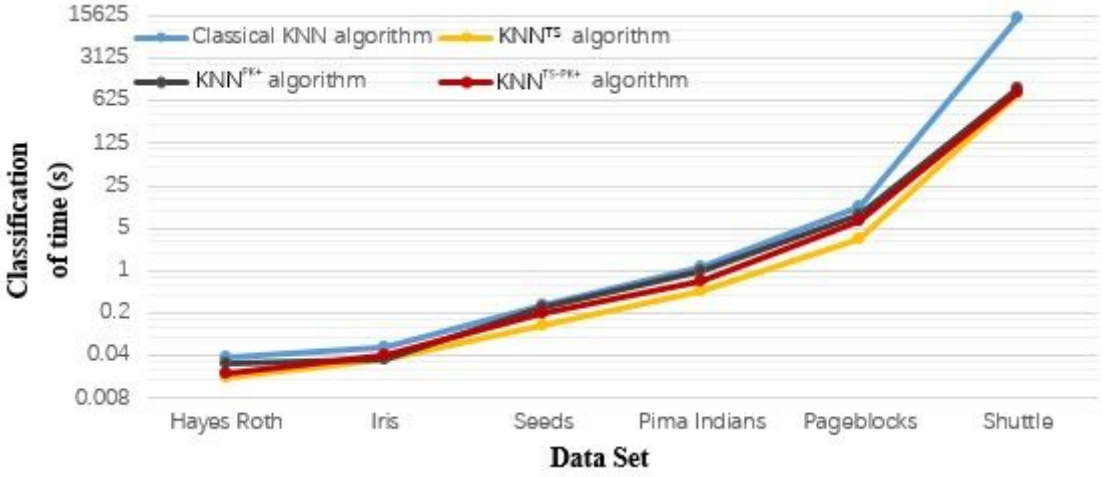

Figure 8

Time log coordinate comparison diagram of classification of different data sets by four algorithms