Peak density algorithm based on KD-tree optimization

Xueyu Huang1*, Chi Xiang1
1 School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China
*Corresponding author’s e-mail: 6120190561@mail.jxust.edu.cn

Abstract: In view of the high time complexity of the density peak algorithm, it is necessary to manually confirm the clustering center according to the decision graph. A density peaking algorithm KT-DPC based on kd-tree optimization is proposed. The algorithm defines the local density $\rho$ through K-nearest neighbor, and uses kd-tree to accelerate the local density $\rho$ and distance $\delta$. In addition, in the confirmation stage of clustering center, a clustering center confirmation strategy (C2BD, clustering center confirmation based on difference) is proposed. The difference between adjacent $\gamma$ is calculated by arranging $\gamma^*=\rho^*\delta$ in ascending order, and the boundary between clustering center and non-clustering center is found according to the change of difference. This method automatically confirms the clustering center of the algorithm, avoiding the problems of strong subjectivity and insufficient clustering accuracy caused by manually confirming the clustering center. Experiments on multiple UCI public data sets show that the running time and clustering accuracy of the KT-DPC algorithm under low-dimensional data are better than traditional DPC algorithms, KNN-DPC and other improved algorithms.

1. Introduction
Clustering algorithm [1,2] is an important part of data mining [3,4,5]. The clustering algorithm divides the data with high similarity into one cluster through the similarity between data, and divides the data with low similarity into different clusters. In this way, clustering Class algorithms can mine useful information from a large amount of data, which makes clustering algorithms widely used in pattern recognition [6], image segmentation [7], machine vision [8], and information retrieval [9].

According to the principle, clustering algorithms can be divided into partition-based clustering [10], grid-based clustering [11], density-based clustering [12], hierarchical clustering [13], etc., but each clustering algorithm may have multiple classifications. For example, the density peak algorithm is a density-based clustering algorithm, but because it still relies on the selection of the cluster center, the algorithm can also be divided into partition-based clustering. These clustering algorithms have their own advantages and disadvantages. Among them, the partition-based clustering mostly depends on the selection of the initial cluster center. Randomly selected or artificially selected initial clustering centers will cause sensitive initial clustering centers and unstable clustering results. Density-based clustering algorithms often need to calculate sample density, resulting in extremely high time complexity and difficulty clustering under large-scale data. The density peak algorithm [14] combines these two disadvantages. Although the density peak algorithm avoids the shortcomings of selecting cluster centers randomly in traditional partition clustering by selecting cluster centers through decision graphs, the method of selecting cluster centers through the decision graph still requires manual confirmation. At the same time, the calculation of local density and distance in the density peak algorithm depends on the cutoff distance $d_c$. To solve these problems, Xie Juanying [15] et al. proposed to define the local density of
samples with the idea of K-nearest neighbors, search and discover the peak density, and take the peak point as the initial clustering center. Then, they proposed the allocation strategy based on K-nearest neighbors, and distributed the non-clustering center samples to the clustering center. The experiments show that the algorithm can quickly identify the data sets of various shapes and sizes, and has good robustness to noise data, while avoiding the influence of truncation distance. Liu Juan et al. proposed a structure based on natural reverse nearest neighbors, which uses natural reverse nearest neighbors to define local density, and then selects the initial cluster center by combining representative points with density. Experiments show that the algorithm has better clustering effect and accuracy, and at the same time has better results in processing complex flow data. Jiang et al. proposed to incorporate the idea of K-nearest neighbors into the distance calculation and allocation strategy. Experiments show that the algorithm has a better effect in processing spiral data clusters. Tang Xinyao et al. proposed to define local density with the idea of natural nearest neighbor, in order to avoid the new artificially defined parameters generated by the definition of local density with K nearest neighbor. Although these methods use the nearest neighbor idea to define the local density or distance of the sample, which avoids the influence of the truncation distance and improves the processing ability of the algorithm to the data set with complex shapes, the algorithm needs to traverse all the samples in all the data sets when finding the nearest neighbor of the sample, and the time complexity is n. At the same time, there are many subjective factors in the method of artificially selecting the initial clustering center through the decision diagram.

To solve the above problems, this paper proposes a density peak algorithm KT-DPC based on kd-tree optimization. The main tasks are: (1) By constructing the kd-tree to accelerate the calculation of the local density and distance defined by the K nearest neighbor in the density peak algorithm, thereby reducing the running time of this stage and improving the efficiency of the algorithm; (2) Propose an initial cluster center confirmation strategy C2BD, by sorting a and calculating the difference between adjacent y after sorting, the initial cluster center is obtained according to the change of the difference. The experimental results show that the KT-DPC algorithm can quickly identify the accurate initial clustering center and the number of clusters, and the running time of the DPC-KNN algorithm is also reduced compared with the traditional DPC algorithm and the improved KNN-DPC algorithm.

2. Introduction to related concepts

2.1. Density Peak Algorithm (DPC)

The density peak algorithm is a clustering algorithm proposed by Alex R et al. in 2014. The algorithm idea is: the local density of the clustering center is greater than the local density of the data around the cluster, and the distance between the object with larger local density is farther. From this, two variables $\rho$ and $\delta$ are obtained, $\rho$ is the local density of sample $i$, and the formula is as follows:

$$\rho_i = \sum_{j} \chi(d_{ij} - d_{c})$$

Among them, $d_{c}$ is the truncation distance, which is manually confirmed. When the variable $x$ in $\chi(x)$ is less than or equal to 0, it is taken as 1, and when it is greater than 0, it is taken as 0; $d_{ij}$ is the Euclidean distance between sample $i$ and sample $j$.

For a sample with a local density greater than sample $i$ and closest to sample $i$, the distance between it and sample $i$ is $\delta$, the formula is as follows:

$$\delta_i = \min_{j \neq i, \rho_j > \rho_i} (d_{ij})$$

$$\delta_i = \max_{j} (d_{ij})$$

$$\min(d_{ij})$$ represents the distance between the sample $i$ and the nearest sample whose local density is greater than it, and the maximum local density is $\max(d_{ij})$.

At the same time, in order to reduce the number of occurrences of the same local density, the local density $\rho_i$ of the DPC algorithm is calculated by using a Gaussian kernel function, as follows:

$$\rho_i = \sum_{j, j \neq i} \exp \left(-\frac{(d_{ij})^2}{d_{c}^2}\right)$$

$$\exp$$
When the local density $\rho_i$ of sample $i$ is the largest in DPC algorithm, the corresponding distance $\delta_i$ will be much larger than that of other samples, and the probability of sample points with large $\delta_i$ becoming cluster centers will be high. Therefore, the DPC algorithm selects the clustering center in the way of two-dimensional decision diagram, and establishes the decision diagram with the local density $\rho$ as the abscissa and the distance $\delta$ as the ordinate. As shown in Figure 1, the frame points in the diagram are the clustering centers.

![Figure 1. Decision-making figure.](image)

### 2.2. Kd-tree
Kd-tree is a lightweight binary tree. The principle is to use a hyperplane perpendicular to the coordinate axis to divide the k-dimensional space continuously, and finally get many super rectangular regions.

Kd-tree is first a root node containing all k-dimensional spaces, and then two left and right nodes are obtained through the hyperplane perpendicular to the coordinate axis. At the same time, the k-dimensional space is divided into two regions, and the above process is repeated until left and right nodes no longer have samples, and the final kd-tree is obtained.

In the entire kd-tree, each node is represented by $T_r$, and the left and right child nodes are represented by $T_{rl}$ and $T_{rr}$. There are:

$$T_r = T_{rl} \cup T_{rr}$$  \hspace{1cm} (4)

$$T_{rl} \cap T_{rr} = \emptyset$$  \hspace{1cm} (5)

When both $T_{rl}$ and $T_{rr}$ are empty, node $T_r$ is a leaf node.

### 3. KT-DPC algorithm

#### 3.1. Algorithm idea
KT-DPC algorithm is mainly divided into three parts: accelerating the neighbor search of samples, obtaining the initial clustering center, and obtaining the clustering results. (1) In the stage of accelerating the nearest neighbor search of samples: a DPC algorithm based on kd-tree is proposed to improve the local density of k nearest neighbor definition, and accelerate the speed of the algorithm in searching the nearest neighbor, so as to solve the problem of high time complexity in calculating local density and distance. (2) Establishing the decision diagram to obtain the initial cluster center stage: C2BD strategy is proposed to sort the product $\gamma$ of local density $\rho$ and distance $\delta$, and the difference between adjacent $\gamma$ is calculated. The initial cluster center and the number of clusters are confirmed by the variance of the difference. (3) Clustering stage: read the initial clustering center, complete the clustering, get the final clustering results.

#### 3.2. Acceleration of density and distance
The traditional DPC algorithm calculates the local density and distance by the truncation distance $d_c$, and the truncation distance is usually confirmed manually. According to experience, manual confirmation of $d_c$ is based on making samples within an average range of 1% ~ 2% of the total...
sample. However, this method has poor robustness in the face of different data sets. Later, many researchers introduced the idea of knn into DPC algorithm, and calculated the local density $\rho$ by the k nearest neighbor of the sample. The formula is as follows:

$$\rho_i = \sum_{j \in knn(i)} \exp(-d_{ij})$$

(6)

Among them, $d_{ij}$ is the Euclidean distance between samples, $knn(i)$ is the set of k nearest neighbors of sample $i$.

This method can effectively avoid the shortcoming that the truncation distance $d_c$ needs to be artificially confirmed. However, this method needs to calculate the Euclidean distance between all samples when searching the nearest neighbor of the sample, and the time complexity is high. In this regard, this paper proposes to accelerate this process by constructing kd-tree.

For a sample $i$, to query its k-nearest neighbors, the root node of the kd-tree first checks the nodes according to the depth-first principle. In order to facilitate the need to define a temporary stack to store the neighbor points. Defined as follows:

(MARK) Set an empty stack $S$, putting the nodes traversed sequentially into stack $S$. (1) The distance between the sample point $i$ and the current node $D_T$ is greater than the distance to all samples in the stack $D_S$, return to the stack $S$, traverse another child node or output k samples from the stack as the nearest neighbor of the sample $i$; (2) If the distance $D_T$ from sample $i$ to the current node is less than the distance $D_S$ from the sample in the stack, the left and right sub-nodes of the node are recursively called, and the sub-nodes with smaller $D_T$ are updated to the stack $S$; (3) If the query node is a leaf node, the distance between all samples in the node and the sample $i$ is calculated, and the samples less than $D_S$ are put into the stack from near to far.

The first K samples of output stack $S$ are the nearest neighbor of sample $i$.

After obtaining the k nearest neighbor of the sample, the local density $\rho$ of the sample is calculated according to formula 6. For the distance $\delta$ of sample $i$, according to the original definition, it is the nearest distance between the sample and the sample whose density is greater than it. According to this definition, if the local density of the sample is the largest in the sample itself and its k nearest neighbors, then the sample distance needs to be confirmed from the samples other than the k nearest neighbors. If not, the samples closest to it and whose density is larger than it must be in the k nearest neighbors of the sample.

3.3. C2BD strategy

The initial cluster centers selected by the density peak algorithm are usually manually selected by the decision diagram, as shown in Fig. 1. However, not every decision diagram will have a completely clear sample for selection. Some decision diagrams have a large local density $\rho$ but a small distance $\delta$, and vice versa. DPC algorithm requires a large density and a long distance to the initial clustering center, so a new quantity $\gamma$ is proposed, and the formula is as follows:

$$\gamma = \rho \times \delta$$

(7)
Taking \( \gamma \) as the ordinate of the new decision diagram, and the sequence number of the corresponding samples after descending order of \( \gamma \) as the ordinate, the decision diagram is constructed. As shown in Fig. 2, the circle samples are taken as the initial clustering center. Compared with the decision diagram of the original algorithm, the new decision diagram is undoubtedly more intuitive and clear. However, this method still has the shortcomings of manual selection. In this regard, according to the characteristics of the new decision diagram and \( \gamma \) itself, for example, the \( \gamma \) value of the sample selected as the initial cluster center is much larger than that of the non-cluster center. In addition, there is a boundary between the \( \gamma \) of the cluster center sample and the \( \gamma \) of the non-cluster center shown in d in Fig. 2 ( d in Fig. 2 ). This paper proposes a mean-based C2BD policy search to confirm this bound, which is defined as follows:

Assuming that all samples are arranged in descending order to obtain \( \gamma = \{ \gamma_0, \gamma_1, ..., \gamma_n \} \), the difference \( d \) between adjacent \( \gamma \) and the mean value \( \bar{d} \) of the difference are calculated, and \( \gamma \) is traversed in reverse. If \( d > \bar{d} \), the boundary between the corresponding two samples is the boundary between the clustering center and the non-clustering center. The formula is as follows:

\[
d = \gamma_m - \gamma_n, n - m = 1
\]

\[
\bar{d} = \frac{\sum_{m=0}^{n-1} \gamma_m - \gamma_n}{m + 1}
\]

(8)

Where \( m, n \) is the subscript of \( \gamma \). In the ordered \( \gamma \) array, the difference \( d \) between \( \gamma \) of non-clustering centers is mostly smaller than that of clustering centers, especially the sample \( \gamma \) on both sides of the dividing line will obviously have a jump. In this paper, greater than \( \bar{d} \) is used as the measurement standard. When \( d > \bar{d} \), the dividing line is found, and \( d = \gamma_{k-1} - \gamma_k \), \( k \) is the number of clusters.

3.4. DPC algorithm steps

Step 1 establishes kd-tree and obtains k nearest neighbors of samples.

Step 2 calculates the local density \( \rho \) and distance \( \delta \) of the sample according to Formulas 6 and 2, and calculates \( \gamma = \rho \times \delta \)

Step 3 obtains the initial cluster center point set and cluster number k.

Step 4 : The non-cluster center samples are assigned according to the distribution method in literature [15], and the final clustering results are obtained.

3.5. lgorithm analysis

The time complexity of KT-DPC algorithm is mainly affected by three parts: the establishment of kd-tree and the nearest neighbor query, the acquisition and allocation of clustering centers.

(1) This paper mainly involves the establishment of kd-tree and the nearest neighbor query. The time complexity of the establishment of kd-tree is \( o(\log(2n)) \), and the time complexity of the nearest neighbor query is \( o(n^{1-1/k} + m) \), where \( k \) is the dimension of the data set, \( m \) is the number of nearest neighbors to query, so the total complexity of the step is \( o(\log(2n)) \).

(2) In the initial clustering center acquisition stage, the first is the calculation of local density. The calculation of local density is based on the k nearest neighbor, so the time complexity of the algorithm is \( o(n \times m) \), and the calculation of distance also depends on the k nearest neighbor. If the nearest sample is in k nearest neighbors, then the time complexity is \( o(n \times m) \), if not then \( o((n - l) \times n) \), where \( l \) is the sample number of distance calculated by the nearest neighbor point. Because the sample outside the density peak point in the algorithm can determine the distance through the nearest neighbor point, this part of the sample is far more than the density peak point, namely \( l \rightarrow n \), so the time complexity of calculating density and distance is \( o(2n \times m + (n - l)n) \). Because \( l \) is large and \( m \) is small, the time complexity is \( o(2n) \). In addition, the C2BD strategy needs to calculate the sum of the difference between adjacent \( \gamma \) and the average \( \bar{d} \) of the difference, and the total time complexity is \( o(n) \) when
the time complexity is $o(n)$.

(3) In the allocation method, the first step is the breadth-first search traversal of similar graphs, and the time complexity is $o(n_t(|O| + 1))$, where $n_t$ is the number of non-offset points, and $|O|$ is the length of the cluster center set; Similarly, the time complexity of the second step is $o(n_2(|O| + 1))$, where $n_2 = n - n'_t$, $n'_t$ is the number of samples that have been allocated. Since the number of non-outliers is much larger than that of outliers, that is, $n_t \rightarrow n$, and the number of cluster centers is small, the time complexity of this part of the algorithm is $o(n)$.

In summary, the time complexity of KT-DPC algorithm is $o(log(2n) + n)$, which is better than DPC algorithm and KNN-DPC algorithm.

4. Analysis of experimental results

4.1. Experimental data set

In order to verify the performance of KT-DPC, multiple real data sets, artificial data sets and scale data sets [19] were used to compare the accuracy of the algorithm, the number of clusters identified and the running time. The comparison algorithms include DPC algorithm, DPC-KNN algorithm and KNN-DPC algorithm.

| Table 1. Experimental data set information. |
|---------------------------------------------|
| Records | Attributes | label |
| Iris | 150 | 4 | 3 |
| Wine | 178 | 13 | 3 |
| WDBC | 569 | 30 | 2 |

| Table 2. Artificial data set. |
|--------------------------------|
| Records | Attributes | label |
| Flame | 240 | 2 | 2 |
| R15 | 600 | 2 | 15 |
| D31 | 3100 | 2 | 31 |

| Table 3. Scale data set. |
|--------------------------|
| Records | Attributes | label |
| User000 | 176182 | 3 | unknown |
| User002 | 248217 | 3 | unknown |
| User000-002 | 528382 | 3 | unknown |
| User000-030 | 764962 | 3 | unknown |
| User000-050 | 11113351 | 3 | unknown |

Among them, the five data sets such as User are selected from the Geolife project of Microsoft Asia. [20]

4.2. Clustering accuracy analysis

Firstly, the accuracy of KT-DPC algorithm, DPC algorithm, KNN-DPC algorithm and DPC-KNN algorithm are compared and analyzed in three real datasets as shown in Figure 3, where $d_c = 2\%$ in DPC algorithm and $k = \text{in k nearest neighbor.}$
As can be seen from Figure 3, in the low-dimensional data set Iris, the accuracy of each algorithm is similar. With the increase of the dimension of the data set and the increase of the amount of data, the accuracy of DPC algorithm is significantly lower than that of the other three algorithms. Especially in the Wine data set, the accuracy of DPC algorithm is 9%, 4% and 10% lower than that of KNN-DPC, DPC-KNN and KT-DPC, respectively. Obviously, the clustering effect of DPC algorithm in high-dimensional data is poor, and selecting a suitable $d_c$ is also an important factor affecting the clustering results.

4.3. Obtaining cluster centers

In order to test whether the algorithm can accurately identify the cluster centers and the number of clusters in the data set with many clusters, this paper takes the Flame, R15 and D31 data sets in the artificial data set as the test data set, and the form of R/W represents the ability to correctly identify the cluster centers. Among them, R refers to the number of cluster centers identified by the algorithm, and W refers to the number of cluster centers in the data set. As shown in Table 4:

| Algorithms | Flame | R15 | D31 |
|------------|-------|-----|-----|
| DPC        | 2/2   | 15/15 | 28/31 |
| KNN-DPC    | 2/2   | 15/15 | 30/31 |
| DPC-KNN    | 2/2   | 15/15 | 30/31 |
| KT-DPC     | 2/2   | 15/15 | 31/31 |

It can be seen from Table 4 that several algorithms can correctly identify the number of cluster centers under the Flame and R15 data sets, but the number of clusters reaches 31, and the D31 data set with a relatively concentrated data set does not recognize the DPC algorithm. Out of 3 clustering centers, KNN-DPC algorithm and DPC-KNN algorithm also missed one. This is because these 3 algorithms rely on human eyes to select cluster centers in the decision diagram, and there are more clusters in the data set, and when the local density and distance gap are not large, people will wrongly select cluster centers. The algorithm in this paper finds the dividing line by calculating the difference of $\gamma$, so that the algorithm can adaptively confirm the clustering center, avoiding mistakes and omissions during manual selection, and improving the algorithm's ability to identify clustering centers.

4.4. Running time comparison

In order to verify that the KT-DPC algorithm accelerates the search of k nearest neighbors through kd-tree and reduces the running time of the algorithm, the KT-DPC algorithm and the DPC algorithm, KNN-DPC algorithm are tested on several large-scale data sets from Microsoft Asia, as shown in Figure 4:
Figure 4. The running time of the three algorithms in each data set.

It can be clearly seen from Fig. 4 that the running time of DPC algorithm and KNN-DPC algorithm on each data set is equivalent. This is because although KNN-DPC algorithm introduces the idea of k-nearest neighbor to redefine the local density and avoid the influence of truncation distance $d_c$, the time complexity of the algorithm in searching the nearest neighbor of the sample is still $O(n^2)$, which is equivalent to that of DPC algorithm. Therefore, although KNN-DPC algorithm avoids the influence of truncation distance $d$ and improves the accuracy of the algorithm, it has not been much improved in the overall running time. KT-DPC algorithm accelerates the search of k-nearest neighbors through kd-tree, which greatly reduces the calculation time of local density and distance of the algorithm. Especially on the largest User000 - 050 data set, the operation time of KT-DPC algorithm is reduced by 33.2 % and 32.2 % compared with DPC algorithm and KNN-DPC algorithm, and the algorithm efficiency is greatly improved.

5. Summary
Introducing the idea of k-nearest neighbor into the density peak algorithm to define a new local density calculation method is a research hotspot in recent years. However, most of the improvements only improve the accuracy of the algorithm to avoid the “domino” phenomenon, and basically have no improvement on the time complexity of the algorithm. In this paper, the kd-tree tree is used to accelerate the calculation of local density and distance, and the time complexity is successfully reduced $O(\log n)$.

At the same time, the C2BD strategy is used to adaptively confirm the clustering center, which avoids the error caused by manual selection and improves the clustering accuracy. However, although these improvements reduce the time complexity, the peak density algorithm is still time-consuming in dealing with large-scale data sets. At the same time, the introduction of the idea of k nearest neighbor also increases the new parameter $k$. How to avoid generating new parameters and further reduce the time complexity is the next research direction.

References
[1] A Pérez-Suárez, JF Martinez-Trinidad, Carrasco-Ochoa J A. A review of conceptual clustering algorithms[J]. Artificial Intelligence Review, 2018.
[2] Zubarolu A, Atalay V. Data Stream Clustering: A Review[J]. Artificial Intelligence Review, 2020(C).
[3] Lakshmi B N, Raghunandhan G H. A conceptual overview of data mining[C]// Innovations in Emerging Technology. IEEE, 2011.
[4] Boudane, Abdelhamid, Jabbour, et al. SAT-Based Data Mining[J]. International Journal of Artificial Intelligence Tools Architectures Languages Algorithms, 2018.
[5] Li L, Wang J, Fei Q. An overview of data mining[C]// Intelligent Control & Automation. IEEE, 2010.
[6] Chen G, Lian W, Hu F, et al. Research on intelligent target recognition method based on pattern recognition and deep learning[C]// Second Target Recognition and Artificial Intelligence Summit Forum. 2020.

[7] Xuan S, Wen J, Li Y, et al. Image segmentation algorithm based on neutrosophic fuzzy clustering with non-local information[J]. IET Image Processing, 2019, 14(3).

[8] Zhang C, Xu X, Fan C, Wang G, et al. Literature Review of Machine Vision in Application Field[C]// 3rd International Conference on Energy Resources and Sustainable Development, ICERSD 2020.

[9] Mahapatra D, Maharana C, Panda S P, et al. A Fuzzy-Cluster based Semantic Information Retrieval System[C]// Fourth International Conference on Computing Methodologies and Communication (ICCMC). 2020.

[10] Velmurugan T. Applications of Partition based Clustering Algorithms: A Survey[C]// 2013 IEEE International Conference on Computational Intelligence and Computing Research. IEEE, 2013.

[11] Lee G H. Grid-based dynamic clustering with grid proximity measure[J]. Intelligent Data Analysis, 2016, 20(4):853-875.

[12] Bhattacharjee P, Mitra P. A survey of density based clustering algorithms[J]. Frontiers of Computer Science (print), 2020, 15(1):151308.

[13] Murtagh F, Contreras P. Algorithms for hierarchical clustering: an overview[J]. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2012, 2(1):86–97.

[14] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[15] Xie Juanyin, Gao Hongchao, Xie Weixin, et al. K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset[J]. Science China: SCIENTIA SINICA Informationis, 2016, 46(02): 258-280.

[16] Liu Wan J, et al. Optimized Density Peak Clustering Algorithm by Natural Reverse Nearest Neighbor [J]. Journal of Frontiers of Computer Science and Technology: 1-12[2021-03-26].http://kns.cnki.net/kcms/detail/11.5602.TP.20200917.1251.002.html.

[17] Tang Xinyao, Zhang Zhengjun, Chu Jie, Yan Tao, et al. Density Peaks Clustering Algorithm Based on Natural Nearest Neighbor[J]. Computer Science: 1-12[2021-01-16].https://kns-cnki-net.webvpn.jxust.edu.cn/kcms/detail/50.1075.TP.20201104.1623.004.html.

[18] Hou W, Li D, Xu C, et al. An Advanced k Nearest Neighbor Classification Algorithm Based on KD-tree[C]// 2018 IEEE International Conference of Safety Produce Informatization (IICSPI). IEEE, 2018.

[19] Lan Hong, Huang Min, et al. Improved FCM algorithm based on KNN optimized density peaks algorithm[J]. Computer Engineering and Applications, 1-13[2021-01-09].

[20] Zheng Y, Zhang L, Xie X, et al. Mining interesting locations and travel sequences from GPS trajectories[J]. International Conference on World Wide Web, 2009: 791-800.