Density Peaks Clustering Algorithm Based on Weighted $k$-Nearest Neighbors and Geodesic Distance

LINA LIU$^{1}$ AND DONGHUA YU$^{2}$

$^{1}$School of Electronic and Information Engineering, Soochow University, Suzhou 215006, China
$^{2}$Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000, China

Corresponding author: Lina Liu (lln@suda.edu.cn)

ABSTRACT Density peaks clustering (DPC) is a density-based clustering algorithm with excellent clustering performance including accuracy, automatically detecting the number of clusters, and identifying center points. However, DPC has some shortcomings to be addressed before it can be widely applied. For example, sensitive predefined parameter is not suitable for manifold datasets, and decision graph easily causes the wrong center points. To address these issues, a new DPC algorithm based on weighted k-nearest neighbors and geodesic distance (DPC-WKNN-GD) is proposed in this article to improve the clustering performance for manifold and non-manifold datasets. The DPC-WKNN-GD introduces the weighted k-nearest neighbors based on Euclidean distance to optimize the local density $\rho$. Inspired by DPC-GD, the DPC-WKNN-GD redefines the distance $\delta$ based on geodesic distance. The experimental results on artificial and real-world datasets, including image datasets, show that the DPC-WKNN-GD outperforms the state-of-the-art comparison algorithms on both manifold and non-manifold datasets. In addition, the DPC-WKNN-GD completely overcomes the DPC-GD defect, in which the decision graph displays misleading center points.

INDEX TERMS Cluster analysis, density peaks clustering, weighted $k$-nearest neighbors, geodesic distance.

I. INTRODUCTION Clustering is the most important method of unsupervised learning for data analysis and interpretation. It is the process of dividing a set of points into non-overlapping subsets. Each subset is a cluster, such that points in the same cluster are similar to one another (intra-similarity) and dissimilar to points in other clusters (inter-dissimilarity) [1]. Countless clustering algorithms have been proposed and are generally categorized into five groups: hierarchical [2], partition-based [3], density-based [4], grid-based [5], and model-based [6] algorithms. Various clustering ensemble algorithms have also been proposed [7]. Clustering is widely used in research and engineering applications, such as data mining, pattern recognition, image segmentation, genetic disease detection, and social networks.

Density-based clustering algorithms have always been popular because they can define clusters with arbitrary shape and handle outliers well. In 2014, density peaks clustering (DPC) was proposed by Rodriguez and Laio [8], resulting in density-based clustering algorithms attracting wider attention and application. The DPC has its basis in the assumptions that cluster center points are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. Based on them, DPC requires neither an iterative process nor more input parameters to identify cluster centers, except for manually determining the center points from the decision graph.

Although the DPC and its improved algorithms have achieved many successful applications [9]–[13], many of its following shortcomings cannot be ignored [14]–[21].

1) The cutoff distance $d_c$ must be pre-specified based on users’ experience and prior knowledge of the dataset. An imperfect parameter $d_c$ may not highlight characteristics for cluster center points and degrade the performance of the DPC. It is also sensitive to the clustering result, especially on small-scale datasets.

2) The DPC assignment strategy assigns remaining points (i.e., non-center points) in descending order of local density to the same cluster as its nearest neighbor of
higher density. This is highly likely to cause cluster label error propagation. Once a point is assigned to an incorrect cluster, other lower local density nearest neighbor points will also be assigned to the incorrect cluster.

3) The cluster center points would be selected manually, and sometimes those on the decision graph may not be obviously separated from the remaining points. It is difficult to define the boundary between the cluster center and non-center points, and will cause the cluster center points to be uncertain, which leads to incorrect clustering.

4) In the DPC clustering result, each cluster has one and only one density peak. Therefore, it is not satisfactory for multi-peak manifold datasets.

Obviously, these shortcomings will reduce the clustering performance and limit its wide application. Therefore, a major motivation of this article is to extend the DPC to manifold datasets to expand the scope of application, and improve the clustering performance on manifold and non-manifold datasets.

Many researchers emphasize the improvement of clustering performance, mainly focusing on improving or overcoming the above first, second, and third shortcomings. Others tend to broaden the scope of application, such as popularizing it to manifold datasets, where the manifold data generally refer to the non-spherical special distribution data. In addition, several accelerated clustering processes make this algorithm efficient. The main work of this article focuses on extending the algorithm to manifold datasets while improving clustering performance on both manifold and non-manifold datasets.

To cluster and analyze manifold data, one of the main tasks is to measure the geometry of the distribution of the data. The manifold structure data is ubiquitous in the real world, e.g., handwritten digit recognition, image segmentation, and web analysis. These multiple low-dimensional manifolds embedded in high-dimensional data always are non-spherical shapes [17]. Many algorithms have been proposed for this. Xu et al. [15] proposed RDPC-DSS, which introduces the density-sensitive similarity instead of the conventional Euclidean distance to obtain the desirable clusters on the manifold datasets. Cheng et al. [22] redefined the graph-based distance between local cores with shared-neighbors-based distance for manifold datasets. Du et al. [23] proposed a distance metric relying on the shortest path, which can measure the geometry of the distribution of the data.

The geodesic distance is a point-to-point distance metric based on the shortest path, and it is often used by manifold learning algorithms to learn the geometric structure of manifolds [24]. Du et al. [17] used the excellent property of the geodesic distance instead of the Euclidean distance directly to optimize the DPC, and proposed density peaks clustering based on geodesic distance (DPC-GD). This algorithm achieves satisfactory clustering results on many manifold datasets. However, the local density of the DPC-GD is susceptible to data distribution, with low accuracy and poor separation. To make matters worse, it is difficult for the DPC-GD’s decision graph to show the correct center points, and it is easy to produce misleading conclusions about the number of clusters. Taking the Aggregation dataset as an example for explanation, Figure 1 shows the DPC-GD’s decision graph and local density distribution of the Aggregation dataset. From the decision graph, Figure 1(a), only four center points were selected. However, it can be seen from the local density distribution, Figure 1(b), that, in fact, six center points are selected, which is two more than the four center points in the decision graph. As an improvement of the DPC, this result of the DPC-GD is confusing. Through further analysis, it was found that the geodesic distance represented by the shortest path calculated based on the k-nearest neighbors may become infinite, which makes the distance δ become infinite and unable to be displayed in the decision graph. When one determines the boundary point between the center and non-center points in the decision graph, since the distance δ is infinite, as long as its local density ρ is higher than the boundary point (although it cannot be displayed in the decision graph), this point will still be selected as the center point. For the above Aggregation dataset, the local density ρ and distance δ of the center points are shown in Table 1. The δ123 = ∞, δ637 = ∞ makes their corresponding points x123, x637 become center points. Therefore, the other main motivations of this article are to inherit the advantages of geodesic distance to optimize the DPC and overcome defects in the DPC-GD due to the introduction of geodesic distance.

To realize the two motivations of this article, a density peaks clustering algorithm based on weighted k-nearest
neighbors and geodesic distance (DPC-WKNN-GD) is proposed. The main contributions of the DPC-WKNN-GD are summarized as follows.

1) The DPC-WKNN-GD is proposed to improve the clustering performance for manifold and non-manifold datasets. The DPC-WKNN-GD introduces weighted k-nearest neighbors based on Euclidean distance to optimize local density $\rho$, and redefines the distance $\delta$ based on geodesic distance.

2) As a DPC-GD-inspired algorithm, the DPC-WKNN-GD overcomes the defects of the DPC-GD that the center points cannot be correctly displayed in the decision graph, or that a misleading cluster number conclusion is reached, and inherits the advantages of geodesic distance.

3) The experimental results on artificial and real-world datasets, including image datasets, show that the DPC-WKNN-GD outperforms the state-of-the-art comparison algorithms on both manifold and non-manifold datasets. In addition, the DPC-WKNN-GD completely overcomes the defect of the DPC-GD, in which the decision graph displays misleading center points.

II. RELATED WORKS
A. DPC ALGORITHM
DPC is proposed to recognize clusters of arbitrary shapes. It is conducted based on two assumptions: (1) cluster centers have higher local density than their neighbors, and (2) center points are positioned far from each other. It maps data with arbitrary dimensions onto a two-dimensional (2D) space, identifies specific density peaks as cluster centers, and then assigns each point to the corresponding cluster.

Assume that point $x_i \in \mathbb{R}^m$, $i = 1, \ldots, n$ belongs to dataset $X$ with $m$ attributes. Let $d_{ij}$ represent the Euclidean distance between $x_i$ and $x_j$. DPC computes local density $\rho_i$ and distance $\delta_i$ from points with higher local density. Local density can be defined by counting the number of points in its neighborhood,

$$\rho_i = \sum_{j=1}^{n} \chi(d_{ij} - d_c), \quad (1)$$

where $\chi(d) = 1$ if $d < 0$ and $\chi(d) = 0$ otherwise, $d_c$ is the pre-specified cutoff distance and the Gaussian kernel local density is

$$\rho_i = \sum_{j=1}^{n} \exp \left(-\frac{d_{ij}^2}{2d_c^2}\right). \quad (2)$$

Thus, $\delta_i$ is measured by computing the minimum distance between $x_i$ and any other points with higher local density,

$$\delta_i = \begin{cases} \min_j(d_{ij}), & \exists j \text{ s.t. } \rho_j > \rho_i, \\ \max_j(d_{ij}), & \text{otherwise}. \end{cases} \quad (3)$$

These points ($\rho_i$, $\delta_i$) are referred to as decision graph plotting in 2D space. Center points are significantly separated from non-center points and can be selected from the decision graph by choosing only points with high $\rho$ and relatively large $\delta$. The number of clusters is also determined to correspond the number of center points. Subsequently, each remaining point is assigned to the same cluster as its nearest-neighbor point with higher local density.

B. LOCAL DENSITY OPTIMIZATION BASED ON kNN
The local density optimization based on k-nearest neighbors (kNN) is mainly performed to solve the difficulty of determining the pre-specified cutoff distance $d_c$. The selection of $k$ is more convenient than the determination of $d_c$, and it will be beneficial to improving clustering performance. Du et al. [25] proposed DPC-KNN using mean kNN distance. Chen et al. [26] computed the local density based on the distance from the point to its $k$th nearest neighbor. Liu et al. (ADPC-KNN) [27] and Xu et al. (EADP) [28] proposed adaptive density peak clustering summing kNN with adaptive $d_c$. Xie et al. [29] proposed FKNNPC and optimized the local density with the distance sum to kNN. Geng et al. [30] used relative kNN kernel density to replace local density. FNDP [31] used fuzzy neighborhood relationships to define local density, and SNNPC [32] used shared-nearest-neighbor to calculate local density. Li et al. [33] used shared-nearest-neighbor to optimize hyperspectral band selection and simultaneously optimize the local density with kNN. Seyedi et al. [34] proposed DPC-DLP, which employs kNN to compute the global cutoff distance $d_c$ to optimize the local density. In general, the above local density based on kNN has the meaning of averaging. In most cases, modifying the calculation method, in which each point contributing to the local density is an average, can improve clustering performance. For example, Yu et al. [14] proposed DPCSA using weighted local density and achieved the purpose of improving clustering performance.

Therefore, in this article, weighted kNN is proposed for calculating local density to increase the separation between center and non-center points in the decision graph, thus improving the clustering performance.

C. DISTANCE OPTIMIZATION
Distance $\delta$ plays an equally important role as local density $\rho$, which will significantly affect the selection of center points and the assignment of cluster labels. Researchers have also proposed many improved methods. Jiang et al. [35] proposed DPC-KNN, which integrates the idea of kNN into the formula of distance $\delta$. Liu et al. [32] proposed a compensation mechanism based on proximity distance, namely multiplying these two points Euclidean distance, to calculate distance $\delta$, which may also be high value of point in low-density clusters. Du et al. [23] redefined $\delta$ using density-adaptive distance instead of Euclidean distance. The density-adaptive distance relies on the shortest path; this core idea was also adopted by DPC-GD [17], which uses geodesic distance instead of
Euclidean distance to calculate distance $\delta$. Note that these two methods only use a new distance instead of Euclidean distance, maintaining the calculation method consistent with the DPC.

D. GEODESIC DISTANCE

Indeed, DPC encounters difficulties when the data are non-linear structures like the spiral illustrated in Figure 2 (a). The distance between two points is measured by Euclidean distance as in Figure 2 (b), which ignores the intrinsic geometry of the data. Instead, they must be measured as in Figure 2 (c), i.e., along the spiral.

The geodesic distance can capture the geodesic manifold distance between pairs of points. The crux is estimating the geodesic distance between faraway points, given only input-space distances. For neighboring points, input space distance provides a good approximation to the geodesic distance. For faraway points, geodesic distance can be approximated by summing a sequence of “short hops” between neighboring points. These approximations are computed efficiently by finding the shortest path in a graph with edges connecting neighboring data points.

III. DPC-WKNN-GD ALGORITHM

A. LOCAL DENSITY $\rho$

The definition of local density should reflect the local geometric of the data as much as possible, and can maximize the density difference. Note that, in this subsection, only Euclidean distance is used.

Let $d_{ij}$ represent Euclidean distance between $x_i \in D$ and $x_j \in D$, and $KNN_i$ be the set of the k-nearest neighbors of $x_i$ based on Euclidean distance. As one of our motivations, increasing the separation of local density based on kNN, the weight $\lambda_{ij}$ between $x_i$ and $x_j$ is defined as follows:

$$\lambda_{ij} = \frac{d_i - d_{ij}}{d_i \cdot (K - 1)},$$

where $d_i = \sum_{j \in KNN_i} d_{ij}$ is the sum of Euclidean distance between $x_i$ and $x_j \in KNN_i$. The weight $\lambda_{ij}$ is inversely proportional to their distance; that is, the greater the distance $d_{ij}$, the smaller the weight $\lambda_{ij}$. Then, the local density $\rho_i$ is defined as follows:

$$\rho_i = \sum_{x_j \in KNN_i} \lambda_{ij} \exp(-d_{ij}),$$

which is a weighted Gaussian kernel density with bandwidth equal to 1 based on kNN. The benefit of weighting is that all points in the neighborhood contribute to the local density changed from no difference to dependent on relative position, which improves the distinction and ultimately helps the clustering performance. The following example will explain this conclusion more clearly. The example data are shown in Figure 3.

Six different methods were used to compute the local density of $a$, $b$, $c$ points, where DPC, DPC-GD adopting Gaussian kernel density without kNN, and ADPC-KNN, FKNNDPC, DPC-KNN, DPC-WKNN-GD adopting Gaussian kernel density based on kNN. All results are shown in Table 2. For the convenience of comparison and analysis, the local densities have been normalized. To the DPC and DPC-GD, the local densities of these three points are significantly different, and the maximum value is obtained at $c$. The results of the ADPC-KNN are contrary to those...
of the DPC and DPC-GD. The difference of $a, b, c$ local densities is so small that they are almost indistinguishable. However, the $a, b$ local densities of ADPC-KNN are equal, which is consistent with the results of FKNNPDC, DPC-KNN, and DPC-WKNN-GD. This conclusion is mostly in line with the real situation, because the third-nearest neighbors are located on the same concentric rings with different distribution positions. The third-nearest neighbors of $c$ are distributed on the second circle at the same time, and only the DPC-WKNN-GD correctly reflects the maximum local density difference. Therefore, the DPC-WKNN-GD can show the local density difference at different points as accurately as possible, just as one of the motivations of this article improves the distinction.

### B. DISTANCE $\delta$

To extend DPC to manifold datasets, the geodesic distance is used as the DPC-GD adopted. However, for the sake of avoiding the defect of the DPC-GD in which the center points in the decision graph disappear and cause a misleading number of clusters, and ensuring the high clustering performance, the calculation method of the distance $\delta$ is redefined based on both Euclidean and geodesic distance.

Let $d_{ij}, d_{ij}'$ represent Euclidean and geodesic distances between $x_i$ and $x_j$, respectively. Then, the distance $\delta_i$ corresponding to $x_i$ can be defined as follows:

$$
\delta_i = \begin{cases} 
\min_{j: \rho_j > \rho_i} d_{ij}' & \forall j \rho_j > \rho_i \text{ and } d_{ij}' \neq \infty, \\
\min_{j: \rho_j > \rho_i} d_{ij} & \forall j s.t. \rho_j > \rho_i \text{ and } d_{ij}' = \infty,
\end{cases}
$$

(6)

where

$$
\delta_{\max} = \max_{i,j,d_{ij}' \neq \infty} d_{ij}'.
$$

(7)

According to the definition and calculation of geodesic distance, it is known that (1) the geodesic distance is likely to be infinite, especially at the peaks points, and (2) any two points $x_i, x_j$ satisfy $d_{ij} \leq d_{ij}'$. Considering the definition Eq. (7), it is easy to know that $\forall i, \delta_i \leq \delta_{\max}$ holds. This can ensure that the point with $\rho_{\max}$ can be selected as the center point. The $d_{ij}' = \infty$ is the most important factor that the DPC-GD’s center points out of the decision graph; namely, they disappear. This will seriously affect the selection of the center points manually, resulting in a confusing selection of the number of center points. The incorrect number of center points (that is, the number of clusters) will cause a sharp drop in clustering performance. This is another motivation of this article. The distance $\delta$ is redefined to overcome the center points being out of the decision graph, so as to determine the number of clusters more accurately and improve the clustering performance for manifold and non-manifold datasets.

### C. ALGORITHM AND COMPLEXITY

The detail algorithm flow of the DPC-WKNN-GD is shown in Algorithm 1. Compared with DPC, only two steps (steps 2 and 4) are added and two steps (steps 3 and 5) are improved. In terms of complexity, geodesic distance calculation (step 4) involves solving the shortest path; in general, the Floyd or Dijkstra algorithms can be used with complexity $O(n^3)$, where $n$ is the total number of points on the dataset. The weight in step 2 based on kNN would cost $O(K_1n^2)$. When sorting local density costs $O(n \log n)$, the assignment procedure in step 7 is just $O(n)$. Therefore, the total complexity of the DPC-WKNN-GD is $O(n^3)$.

**Algorithm 1 DPC-WKNN-GD**

**Input:** Dataset: $D$; kNN to local density: $K_1$; kNN to geodesic distance: $K_2$

**Output:** Cluster label of each point

1. Calculate Euclidean distance matrix $[d_{ij}]$
2. Calculate weight matrix $[w_{ij}]$ using Eq. (4).
3. Calculate local density $\rho_i$ for each $x_i$ based on $K_1$ using Eq. (5).
4. Calculate geodesic distance matrix $[d_{ij}']$ based on Euclidean distance matrix and $K_2$.
5. Calculate distance $\delta_i$ for each $x_i$ using Eqs. (6) and (7).
6. Plot decision graph and select cluster centers.
7. Assign each remaining point to a cluster as DPC.
8. Return each point cluster label.

### IV. EXPERIMENTS AND RESULTS

In this section, DPC-WKNN-GD testing and verification for clustering performance compared with the well-known DPC-GD [17], SNNDPC [32], and including two non-DPC algorithms, DBSCAN and k-means++, is discussed. DPC-GD is an important reference for the proposed algorithm, DPC-WKNN-GD, especially the geodesic distance, is introduced into it. SNNDPC is one of the state-of-the-art DPC improved algorithms. DBSCAN is a common density-based algorithm that is different from DPC. K-means++ is the most commonly used partition-based algorithm with different initializations of the centroids to reduce the sensitivity. Its results are the same for every run, and there is no providing the statistical results. DBSCAN and k-means++ are called from scikit-learn [36]. SNNDPC source code is provided by the original authors with MATLAB [32]. DPC-GD is designed referring to [17] by us with Python.

### A. DATASETS AND EVALUATION METRICS

All datasets were acquired from various papers [8], [17], [32], [37]–[39] or the UCI repository [40], and they are

### TABLE 2. Local density of $a, b, c$ in Figure 3.

| Algorithm | Local density of $a, b, c$ | Parameter | Ref. |
|-----------|-----------------------------|-----------|-----|
| DPC       | $0.1092, 0.4386, 0.7049$    | $d_c = 3.0$ | [8] |
| DPC-GD    | $0.0322, 0.8995, 0.9194$    | $d_c = 3.0, K = 1$ | [17] |
| ADPC-KNN  | $0.6068, 0.6068, 0.6426$    | $K = 3$ | [27] |
| FKNNPDC   | $0.5603, 0.5603, 0.4527$    | $K = 3$ | [29] |
| DPC-KNN   | $0.4332, 0.4332, 0.5416$    | $K = 3$ | [25] |
| DPC-WKNN-GD | $0.6124, 0.6124, 0.4006$ | $K = 3$ | this paper |
summarized in Tables 3 and 4. Except for three image datasets, each attribute was normalized to avoid numerical range influence from the attribute values.

For image preprocessing, consistency with the DPC-GD was chosen. The Columbia University Image Library (COIL-20) [37] consists of $128 \times 128$ grayscale images drawn from 20 clusters. Some examples are shown in Fig. 4 (a). The UMIST face database [38] consists of 575 images of 20 people. The cropped versions of $112 \times 92$ images are selected in this article. All images of the first 10 people on the cropped dataset are taken to form this dataset, which is used in this article. It consists of 265 images of 10 people. The number of each cluster is uncertain, the largest one contains 38 objects (images), and the minimum contains 19 objects (images). Some examples are shown in Fig. 4 (b). The USPS handwritten digit database [39] contains $16 \times 16$ handwritten digit images. The clustering of the digits from five digits \{0 2 4 6 8\} on the dataset containing 1093 images was selected. Similar to UMIST, the number of each cluster is also uncertain; the largest one contains 359 objects and the minimum 166 objects. Some examples are shown in Fig. 4 (c). Note that, because the image is high-dimensional data, the Euclidean distance of the image is not directly calculated in this article, but the Complex-Wavelet Structural similarity (CW-SSIM) [17], [41] is computed and then subtracted from 1 to replace the Euclidean distance in the algorithm.

An appropriate and uniform evaluation index is both required and meaningful to compare the different clustering algorithms. Therefore, the quality was measured via Clustering Accuracy (ACC) [42] and the Normalized Mutual Information (NMI) [43] between the produced clusters and the truth categories. Larger evaluation index values indicate improved clustering performance, and all index upper bounds $= 1$, representing perfectly correct clustering:

$$ACC = \sum_{i=1}^{n} \delta(s_i, map(r_i)),$$

(8)

$$NMI = \frac{I(\text{true label}; \text{predict label})}{\sqrt{H(\text{true label})H(\text{predict label})}},$$

(9)

B. EXPERIMENTS ON ARTIFICIAL DATASETS

In this section, SNNDPC, DPC-GD, DBSCAN, k-means++, and DPC-WKNN-DPC clustering performance is shown on artificial datasets from Table 3. The NMI and ACC are shown in Tables 5 and 6, respectively. The best clustering performance evaluation values are shown in bold, and 1 means that the clustering result is completely correct.
From the view of NMI in Table 5, the most significant conclusion is that the best algorithm for clustering performance is DPC-WKNN-GD, which achieves the maximum NMI value for 13 of the 14 datasets, and the worst is k-means++, which achieves the maximum NMI value for only one of the 14 datasets. For the remaining three algorithms, DPC-GD and DBSCAN provide accurate clustering results on several manifold datasets, such as CirclesA1, CirclesA3, FourLines, Jain, and ThreeCircles, which outperform the SNNDPC. However, on other non-manifold datasets, such as R15, S1, and S3, the SNNDPC shows its advantages and is superior to the DPC-GD and DBSCAN. Further analysis can find, in addition to those datasets with completely accurate clustering results, whether it is on manifold or non-manifold datasets, that the DPC-WKNN-GD has improved clustering performance compared to the DPC-GD, such as the Aggregation, Pathbased, R15, S1, and S3 datasets.

The evaluation conclusion of ACC in Table 6 is almost the same as that of NMI in Table 5. Two different evaluation indexes obtained the same evaluation conclusion, which increases the reliability of the clustering performance improvement of the DPC-WKNN-GD. The only difference between DPC-WKNN-GD and k-means++ is that their clustering results are reversed on the S3 dataset; that is, the latter outperforms the former. In short, the overall conclusion remains the same; that is, DPC-WKNN-GD is still clearly superior to the other four algorithms.

Since these are 2D datasets, it is more straightforward to display their clustering results as colored plots. Therefore, in the following analysis, the clustering results of various algorithms are displayed in the form of scatter plot. Each cluster, both center and non-center points, corresponds to each color. Non-center points are shown as filled circles and center points are drawn as large solid shapes, squares for SNNDPC and pentagrams for the other three algorithms, except for DBSCAN without center points.

Figures 5 (a)-(e) show the SNNDPC, DPC-GD, DBSCAN, k-means++, and DPC-WKNN-GD clustering results on the Aggregation dataset. The number of k-means++ was set to 7. Its clustering result in Figure 5 (d) shows that two center points are located into the same cluster, and two different clusters are divided together. The SNNDPC, DBSCAN, and DPC-WKNN-GD identify seven clusters successfully, while the DPC-GD makes an erroneous decision on the S3 dataset; that is, DPC-GD detected eight center points. However, the DPC-GD decision graph only shows four center points in Figure 12 (a) in Section IV-D. This difference in the number of center points is due to the fact that distance $\delta$ of some points are infinite due to the direct replacement of the Euclidean distance with the geodesic distance. In the DPC-WKNN-GD, this defect is overcome. The decision graph in Figure 13 (a) detected seven center points, and the correct clustering was performed accordingly as shown in Figure 5 (e). Compared with SNNDPC, the boundary points cluster label assignment of the DPC-WKNN-GD is more reasonable. For example, some boundary points between the green cluster and the light blue cluster in Figure 5 (a) are erroneously assigned, while the corresponding boundary points in Figure 5 (e) are more accurately assigned.
The clustering results on the CirclesA1 dataset are shown in Figure 6. The DPC-GD, DBSCAN, and DPC-WKNN-GD accurately divided those points into three clusters, but the SNNDPC accidentally detected only two center points, which led to incorrect clustering results. The closed-loop data points distribution affects the center point recognition of the SNNDPC.

The CirclesA3 dataset contains three clusters with a semi-circular and two rectangular distributions. The clustering results are shown in Figure 7. Since the semi-circle surrounds the two rectangles, the k-means++ splits the semi-circle that originally belonged to the same cluster into three different clusters [Figure 7 (d)]. Therefore, the k-means++ is not suitable for clustering the CirclesA3 dataset. The remaining four density-based algorithms, SNNDPC, DPC-GD, DBSCAN, and DPC-WKNN-GD, can correctly distinguish three clusters.

Figure 8 shows the results of each algorithm on the Flame dataset. As shown, the SNNDPC, DPC-GD, and DPC-WKNN-GD can correctly identify two center points. However, only the clustering results of the DPC-GD and DPC-WKNN-GD are completely correct. The following algorithms are the SNNDPC and DBSCAN with NMI = 0.8993. Note that the DBSCAN gives three isolated clusters and divides into five clusters eventually. The k-means++ assigns the left-hand end of the lower cluster to the upper cluster such that the entire dataset resembles being slashed from the diagonal, which also leads to serious errors.

The clustering results on the FourLines dataset are shown in Figure 9. From the scatter plot, the conclusion is very similar to that of the CirclesA3 dataset. The k-means++ also gives a poor clustering performance. The SNNDPC just detects three center points such that the lower two clusters are merged into one cluster, which is beyond our understanding. However, the DPC-GD, DBSCAN, and DPC-WKNN-DPC do not have the same error as the SNNDPC, and show perfect clustering results.

The Jain dataset shows two crescent-shaped clusters of different densities intertwined with each other. Its clustering results of five algorithms are shown in Figures 10 (a)-(e), respectively. The two center points of the k-means++ are located in the lower cluster, leaving the upper cluster with no center point and splitting the lower cluster (real cluster) into two clusters mistakenly [see Figure 10 (d)]. The SNNDPC, DPC-GD, and DPC-WKNN-GD all correctly identify the two center points and distinguish the two clusters perfectly. The DBSCAN also clusters this dataset correctly.

The points of the Pathbased dataset form two spherical clusters and a ring cluster, and the ring cluster surrounds the two spherical clusters. The clustering results of the five algorithms on this dataset are shown in Figure 11. Figures 11 (b) and (e) show that, although the DPC-GD and DPC-WKNN-DPC correctly identified the three center points, they fail to cluster. The real ring clusters are split into two or three clusters. For the k-means++, although set to
three clusters, the ring cluster is split into three clusters, for which the left- and right-hand sides of the half-ring clusters are incorrectly assigned to the other two clusters, leaving the half-ring cluster only a small part on the top. This result is very similar to that of DPC-GD. Since the left-hand and top parts are clustered together, the DPC-WKNN-GD outperforms the DPC-GD and k-means++. However, its clustering performance is still inferior to that of the SNNDPC.

R15, S1, and S3 are three datasets without manifold distribution and having 15 clusters. Their clustering results for the five different algorithms are shown in Figures S1, S2 and S3 (‘S’ represents Supplement), respectively. Except for the k-means++ being set to 15 clusters, all other algorithms detect and cluster into 15 clusters. Since there is low overlap and large separation between clusters in the R15 and S1 datasets, all five algorithms achieved high clustering performance. In the S1 dataset, it can be found from the clustering graph that the DBSCAN easily divides the points distributed on the edge of the cluster into other clusters or being isolated clusters by mistake. This type of error is successfully avoided in the SNNDPC, DPC-GD, and DPC-WKNN-GD. In the S3 dataset, due to the addition of more noisy data, the overlap is severe and the boundaries are blurred, and the clustering performance of the five algorithms has decreased. Nevertheless, the SNNDPC, DPC-GD, and DPC-WKNN-GD still detect and identify 15 center points. Furthermore, the DPC-WKNN-GD obtains no distorted clusters and maintains high clustering performance. The evaluation value $NMI = 0.7994$ shows that the DPC-WKNN-GD still outperforms the other four algorithms.

The data distribution of the four datasets Spiral, Three-Circles, ThreeLines, and ThreeMoons is very special. Many algorithms are difficult to work, such as k-means++, which is a partition-based clustering algorithm. It can be seen from sub-figures (d) of Figures S4-S7 that, although set to the correct number of clusters, satisfactory clustering results are not given on any one dataset. Unlike the clustering results of the k-means++, the results in sub-figures (b), (c), and (e) of Figures S4-S7 show that the DPC-GD, DBSCAN, and DPC-WKNN-GD give perfect clustering results. Owing to the identification of the incorrect center points, the clustering performance of the SNNDPC on the ThreeCircles dataset drops sharply [see Figure S5 (a)]. However, on the other three datasets, Spiral, ThreeLines, and ThreeMoons, the SNNDPC also gives perfect clustering results. Examining the results of its clustering on all datasets, an interesting finding is that the SNNDPC does not seem to work well on manifold datasets with closed ring distribution; for example, Figures 6 (a) and S6 (a).

In conclusion, it can be observed that the DPC-WKNN-GD outperforms the other four algorithms on most datasets, and obtains perfect clustering results on many datasets. For a few datasets, such as Pathbased, it is slightly poorer than the SNNDPC and DBSCAN, but still better than the DPC-GD.
and k-means++. As an improved DPC algorithm that introduces geodesic distance, the DPC-WKNN-GD outperforms the DPC-GD on all datasets, whether it is manifold or non-manifold. In addition, on non-manifold datasets, such as Aggregation, R15, S1, and S3, the DPC-WKNN-GD either maintains or exceeds the clustering accuracy of the SNNDPC, while the latter has an insufficient number of advantages on manifold datasets.

C. EXPERIMENTS ON REAL-WORLD DATASETS

In this section, a comparative analysis of clustering performance on 12 datasets, including three image datasets, is conducted. Please note that the k-means++ is not suitable for clustering these three image datasets due to the distance matrix they provide. Therefore, the clustering results of the k-means++ will not be provided in this article. In the corresponding table, there is no value indicated by “−”. Values corresponding to the best clustering performances are in bold.

Table 7 shows the NMI of different algorithms on real-world datasets. The DPC-WKNN-GD achieved the best clustering performance with highest NMI for seven of the 12 datasets and outperformed the other four algorithms with a significant advantage, especially in the three image datasets. In the remaining five datasets, the DPC-WKNN-GD is inferior to the optimal comparison algorithm with a slight gap. The SNNDPC follows the DPC-WKNN-GD and achieves the best clustering performance for four of the 12 datasets. The DBSCAN and k-means++ provides the poorest clustering performance. As a DPC-GD-inspired algorithm, the DPC-WKNN-GD is superior to the DPC-GD on all datasets. Only on the Parkinsons and COIL-20 datasets did they achieve the same performance and outperform the other three algorithms. In other words, the DPC-WKNN-GD is significantly better than the DPC-GD in clustering performance.

Images (pixel matrix) are very special data types. Generally, pixel matrices are stretched into a vector and image clustering is converted into challenging high-dimensional data clustering. Many clustering algorithms are unable to obtain satisfactory clustering results. In this article, three image datasets (see Table 4) are used to verify the performance of the proposed clustering algorithm. The SNNDPC, DPC-GD, DBSCAN, and DPC-WKNN-GD are used for
On the COIL-20 dataset, the DPC-WKNN-GD and DPC-GD achieved perfect clustering results with $NMI = 1$. The second one was the DBSCAN with weak disadvantage. The worst was the SNNDPC with $NMI = 0$. The proposed algorithm, DPC-WKNN-GD, was in second place.

Although the clustering results of the four algorithms on the USPS dataset far from completely distinguished all clusters, the DPC-KNN-GD outperformed the DPC-GD with a small advantage and became the optimal clustering algorithm with $NMI = 0.7825$. Therefore, the DPC-WKNN-GD has great performance advantages in clustering analysis image data.

The ACC evaluation results are shown in Table 8 and are consistent with the NMI in Table 7. The algorithm with the best clustering performance is the DPC-WKNN-GD, followed by the SNNDPC and DPC-GD. The two evaluation indexes achieve consistent evaluation results, which improves reliability.

In conclusion, on the real-world datasets, the DPC-WKNN-GD is a clustering algorithm with great advantages, especially on image datasets. It obtains the optimal performance on most datasets. If it is inferior to other algorithms, the gap between them is small. For example, on the Iris, Wine, Wbdc, and Dermatology datasets, the DPC-WKNN-GD clustering results are very close to those of the optimal clustering algorithm. The DBSCAN and k-means++ have poor clustering performance on the real-world datasets. In addition, compared with the DPC-GD, the DPC-WKNN-GD improves the clustering performance on all real-world datasets.

The decision graph is a major innovation of the DPC, mainly to solve the selection of the center points and the determination of the number of clusters. As mentioned in Section I, the DPC-GD introduces geodesic distance to promote clustering analysis on manifold datasets. However, the concomitant defect is that the center points in the decision graph are confusing. Without knowing the actual number of clusters, one cannot simply select the appropriate center points and determine the correct number of clusters.

Figure 12 shows the decision graphs of the DPC-GD on the Aggregation, CirclesA1, CirclesA3 and Flame datasets. In the decision graph of Figure 12(a), the DPC-GD detected four center points; however, the clustering result in Figure 5(b) has eight center points, which is obviously more than the number of center points in the decision graph. It can be seen that the decision graph of the DPC-GD has lost its meaning and even played a misleading role. This phenomenon has confused users. The reason has been thoroughly analyzed and an improved algorithm, DPC-WKNN-GD, has been proposed. Figure 13(a) shows the decision graph of the DPC-WKNN-GD. All seven center points are displayed in the decision graph, and the final clustering result in Figure 6(e) comprises only center points corresponding to the decision graph, where five center points are significantly far from the non-center points, and two center points are slightly close to the non-center points, but can still be distinguished. For the CirclesA1 dataset, only by selecting a center point from the decision graph [see Figure 12(b)] can the DPC-GD have three correct center points and completely distinguish three clusters during actual clustering. While the DPC-WKNN-GD displays three center points in the decision graph of Figure 13(b), the perfect clustering result in Figure 6(e) with $NMI = 1$ indicates that the center points selected in the decision graph are correct. The situation is similar for the CirclesA3 dataset. The DPC-GD only selects two center points from the decision graph of Figure 12(c), but the final clustering result has three center points. For the DPC-WKNN-GD, three center points are selected in the decision graph of Figure 13(c) and become the final correct clustering center points.

For the Flame and Jain datasets, the DPC-GD and DPC-WKNN-GD decision graphs [Figures 12(d), 13(d), 14(b), and 15(b)] identify two center points, respectively. Based on the correct identification of the center points,
the gap between the center and non-center points of the DPC-WKNN-GD is greater than that of the DPC-GD, which will make it easier to obtain the correct center points. For the FourLines dataset, the defect of the DPC-GD has appeared, and it just identified one center point from the decision graph of Figure 14(a), while for the DPC-WKNN-GD, all four center points appear in the upper right-hand corner of the decision graph of Figure 15(a) and are a significant distance from the non-center points.

For the Pathbased, R15, S1, S3, Spiral, and ThreeCircles datasets, from Figures 14 to 17, their overall trend is consistent with the six datasets analyzed above. Except for the Pathbased and S3 datasets, the DPC-GD failed to show the correct number of center points on any dataset. The DPC-WKNN-GD shows and selects the correct center points on all datasets, including Pathbased and S3.

Table 9 summarizes the number of center points in the decision graph of the DPC-GD and the DPC-WKNN-GD. The result that the number of center points determined in the decision graph is different from the real clusters is marked in black. Table 9 shows that the DPC-GD results are marked in black in nine of the 14 datasets, which means that its decision graph cannot display the correct center points in the nine datasets. If the number of clusters of the dataset is not known in advance to adjust the selection of the center points in the decision graph, it is almost impossible to determine the center points and the number of clusters in a dataset through the DPC-GD decision graph. However, although the DPC-WKNN-GD also introduces the geodesic distance, it completely overcomes the defects in the DPC-GD. The results in Table 9 show that the DPC-WKNN-GD decision graph can display the center points exactly and determine the correct number of clusters.

In conclusion, from the comparison and analysis of the number of center points in decision graph of the DPC-GD and DPC-WKNN-GD, it is difficult for the DPC-GD to determine the correct number of clusters and their corresponding center points.
FIGURE 16. Decision graphs of DPC-GD on S1, S3, Spiral, and ThreeCircles datasets.

FIGURE 17. Decision graphs of DPC-WKNN-GD on S1, S3, Spiral, and ThreeCircles datasets.

TABLE 10. Algorithm runtime (unit: s).

| Datasets  | DPC-GD | DBSCAN | k-means++ | DPC-WKNN-GD |
|-----------|--------|--------|-----------|-------------|
| Aggregation | 0.1716 | 0.0000 | 0.0530    | 0.1092      |
| CirclesA1 | 0.0936 | 0.0000 | 0.0390    | 0.0780      |
| CirclesA3 | 0.0468 | 0.0000 | 0.0359    | 0.0312      |
| Flame     | 0.0312 | 0.0000 | 0.0234    | 0.0312      |
| FourLines | 0.0936 | 0.0000 | 0.0452    | 0.0468      |
| Jain      | 0.0780 | 0.0000 | 0.0265    | 0.0624      |
| Patbased  | 0.0624 | 0.0000 | 0.0359    | 0.0468      |
| R15       | 0.0936 | 0.0156 | 0.0608    | 0.0468      |
| S1        | 13.7436| 0.0624 | 0.1264    | 10.7640     |
| S3        | 17.3004| 0.0624 | 0.2012    | 17.8776     |
| Spiral    | 0.0156 | 0.0000 | 0.0530    | 0.0468      |
| ThreeCircles | 0.1404 | 0.0000 | 0.0406    | 0.0624      |
| ThreeLines | 0.0312 | 0.0000 | 0.0312    | 0.0156      |
| ThreeMoons| 0.1248 | 0.0000 | 0.0328    | 0.0624      |

Note: 0.0000 means that its running time is less than 0.0001 s.

points through the decision graph, and the DPC-WKNN-GD successfully overcomes this defect.

E. RUNTIME COMPARISON
This section compares real runtime of DPC-GD, DBSCAN, k-means++, and DPC-WKNN-GD algorithms for different datasets, as shown in Table 10. DBSCAN was the fastest and more efficient than other three algorithms significantly. K-means++ cost less time to cluster these datasets than DPC-GD and DPC-WKNN-GD. The clustering efficiency of DPC-GD and DPC-WKNN-GD were close, and there was no significant difference.

F. SENSITIVITY ANALYSIS
The sensitivity is a measure of the stability of algorithm and avoids the erroneous evaluation of algorithm performance caused by the different parameter. We used DPC-WKNN-GD to cluster Aggregation, CirclesA1, Flame, Iris and Wine five datasets again with the different parameter $k$ ranged from 2 to 39. If there is significant difference among the NMI results, we believe that DPC-WKNN-GD is sensitivity; otherwise, it is not sensitivity. Figure 18 shows that the NMI values are stable for the same dataset without any abrupt change or severe fluctuations. In conclusion, it is clear that DPC-WKNN-GD will be affected by the parameter $k$, but not sensitivity.

V. CONCLUSION
In this article, a novel density peaks clustering algorithm based on weighted $k$-nearest neighbors and geodesic distance, namely, DPC-WKNN-GD, is proposed. This algorithm is inspired by the DPC-GD, which introduces geodesic distance to replace Euclidean distance. Moreover, the DPC-WKNN-GD overcomes the DPC-GD defect in which the center points cannot be correctly displayed in the decision graph. The DPC-WKNN-GD redefines the local density $\rho$ based on weighted $k$-nearest neighbors and Euclidean distance. It also provides a new computation method with which to compute distance $\delta$ based on geodesic
distance. Therefore, the newly defined $\rho$ and $\delta$ can reflect the attributes of points more objectively and precisely, avoiding the problem that the DPC cannot effectively address variable-density clusters. In addition, the DPC-WKNN-GD is suitable for manifold datasets.

The experimental results on artificial and real-world datasets, including image datasets, show that the DPC-WKNN-GD improves the clustering performance and significantly outperforms comparison algorithms. Moreover, the DPC-WKNN-GD overcomes the DPC-GD defect in which the decision graph cannot correctly display center points.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2011.
[2] F. Murtagh and P. Contreras, “Algorithms for hierarchical clustering: An overview,” II, WIREs Data Mining Knowl. Discovery, vol. 7, no. 6, p. e1219, Nov. 2017.
[3] D. Yu, G. Liu, M. Guo, and X. Liu, “An improved K-medoids algorithm based on step increasing and optimizing medoids,” Expert Syst. Appl., vol. 92, pp. 464–473, Feb. 2018.
[4] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in a density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. 2nd Int. Conf. Knowl. Discovery Data Mining, Portland, OR, USA, 1996, pp. 226–231.
[5] W. Wang, J. Yang, and R. Munzt, “Sting: A statistical information grid approach to spatial data mining,” in Proc. 23rd Int. Conf. Very Large Data Bases, Athens, Greece, 1997, pp. 186–195.
[6] T. Chen, N. Zhang, T. Liu, K. Poon, and Y. Wang, “Model-based multi-dimensional clustering of categorical data,” Artif. Intell., vol. 176, no. 1, pp. 2246–2269, 2012.
[7] S. Vega-Pons and J. Ruiz-Shulcloper, “A survey of clustering ensemble algorithms,” Int. J. Pattern Recognit. Artif. Intell., vol. 25, no. 03, pp. 337–372, May 2011.
[8] A. Rodríguez and A. Laio, “Clustering by fast search and find of density peaks,” Science, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[9] M. A. Masud, J. Z. Huang, C. Wei, J. Wang, I. Khan, and M. Zhong, “I-nice: A new approach for identifying the number of clusters and initial cluster centres,” Inf. Sci., vol. 466, pp. 129–151, Oct. 2018.
[10] M. Wang, W. Zuo, and Y. Yang, “An improved density peaks-based clustering method for social circle discovery in social networks,” Neurocomputing, vol. 179, pp. 219–227, Feb. 2016.
[11] S. Ding, M. Du, T. Sun, X. Xu, and Y. Xue, “An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighbor knowledge,” Knowl.-Based Syst., vol. 133, pp. 294–313, Oct. 2017.
[12] J. Cao, J. S. Packer, V. Ramani, D. A. Cusanovich, C. Huynh, R. Daza, X. Qiu, C. Lee, S. N. Furlan, F. J. Steemers, A. Adey, R. H. Waterston, C. Trappell, and J. Shendure, “Comprehensive single-cell transcriptional profiling of a multicellular organism,” Science, vol. 357, no. 6352, pp. 661–667, Aug. 2017.
[13] M. Guo, D. Yu, G. Liu, X. Liu, and S. Cheng, “Drug-target interaction clustering by finding density backbone based-on kNN,” Knowl.-Based Syst., vol. 23, no. 4, pp. 1335–1349, Aug. 2018.
[14] Z. Li and Y. Tang, “Comparative density peaks clustering,” Expert Syst. Appl., vol. 95, pp. 236–247, Apr. 2018.
[15] M. Du, S. Ding, X. Xu, and X. Xue, “Density peaks clustering using geodesic distances,” Int. J. Mach. Learn. Cybern., vol. 9, no. 8, pp. 1335–1349, Aug. 2018.
[16] X. Li and K.-C. Wong, “Evolutionary multiobjective clustering and its applications to patient stratification,” IEEE Trans. Cybern., vol. 49, no. 5, pp. 1680–1693, May 2019.
[17] M. Chen, L. Li, B. Wang, J. Cheng, L. Pan, and X. Chen, “Effectively clustering by finding density backbone based-on kNN,” Pattern Recognit., vol. 60, pp. 486–498, Dec. 2016.
[18] W. Wu and B. M. Wilamowski, “A fast density and grid based clustering method for data with arbitrary shapes and noise,” IEEE Trans. Ind. Informat., vol. 13, no. 4, pp. 1620–1628, Aug. 2017.
[19] R. Mehmoood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, “Clustering by fast search and find of density peaks via heat diffusion,” Neurocomputing, vol. 208, pp. 210–217, Oct. 2016.
[20] D. Cheng, J. Huang, S. Zhang, and H. Liu, “Improved density peaks clustering based on discovered-neighbors of local cores for manifold data sets,” IEEE Access, vol. 7, pp. 151339–151349, 2019.
[21] M. Du, S. Ding, Y. Xue, and Z. Shi, “A novel density peaks clustering with sensitivity of local density and density-adaptive metric,” Knowl. Inf. Syst., vol. 59, no. 2, pp. 285–309, May 2019.
[22] J. B. Tenenbaum, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
LINA LIU received the M.S. degree from the College of Science, Harbin Engineering University, in 2011, and the Ph.D. degree in system engineering from Harbin Engineering University, in 2015. She is currently a Lecturer with the School of Electronic and Information Engineering, Soochow University, China. Her research interests include artificial intelligence and data mining.

DONGHUA YU received the M.S. degree from the College of Science, Harbin Engineering University, Harbin, China, in 2015, and the Ph.D. degree from the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, in 2020. He is currently a Lecturer with the Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China. His current research interests include machine learning and bioinformatics.