Expert Systems With Applications 115 (2019) 314–328

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Dynamic graph-based label propagation for density peaks clustering

Seyed Amjad Seyedi[a], Abdulrahman Lotfi[a], Parham Moradi[a,*], Nooruldeen Nasih Qader[b]

[a] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran
[b] Department of Computer Science, University of Human Development, Sulaymanyah, Iraq

ARTICLE INFO

Article history:
Received 19 January 2018
Revised 30 July 2018
Accepted 31 July 2018
Available online 1 August 2018

Keywords:
Density peaks clustering
Soft clustering
Label propagation
Graph-based clustering

ABSTRACT

Clustering is a major approach in data mining and machine learning and has been successful in many real-world applications. Density peaks clustering (DPC) is a recently published method that uses an intuitive to cluster data objects efficiently and effectively. However, DPC and most of its improvements suffer from some shortcomings to be addressed. For instance, this method only considers the global structure of data which leading to missing many clusters. The cut-off distance affects the local density values and is calculated in different ways depending on the size of the datasets, which can influence the quality of clustering. Then, the original label assignment can cause a *"chain reaction"*, whereby if a wrong label is assigned to a data point, and then there may be many more wrong labels subsequently assigned to the other points. In this paper, a density peaks clustering method called DPC-DLP is proposed. The proposed method employs the idea of k-nearest neighbors to compute the global cut-off parameter and the local density of each point. Moreover, the proposed method uses a graph-based label propagation to assign labels to remaining points and form final clusters. The proposed label propagation can effectively assign true labels to those of data instances which located in border and overlapped regions. The proposed method can be applied to some applications. To make the method practical for image clustering, the local structure is used to achieve low-dimensional space. In addition, proposed method considers label space correlation, to be effective in the gene expression problems. Several experiments are performed to evaluate the performance of the proposed method on both synthetic and real-world datasets. The results demonstrate that in most cases, the proposed method outperformed some state-of-the-art methods.

1. Introduction

Clustering is a fundamental approach in data mining and its aim is to organize data into distinct groups to identify intrinsic hidden patterns of data. In other words, clustering methods divide a set of instances into several groups without any prior knowledge using the similarity of objects in which patterns in the same group have more similarities to each other rather than patterns in different groups. It has been successfully applied to various expert system fields such as image processing (Wu & Leahy, 1993) cybersecurity (Kozma, Rosa, & Piazentin, 2013), pattern recognition (Haghtalab, Xanthopoulos, & Madani, 2015), bioinformatics (Xu & Su, 2015), protein analysis (de Andrades, Dorn, Farenzena, & Lamb, 2013), microarray analysis (Castellanos-Garzón, García, Novais, & Díaz, 2013) and social networks (McGarry, 2013). Up to now, many clustering methods have been proposed in the literature. For instance, clustering methods are applied on an image to split it into several regions which dovetail with different objects. In pattern recognition, clustering methods are used in the initial steps of data analysis to identify groups of related patterns with the aim of exploring further relationships. In bioinformatics clustering methods are used to characterize genes of unknown function, reveal natural structure inherent in gene expression data, microarray dimension reduction and discovering implicit links between the genes. In cybersecurity, clustering techniques are used for reducing the amount of information to process and the energy to consume with the aim of identifying network anomalies. These methods can be classified into (Jain, 2010): partitioning-based, model-based, hierarchical-based, grid-based and density-based approaches.

The aim of partitioning methods is to group the data into a fixed number of clusters using an iterative optimization method. K-means (MacQueen, 1967) and Fuzzy C-means (Bezdek, Ehrlich, & Full, 1984) are well-known partitioning based methods. Although partitioning-based methods are simple to understand and easy to implement; they generally produce spherical clusters and not sensible to detect outlier or noise in the data. Furthermore, these methods require defining the number of clusters as an input to the

* Corresponding author.
E-mail addresses: amjadseyedi@eng.uok.ac.ir (S.A. Seyedi), a.lotfi@eng.uok.ac.ir (A. Lotfi), p.moradi@uok.ac.ir (P. Moradi), nooruldeen.qader@uhd.edu.iq (N.N. Qader).

https://doi.org/10.1016/j.eswa.2018.07.075
0957-4174/

algorithm. Model-based clustering methods generally use a statistical model to represent clusters by a different probability distribution. These algorithms often use the expectation-maximization approach to maximize the likelihood function (Dempster, Laird, & Rubin, 1977). Hierarchical clustering methods divide the data into several groups represented by a tree of nodes. Clusters are identified by merging the groups with different levels of the tree by using a minimum distance criterion (Murtagh & Contreras, 2012). Generally, these methods suffer from chaining effect and need a high computational cost. Furthermore, they require a threshold to define an appropriate stopping condition for splitting or merging of partitions (Johnson, 1967). Although they have several advantages such as a better visualization of clusters by generating a tree without predefining the number of clusters, calculating and sorting the Euclidean distances require a high computational and memory costs. On the other hand, grid-based algorithms, with high efficiency and time complexity, which are independent of the number of data objects (Yue, Wang, Tao, & Wang, 2010).

Recently, density-based clustering approaches have gained much attention among researchers. These methods suppose that clusters span in high-density regions separated by lower density areas. They are seeking to identify clusters with arbitrary shapes. Furthermore, these methods require a minimum domain knowledge to organize data into clusters (Sharma & Ramya, 2013). DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) is a well-known density-based clustering method and its aim is to identify a maximum set of density-connected points. This method has several advantages such as identifying clusters with arbitrary shapes, handles noise or outlier effectively and does not require predefining the number of clusters. However, this method suffers from several shortcomings. For example, it cannot deal with uneven density datasets. This method needs a quadratic time complexity, and its effectiveness depends on appropriate selecting its adjustable parameters. Moreover, it is not fully deterministic for border points and cannot perform well in overlapping densities. To tackle these issues, various methods such as: OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999), DVBSCAN (Ram, Jalal, Jalal, & Kumar, 2010), DBCLASD (Xu, Ester, Kriegel, & Sander, 1998), VDBSCAN (Liu, Zhou, & Wu, 2007), STDBSCAN (Birant & Kut, 2007) and DPC (Rodriguez & Laio, 2014) have been proposed.

Density Peaks Method (DPC) (Rodriguez & Laio, 2014) was recently proposed to identify non-spherical clusters and has gained much attention compared with the others. The main idea of DPC is that the cluster centers are characterized by a higher density than their neighbors. Moreover, it is supposed that these centers are located at a relatively large distance from points with higher densities. Using these assumptions, DPC first defines two metrics called: local density and minimum density-based distance. For each data point, these metrics are calculated by using the distances of this data point to the others. On the other hand, the data is mapped to a two-dimensional space and is plotted in a 2D decision graph. Then, those of the data points with both greater local density and minimum density-based distance are identified as clustering centers (or density peaks). On the other hand, the centers are identified by using a cut-off distance over the decision graph. Subsequently, in the next step, the data points are assigned to the same cluster as its nearest neighbor of higher density. DPC has several advantages compared to the other clustering methods such as: (1) it does not need to specify the predefined number of clusters, (2) it requires only a parameter with lower sensitivity, and (3) It is computationally efficient compared to the other density-based methods. However, DPC suffers from several issues. First, arbitrarily selecting the cut-off distance may greatly influence the clustering of small datasets. Second, identifying clustering centers needs assigning minimum thresholds for the mentioned metrics. The proper setting of these thresholds depends on the data. Third, after identifying the density peaks, the strategy of assigning each point to its nearest neighbor with higher density can cause the propagation of errors. Consequently, the human-based selection of cluster centers and the sensitivity to its parameter are the main limitations toward DPC.

Recently, Du, Ding, and Jia (2016) hybridized the density peaks clustering with k-NN in order to take into account both local and global structures of data. This method, called DPC-KNN also employs PCA (Pearson, 1901) method to perform well on high dimensional data. However the method has also suffered from several limitations. First, it cannot discover clusters with non-spherical shapes. Second, it does not perform well to identify overlapped clusters. Third, this method leads to identifying weak clusters on data with imbalanced clusters. In Lotfi, Seyedi, and Moradi (2016) a method called IDPC is proposed which improves DPC-KNN by utilizing a specific label propagation method based on a voting strategy and local densities of objects. The method cannot perform well on data with overlapped clusters either. In this paper, a novel density peaks clustering method called DPC-DLP is proposed. The proposed method uses a novel dynamic graph-based label propagation method by considering the correlation between instances and local structure of data. DPC-DLP consists of three main steps. In the first step, a local density measure is utilized to identify cluster centers. In the second step, a graph is constructed using the k-NN method for cluster cores. In this step, each cluster center and its corresponding k-nearest neighbors form a cluster core. In the third step, a novel graph-based label propagation is used to spread the labels to the rest instances. The proposed method can be effective for some real-world applications such as image clustering and gene expressions. In traditional image clustering, images are represented as points in a high-dimensional space where each pixel is one dimension. In this case, a very high-dimensional space is clustered. To tackle this issue, the proposed manifold learning is simply used to find a low-dimensional representation of the original image set. Moreover, in gene expression tasks, clustering methods are applied for discovering groups of genes having a homological pattern of variation in their expression values. To detect highly correlated clusters with the biological concept, a fusion graph is constructed in such a way that the correlation between instances is employed in edge-weighting process.The proposed method has several novelties compared to state-of-the-art methods listed as follows:

- Most of DPC papers such as Ding, He, Yuan, and Jiang (2017) and Rodriguez and Laio (2014) employ the cut-off distance to estimate densities, thus using this measure it is difficult to make a reliable estimate of densities, especially for small datasets. To this end, in this paper, a local density metric introduced in Du et al. (2016) is employed to calculate the densities. This measure seeks to widen the gap of the density between cluster cores and border instances (i.e. outliers) in order to easy identification of core objects. In other words, it increases the separability of cluster cores by reducing the impact of outlier objects in calculating densities. Consequently, using this local measure leads to identify precise and effective cluster cores.
- In most of the state-of-the-art DPC-based methods such as Lotfi et al. (2016), each data instance gets the label of its nearest cluster core using a well-known distance metric such as Euclidean distance metric. This type of label propagation method leads to identifying only those of spherical shapes clusters. Inspired by Souvenir and Pless (2005) and Yankov and Keogh (2006), in this paper, a novel graph-based label propagation method is used to form both spherical and non-spherical cluster shapes with various densities. To this end, a nearest-neighbor graph is generated in order to preserve the geometrical structure of data space. Using this idea, it is supposed that

most data instances of the same cluster are close to each other through sequences of nearby instances. This kind of local structure provides an ability to discover clusters of complex data with non-spherical shapes.

- Inspired by the original DPC method, most of the state-of-the-art density peaks clustering methods (Du et al., 2016; Xie, Gao, Xie, Liu, & Grant, 2016), a single-step label assignment strategy is used to form final clusters leading to misclassifying those of data instances located in the border and overlapped regions. A multi-step label propagation which assigns cluster labels to data instances through several iterations is used in this paper. To this aim, in each iteration, the gradual membership values for data instances are updated using a specific similarity measure. In other words, the proposed label propagation method allows the definition of clusters with tightly co-expressed samples. The proposed method belongs to soft clustering (Yu, Yu, & Tresp, 2006) that can effectively reflect the strength of a sample's association with a cluster as well as displays more noise robustness.
- In most of existing density peaks clustering methods such as Du, Ding, and Xue (2017b) and Rodriguez and Laio (2014), if a data instance takes a wrong label, due to the *"chain reaction"*, leading to propagating the error and thus more points are assigned with incorrect labels. In other words, errors may be propagated in the next assignment process and there is no way to handle it. This is due to the fact that existing label propagation processes based on the feature space similarity are poorly guided. To solve this issue, in this paper, inspired by Wang, Tu, and Tsotsos (2013) a dynamic label propagation that updates similarity values (i.e. fusion matrix) by hybridizing label space correlation and local structure. This label propagation controls the propagation of labels and prevents the *"chain reaction"*. The main idea behind this kind of label propagation is based on the assumption that if two instances have a high similarity value in the input data space, they are likely to have a high correlation value in the label space. Consequently, the proposed label propagation strategy has a high separation capability of data with imbalanced clusters resulting in improving the quality of the clustering results.

2. Related works

Clustering aims to group a set of data objects into different clusters in order to achieve high intra-cluster connectivity as well as a low inter-cluster similarity. Data clustering methods have been widely used in machine learning, pattern recognition and video and image analysis. Over the past decades, a large number of clustering algorithms has been successfully proposed from various perspectives. DPC method is a novel and recently published clustering method. DPC is relying on the idea that cluster centers are placed on higher density regions compared to their neighbors. Also, they are relatively far from points with higher densities. Using this idea the number of clusters is determined intuitively, outliers are automatically identified and ignored from the analysis, and clusters are formed regardless of their shapes (Rodriguez & Laio, 2014). In other words, DPC is able to form non-spherical clusters without requiring the number of clusters. Furthermore, DPC identifies final clusters without any iteration and has many advantages. However, the human-based selection of cluster centers and the sensitivity to its parameter are the main limitations toward DPC. There are several improvements to address the limitations of the original DPC method. The DPC extensions are focused on (1) proposing more effective metrics for measuring the density values, (2) proposing some strategies for determining the number of clusters automatically, (3) proposing effective label assignment mechanisms and (4) extending the DPC to be applied on specific applications types such as mixed data, hyperspectral band selection and big data analysis.

The first category of DPC extensions aims to develop effective kernel methods for computation of local densities. These kernels are proposed in such a way to have a higher impact on choosing cluster centers and reducing the sensitivity of the clustering method to the value of the cut-off distance parameter. Mehmood, Zhang, Bie, Dawood, and Ahmad (2016) proposed a method called CFSFDP-HD by using a heat equation to better estimate the density results in reducing the sensitivity to the value of the cut-off parameter. This method uses a non-parametric density estimator with lower sensitivity to the cut-off distance parameter. However, it is computationally expensive which limits its applicability to large-scale applications. Du et al., (2016) proposed a method called DPC-KNN that hybridized the idea of k nearest neighbors (k-NN) and principal component analysis (PCA) into DPC. Some other methods have employed the idea of k-NN in DPC clustering method. The main advantages and drawback of this method are briefly discussed in the introduction section. In this paper, an effective kernel method is proposed for computing local densities by taking into account the local structure of data.

The second category of improvements focused on automatically determining cluster centers. For example, Bie, Mehmood, Ruan, Sun, and Dawood (2016) proposed a method to overcome the manual selection of cluster centers. To this end, first a large number of centers are selected to form sub-clusters and then uses a heuristic method to merge sub-clusters having similar patterns. This method considers only the distance between the clusters in merging them and ignores the other properties, results reducing its performance in facing with complex data. Wang and Song (2016) employed a statistical test method to identify cluster centers instead of observing the decision graph. They have proposed a measure for computation of the local densities, which gives the better performance in distinguishing different objects. However, this method did not report desirable performance when clustering complex-manifold datasets. Moreover, by increasing the number of neighbors in density computations the performance of the method starts dropping down. In Liang and Chen (2016), a divide-and-conquer strategy is used to improve DPC method. This method can identify the number of clusters automatically without utilizing any prior knowledge from the experts. However, this method takes into account only the global structure of data which results in missing many clusters and reducing its performance compare to the original DPC. In Xu, Wang, and Deng (2016) a hierarchical based density peak method (called DenPEHC) is proposed by introducing a grid granulation framework to enable clustering high-dimensional and large-scale data. This method automatically detects all possible centers and builds a hierarchy presentation for the dataset if it is hierarchically structured in nature. Major drawbacks of these methods are their ineffective label assignment strategy. Also, they only consider the global structure of data in the computation of density values and ignore the local structure which might lead to missing some clusters.

The third category of DPC improvements has focused on developing new methods to properly assign instances to clusters. In Xu, Ju, Liang, and He (2015) a method called Manifold Density Peaks Clustering (MDPC) is proposed which first maps the geodesic space into a manifold space. In other words, it maps high dimensional datasets into lower dimensions by using a manifold-based method. Then, cluster centers are identified automatically in an easy way by dragging a rectangle manually. The main advantage of this method is using the manifold-space to reduce the dimensions of the data space but it requires a high computational cost due to computing all shortest paths that. However, in this method cluster centers are identified manually. In FKNN (Xie et al., 2016) a uniform local density metric is proposed based on the k near-

est neighbor(k-NN) method to identify cluster centers efficiently and correctly without depending on the size of the dataset. In this work, two strategies are employed to assign the remaining points to their associated clusters to avoid the error propagation. This method suffers from several shortcomings. First, it does not have any strategy to cluster outlier data and it does not make difference between border points and outliers. Another shortcoming is its high sensitivity to the number of neighbors in the k-NN method. Finally, this method requires a high computational time due to using a complex model in its process. In Yaohui, Zhengming, and Fang (2017) an adaptive density peak clustering based on k-nearest neighbors with aggregating strategy (ADPC-KNN) is proposed. In this method the idea of k-NN is used to compute the local density values, selecting initial cluster centers and merging clusters by applying an aggregation strategy. Recently, Lotfi et al. (2016) proposed a method called improved density peaks clustering (IDPC) for clustering complex data in tow steps. In the first step, the cluster centers are identified by assuming that they are surrounded by neighbors with lower local densities and should have a relatively larger distance to the other dense regions. To this aim, IDPC proposed a specific metric to rank data points considering their local densities. In the second step, for each center, a cluster core is formed by using its k nearest neighbors. To label the remaining points, IDPC uses a label assignment by considering the local density concept. This method is fast due to using a single-step label assignment mechanism. However, its results have a high sensitivity to choosing the number of the nearest neighbor. In other words, it generates high-quality results when the number of neighbors is set to a high value.

The fourth category of DPC extensions focused on applying DPC for various real-world applications. For example, in Jia, Tang, Zhu, and Li (2016) the DPC is improved on to make it suitable for hyperspectral band selection by determining the cluster centers automatically. In this method, a certain criterion is used to quantify the importance of each band then a given number of top-ranked bands is selected to form the subset. In Zhang, Chen, and Yu (2016) a method called LSH-based distributed DPC (LSHDDP) is proposed by reducing computation and communication costs of distributed DPC. This method employs locality-sensitive hashing (LSH) and parallelizes DPC with a MapReduce model. Du, Ding, and Xue (2017a) presented a method called DPC-MD, generalizing original DPC by using a specific similarity measure to deal with numerical, categorical and mixed data. They prove through experiments that their metric performs much better than Euclidean distance especially when facing with datasets having categorical attributes with hundreds or thousands of categories.

3. Proposed method

In this section, a Density Peaks Clustering method with Dynamic Label propagation method is presented (in short DPC-DLP). The proposed method consists of three main steps including (1) Identifying cluster centers, (2) forming local backbones, and (3) dynamic label propagation. In the first step, the cluster centers are identified using local densities. These centers are then used in the second step to form cluster cores. Finally, in the third step, a novel graph-based label propagation is used to spread the labels of cluster cores to the rest instances. The details of these steps are described in their corresponding sections.

3.1. Identifying cluster centers

The aim of this section is to identify a set of cluster centers. Generally, the proper cluster centers are distinguished by two properties include; (1) they are associated with higher densities compared to their neighbors; and (2) each of them is located in a dense region and is far enough from the other centers. To formulate the first property, local density is defined as the mean distance of each data point with its neighbors (Du et al., 2016). In other words, for each point, a local density ρ_i for an instance x_i is defined as follows:

$$\rho_i = \exp\left(-\left(\frac{1}{k}\sum_{x_j \in kNN(x_i)} d\left(x_i, x_j\right)^2\right)\right) \tag{1}$$

where $kNN(x_i)$ is a set of nearest neighbors of x_i and it is defined as:

$$kNN(x_i) = \left\{x_j \middle| d\left(x_i, x_j\right) \le d(x_i, x_k)\right\} \tag{2}$$

where $d(x_i,x_j)$ is the Euclidean distance between x_i and x_j and x_k is kth nearest neighbor of x_i.

The cluster center of data point x_i is chosen among points whose local density is greater than x_i and whose distance is the nearest to x_i. Besides, to measure the second property, it is required to compute the distance between each couple of instances. Then, for each instance, its minimum distance with the other instances that have higher densities is calculated. This measure is denoted by δ_i and is defined as follows:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j}\left\{d\left(x_i, x_j\right)\right\}, & if\ \exists j\ s.t.\ \rho_i < \rho_j \\ \max\limits_{j}\left\{d\left(x_i, x_j\right)\right\}, & otherwise \end{cases} \tag{3}$$

The second part of (3) ensures assigning a high δ_i value to an instance with the maximum local density. Generally, a decision graph is graphically drawn using ρ and δ values in order to identify cluster centers. This graph shows the plot of δ_i as a function of ρ_i for each instance x_i. Therefore, as interpreted from the decision graph, those of instances where associated with high distance along with high-density values are good choices for the cluster centers. Thereby, it can be ensured that identified cluster centers not only have larger local densities but also, they are far away enough from each other. These instances are also mentioned as peaks due to having higher densities compared to the others. Using the decision graph, requires interaction with a supervisor to select cluster centers.

In order to eliminate the human effort, there are a number of methods that automatically identify cluster centers (Xu et al., 2016; Liang & Chen, 2016). In this paper, a fast and simple mechanism is proposed to determine cluster centers automatically. To this end, the instances are first ranked by using their ρ and δ values and then those of top c instances are identified as cluster centers. The proposed score function is:

$$score(x_i) = \delta_i.\rho_i \tag{4}$$

Using this measure, only those instances associated with both high local density and high delta values, are assigned with high scores. Thus, this measure is a proper alternative to the decision graph while it does not need any human effort. To show the effectiveness of this measure more clearly, a sample data instances with seven clusters are presented in Fig. 2(a). The instances are plotted using their corresponding local density and delta values in Fig. 2(b). Using this graph, cluster centers are identified with their associated colors. Besides, the instances are ranked using the score function in (8), and then top seven instances are chosen as peaks. These peaks are shown in Fig. 2(a) with gray triangle instances.

3.2. Forming local backbones

The aim of this step is to assign labels to those of unlabeled points using identified cluster centers. Generally, a label propagation is used where each cluster center seeks to propagate its

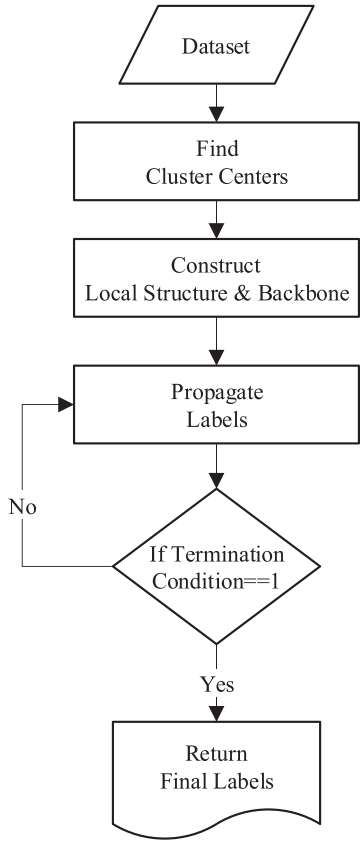


Fig. 1. Diagram of dynamic label propagation for density peaks clustering algorithm.

label to the unlabeled points. To this end, the Euclidian evaluation measure is used to compute the distance between a cluster center and a point. These types of measures exploit the global structure of data that leads to increasing the computational complexity and thus they are not suitable for proper label propagation. Due to this fact, in this paper, a label propagation method is proposed by exploiting the local structure of data. The simplest assumption is the original neighbors in a Euclidean space, which are also neighbors of an n-dimensional manifold. A manifold, in mathematics, is a topological space that locally approximates Euclidean space near each point. In other words, each point in manifold space has a neighborhood that is homeomorphic in the Euclidean space. A simple way to approximate the manifold is to construct k-NN, ϵ-NN or b-matching graphs (Jebara, Wang, & Chang, 2009). In this section, the k-NN graph is used to exploit the local structure of data in the label propagation process. Using k-NN method, a graph is constructed where each point is a node in the graph and is connected to its k-nearest neighbors. The next step is to form cluster backbones. To this aim, each center assigns its deterministic label to its neighbors as follows:

$$Label(x_i) = \begin{cases} Label\left(peak_j\right), & if\ x_i \in kNN\left(peak_j\right) \\ 0 \quad , & otherwise \end{cases} \tag{5}$$

where peaks are a set of identified cluster centers and $N(peak_j)$ is a set of neighbors for the center j. In other words, each center with its neighbors is defined as a backbone in the graph. The points that belong to a backbone generally have the highest membership degree to their corresponding clusters, while boundary points refer to those points located between a backbone and the other clusters. Using cluster backbones in the label propagation process leads to identifying both different cluster shapes and imbalanced clusters more precisely. Furthermore, exploiting several backbones leads to improving the quality and speeding up the label propagation.

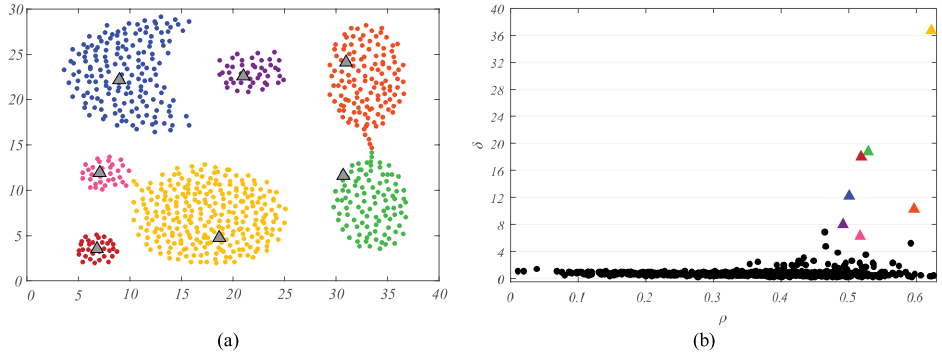
3.3. Dynamic label propagation

In this section, a graph-based label propagation method is proposed. By using this method, the labels are propagated to their neighboring until all nodes take a label. In other words, the labeled data act such as supervised sources that push out labels through unlabeled data. In the proposed label propagation, first, the data are represented by a weighted graph $G=(V,E,W)$, where V is a set of vertices corresponding to all data instances $X = \{x_i, i = 1, \ldots, n\}$, E is a set of edges and W is a nonnegative symmetric weight in the range of [0 1]. Using this representation, each $W_{i,j}$ shows a similarity value between the vertices x_i and x_j and is defined as follows:

$$W_{i,j} = exp\left(-\frac{d\left(x_i, x_j\right)^2}{\mu\sigma^2}\right) \tag{6}$$

where $d(x_i, x_j)$ is distance measure computed by some shape distance function, μ and σ are hyper-parameters. Note that the aim is to construct an affinity matrix, as a Gaussian kernel is defined to convert similarity values. Previous research has shown that the propagation results highly depend on the kernel size σ selection (Yang, Bai, Latecki, & Tu, 2008). In this paper, an adaptive kernel size is used to define σ by using the average distance from k-nearest neighbors as follows:

$$\sigma_{i,j} = b.Avg\left(\left\{knnd(x_i), knnd\left(x_j\right)\right\}\right) \tag{7}$$



(a) (b)

Fig. 2. (a) A sample data instances and (b) its corresponding decision graph.

where $Avg(\{knnd(x_i), knnd(x_j)\})$ shows the average distance of between the k-nearest neighbors of the x_i *and* x_j, also, b is an extra parameter. Note that both k and b parameters are obtained through the experiments (Yang et al., 2008).

The above weight values are defined using a fully connected graph. Generally, the full connected graph does not inherit the natural structure of data points where each data point is only similar to a few numbers of instances. Also, using a fully connected graph leads to increasing the computational complexity. Thus, a better way is to connect a graph based on k-nearest neighbors in which each node is only connected to its neighbors. In other words, the assumption is that local similarities are more reliable than global ones. This is a mild assumption widely used to model the manifold smoothness (Roweis & Saul, 2000; Tenenbaum, De Silva, & Langford, 2000). Using this concept, the weigh values are defined as follows:

$$W_{i,j}^{(kNN)} = \begin{cases} W_{i,j} & if \quad x_j \in kNN(x_i) \\ 0 & otherwise \end{cases} \tag{8}$$

Then a transition matrix is defined over the graph by normalizing the weight values as follows:

$$P_{i,j} = \frac{W_{i,j}^{(kNN)}}{\sum_{k \in V} W_{i,k}^{(kNN)}} \tag{9}$$

where P_{ij} is the probability of transit from node i to j. Note that $\sum_{j \in V} P_{i,j} = 1$ and P is asymmetric after the normalization. However, P incorporates the robust structural information about the input data space.

Then, a semi-supervised mechanism is used to propagate labels through the whole graph. To this end, the data points are divided into labeled and unlabeled ones. The labeled instances (X_l) refer to those data instances previously defined as cluster cores identified in Section 3.2 and the remaining instances are considered as unlabeled ones (X_u). Note that, each data point is associated with a label. Let define the initial label matrix (or initial cluster membership values) as $Y_0 = [Y^{(l)};\ Y^{(u)}] \in R^{n \times c}$, where n shows the number of data points, c refers to the number of clusters, $Y^{(l)}$ is the label matrix for the cluster cores, and $Y^{(u)}$ is the label matrix of unlabeled ones. A binary representation of the cluster membership values is used so that $Y_{i,k}^{(l)}$ is 1 if x_i belongs to the kth cluster, while otherwise, it is set to 0. The proposed label propagation method is a type of semi-supervised learning (SSL) method aiming to distribute labels to unlabeled instances by combining a local structure of data (i.e. a transition matrix P) with correlation between instances (i.e. YY^T) through several iterations. In each iteration, unlabeled data points update their labels as follows:

$$Y_{t+1} = F_t * Y_t \tag{10}$$

where Y_t refer to the label matrix (i.e. cluster membership values) and F_t is a fusion graph and defined as follows:

$$F_{t+1} = \begin{cases} P & if\ t = 0 \\ P\left(F_t + \alpha Y_t Y_t^T\right)P^T & if\ t > 0 \end{cases} \tag{11}$$

where YY^T shows the pairwise correlation between instances and $\alpha \in [0\ 1]$ that controls the correlation rate. Note that in each iteration, those of originally labeled instances may take a different label. Thus, their original labels needs to be reset as follows:

$$Y_{t+1}^{(l)} = Y_0^{(l)} \tag{12}$$

The pseudo-code of the proposed method is provided in Algorithm 1.

3.4. An illustrated example

In this section, an example is presented to describe more details about the proposed method. To this end, the proposed method is applied to a synthetic data is considered and the results are presented in Fig. 3. Fig. 3(a) shows the original data that contain two classes: Full-moon and crescent. In Fig. 3(a) ground-truth clusters are presented with different colors. The first step of the proposed method aims to identify cluster centers. These centers are shown in Fig. 3(b). These centers are found by employing both local (by using k-NN method) and global information (by applying density peaks method). The second step aims to create a graph by using k nearest neighbors. In other words, this graph is built by using the local structure of the data. Fig. 3(c) shows the corresponding graph while 10-NN method is applied to the data. In this step, the labels of cluster centers are propagated to their neighbors to form initial graph backbones. Finally, Fig. 3(d) shows final clusters while the proposed label propagation method is applied to graph backbones. Consider the point p in Fig. 3(c), without using the graph, this point has a minimum Euclidean distance to the first cluster (green points), while by using the graph, this point has a minimum distance from the second graph (red points). This fact shows that the proposed method can effectively cluster those of border points. In other words, the proposed method can effectively cluster data with varying shapes.

3.5. Time complexity

This section aims to present the computational complexity of the proposed method. According to Algorithm 1, the time complexity of the proposed method depends on the three main steps: (a) Identifying cluster centers (i.e. lines 2–7), (b) Forming cluster backbones (i.e. line 8) and (c) Dynamic label propagation (i.e. lines 9–15). To identify cluster centers, the distance between data points needs to be computed (line 2); the local density ρ_i (line 3) and minimum distance δ_i (line 4) for each point i are calculated, scores are computed (line 5), peaks are selected (line 6) and a distinct label is assigned to each identified center. Note that calculating the distances between data points require $O(n^2)$ computational times where n is the number of data points. Computing the local densities and minimum distances requires $O(n^2)$ time steps. Additionally, the computational time for computing the scores requires $O(n)$. To identify peaks, data points are needed to be sorted and those of top c high score points are selected with $O(nlogn)$ time steps. Consequently, the time complexity of the first step is $O(n^2 + n^2 + n + nlogn)$ which can be reduced to $O(n^2)$ mainly due to storing the distance matrix. Moreover, to form cluster backbones, a label of each identified center is propagated to each directed neighbor; thus, it requires $O(ck)$ time steps where k is the average number of neighbors and c is the number of identified peaks. Finally, the proposed dynamic label propagation method requires $O(n^2)$ time steps for generating k-NN graph structure (line 9). Moreover, it requires $O(Tkn^2)$ for dynamic label propagation (lines 11–15) where T is the total number of iterations. Consequently, the overall complexity of the proposed method is $O(n^2 + ck + Tkn^2)$. Note that $c \ll n$, $k \ll n$ *and* $T \ll n$, thus the total complexity can be reduced to $O(n^2)$.

4. Experiments

In this section, a set of experiments is conducted to evaluate the performance of the proposed DPC-DLP[1] method and compare it to well-known and state-of-the-art clustering methods in-

[1] http://github.com/amjadseyedi/DPC-DLP

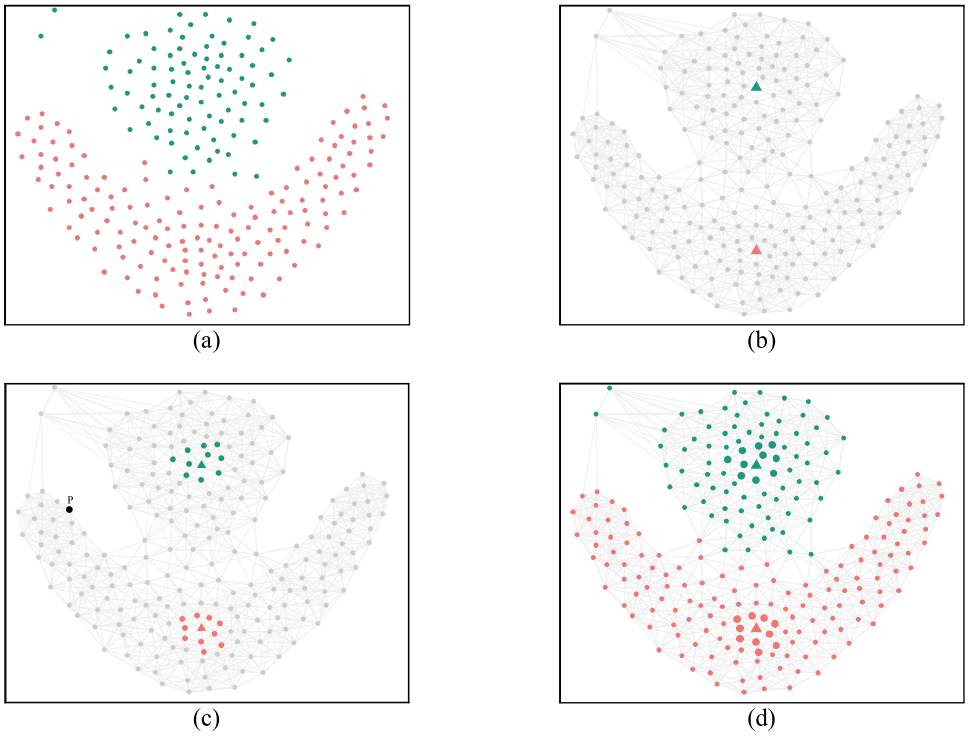


Fig. 3. Steps of DPC-DLP on simple data. (a) ground-truth. (b) Cluster centers and local structure. (c) Cluster backbones. (d) Final clusters of the DPC-DLP algorithm.

Algorithm 1 *DPC-DLP.*

Inputs:
X: matrix $n \times m$, m dimensional dataset with n instances
p: fraction of instances
T: number of iterations
Output:
Y: matrix $n \times 1$
Begin algorithm
$neighborhood\ [i,:] = \textbf{kNN}(x_i),\ \forall\ i = 1, \cdots, n$
$\rho_i = \textbf{calculateRho}\ (x_i),\ \forall\ i = 1, \cdots, n$
$\delta_i = \textbf{calculateDelta}\ (x_i),\ \forall\ i = 1, \cdots, n$
$score_i = \textbf{calculateScore}\ (\rho_i, \delta_i),\ \forall\ i = 1, \cdots, n$
$peaks = \textbf{selectPeaks}\ (score)$
Assign a distinct label to any cluster center.
Assign the label of cluster centers to their nearest neighbors to form cluster backbones. (5)
$P = \textbf{constructkTransitionMatrix}$
$F_0 = P;$
For $t = 1$ *to* T ***do***
$Y_{t+1} = F_t * Y_t$
$Y_{t+1}^{(l)} = Y_0^{(l)}$
$F_{t+1} = P(F_t * \alpha Y_t Y_t^T)P^T$
End for
End algorithm

clude: FCM (Bezdek et al., 1984), DPC-KNN (Du et al., 2016), IDPC[2] (Lotfi et al., 2016). The experiments are performed on a PC with an Intel i5-2520 M CPU, 6 G RAM, Windows 7 64bit OS, and MATLAB 2015b programming environment.

4.1. Datasets

The experiments are performed on 13 datasets: six synthetics and nine real-world datasets gathered from the UCI Machine Learning Repository (Lichman, 2013). The synthetic datasets include: Flame, CMC, Aggregation, and Compound frequently used to assess density-based data clustering methods. The Flame dataset consists of 515 two-dimensional instances distributed in two clusters. CMC[3] consists of 500 samples distributed in three clusters, two of them are moon-shaped clusters and the other one is a spherical-shaped cluster. Aggregation is a synthetic dataset that consists of 788 samples that belong to seven clusters of different sizes. Finally, Compound consists of 399 samples distributed in six clusters. In this dataset, two of the clusters are surrounded by another two clusters. The experiments were performed on nine real-world datasets taken from UCI. These datasets are commonly used in the literature to evaluate the performance of clustering algorithms. Additional details of these datasets are provided in Table 1.

[2] http://github.com/abdulrahmanlotfi/IDPC

[3] http://github.com/amjadseyedi/DPC-DLP/tree/master/dataset

Table 1
Real-world datasets used in the experiments.

Dataset	#Instances	#Features	#Classes	Application
Iris	150	4	3	Plant biology
Parkinson	195	22	2	Medicine
Sonar	208	60	2	Physical
Seeds	210	7	3	Agrophysics
Thyroid	215	5	3	Medicine
Ecoli	336	7	8	Molecular biology
WDBC	569	30	2	Medicine
Diabetes	768	8	2	Medicine
Vehicle	846	18	4	Vehicle recognition

In the experiments to assess the scalability of the proposed method facing with large-scale data, nine high-dimensional datasets are used to perform a thorough comparison. These datasets cover different research fields (e.g., signal processing, computational biology) and exhibit different properties. Additional details of these datasets are provided in Table 2.

4.2. Evaluation metrics

To report the comparison results several well-known evaluation metrics are used to assess the performance of the clustering methods. These measures include; Accuracy (ACC), Rand Index (RI), F-measure and Adjusted Rand index (ARI).

Accuracy is metric that compares the results of a clustering algorithm with truth-ground clusters (Huang, 1997). This measure is defined as follows:

$$ACC(Y,C)=\sum_{i=1}^{n}\delta(y_i,c_i)/n \tag{13}$$

where n denotes the number of samples, $y_i$and c_i are the true label and the predicted label of a sample x_i, respectively.

Rand-Index (RI) (Rand, 1971) is another metric that is also used to compare the results of a clustering algorithm with real clusters. This measure can also be used for comparing the results of two clustering algorithms. RI is defined as follows:

$$Rand\ Index(Y,C)=\frac{(a+b)}{\binom{n}{2}} \tag{14}$$

where n is the number of all samples, Y, C denote two different sets of clusters, a denotes the number of pairs of samples belonging to same clusters in both Y and C while, b denotes the number of pairs belonging to different groups (non-similar groups) in Y and C. Note that RI lies between 0 and 1. It takes the value of 1 when the two sets of clustering results are same, and it is taken by the value of 0 while both clustering results are completely different from each other.

The Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) is a measure that evaluates the clustering results by assuming a generalized hypergeometric distribution as the model of randomness. This measure is defined as follows:

$$ARI(Y,C)=\frac{\sum_{i,j}\binom{n_{ij}}{2}-\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}/\binom{n}{2}}{1/2\left[\sum_i\binom{n_{i.}}{2}+\sum_j\binom{n_{.j}}{2}\right]-\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}/\binom{n}{2}} \tag{15}$$

where Y is the clustering results and C denotes the real clustering labels. n_{ij} is the number of samples same in both clusters $c_i$and cluster y_j, $n_{i.}$and $n_{.j}$ are respectively the number of the same samples of $y_i$and c_j clusters. Note that the ARI is the corrected-for-chance version of RI used as an external criterion to compare the clustering results. RI takes its value in the range of [0 1] while ARI takes its value in the range of [−1 1].

Finally, Normalized Mutual Information (NMI) is a well-known metric to evaluate clustering methods. This measure employs information theory to quantify the differences between two clustering partitions and defined as:

$$NMI(Y,C)=\frac{MI(Y,C)}{\sqrt{H(Y)H(C)}} \tag{16}$$

where Mutual Information (MI) scores the amount of information between the two random variables and is defined as:

$$MI(Y,C)=\sum_{y_i\in Y,c_i\in C}p(y_i,c_i)log\left(\frac{p(y_i,c_i)}{p(y_i)p(c_i)}\right) \tag{17}$$

where $p(y_i,c_i)$ is the probability of belonging an instance to clusters y_i and c_i at the same time. Moreover, $p(y_i)$ and $p(c_i)$ are the probabilities that an instance to clusters y_i and c_i, respectively. Note that NMI takes its values in the range of 0 to 1. If two clusters are exactly identical, it takes the value of 1, and if two clusters are independent, then it will be equal to 0.

4.3. Results

4.3.1. Results on synthetic datasets

The aim of this section is to compare the clustering algorithms on four synthetic datasets. Fig. 4 compares the results of the proposed method with the DPC-KNN clustering algorithm. The results show that proposed method can successfully assign true labels to data instances. While in this case, DPC-KNN wrongly clusters tail points and some of the border points between clusters. For instance, the proposed DPC-DLP method only clusters two points incorrectly. Note that these two points lay on the border area between two clusters and generally it is hard to correctly classify them. However, Fig. 4(b) clearly illustrates that DPC-KNN wrongly clusters 34 points.

The experiments are repeated for CMC, Aggregation and Compound datasets and the results are respectively reported in Figs. (5-7). The results of Fig. 5(b) show that the DPC-KNN method can not correctly classify connectivity or border points. However, in this situation, the proposed method can classify all samples correctly.

Figs. (5) and (6) report similar results for Aggregation and Compound datasets. Note that the Aggregation datasets consists of seven imbalanced clusters with different sizes that two of them have connectivity points. Clustering this type of datasets is a challenging issue for data clustering algorithms. The results of Fig. 5(b) shows that the DPC-KNN method fails to correctly classify several points (60 samples) located in both border areas and also the other areas. However, in this dataset, the proposed method can classify most of the points and it only fails to correctly classify a few numbers of samples (i.e. only six points) belonging to connectivity points.

The experiments are also repeated for Compound dataset. Clustering this dataset is belong to most challenging tasks because it consists of two clusters are surrounded by the others. From Fig. 7(a), those points spread on the right side of the figure and surround the U shape cluster (the gray points) generally known as noisy points. In this case, both proposed DPC-KNN and DPC-DLP failed to classify noisy points correctly. Also, the result of Fig. 7 shows that the DPC-KNN method cannot classify the ring type cluster (i.e. yellow points on the left-bottom side of Fig. 7) the DPC-DLP method employs a local structure and label correlation in its process. Using local structure leads to preserving the main structures of clusters. Also, using the label correlation leads to correctly propagation of labels among this local structure. Consequently, from Figs. (4–7), it can be inferred that the proposed method performed better that DPC-KNN method. More specifically, the proposed method might correctly classify complex clusters that include, connectivity points (those of points locating in border areas) and ring shape clusters. However, it is failing to classify noisy samples.

Table 2
High-dimensional datasets.

Dataset	#Instances	#Features	#Classes	Application
ORL	400	1024	40	Face recognition
Drivface	606	6400	3	Head pose estimation
Coil20	1440	1024	20	Object recognition
Yeast	1483	8	10	Cellular biology
Mfeat	2000	649	10	Handwritten digit recognition
Segment	2310	19	7	Image processing
Abalone	4177	7	28	Population biology
Waveform	5000	21	3	Physical
USPS	9298	256	10	Handwritten digit recognition

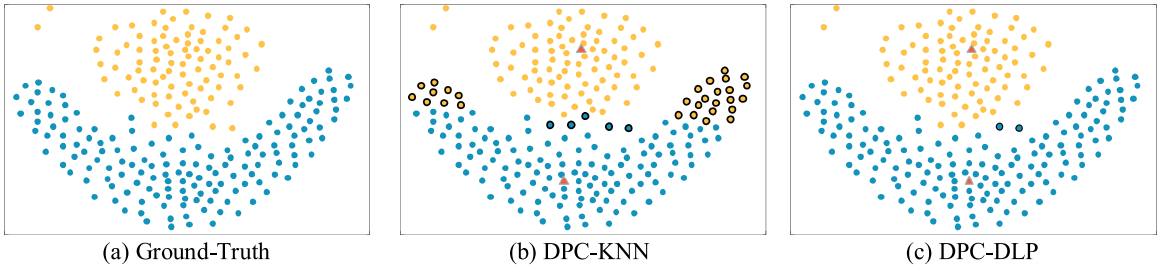


Fig. 4. (a) Ground-truth clusters and the results of, (b) DPC-KNN and (c) the proposed method (DPC-DLP) clustering methods on the *Flame* dataset.

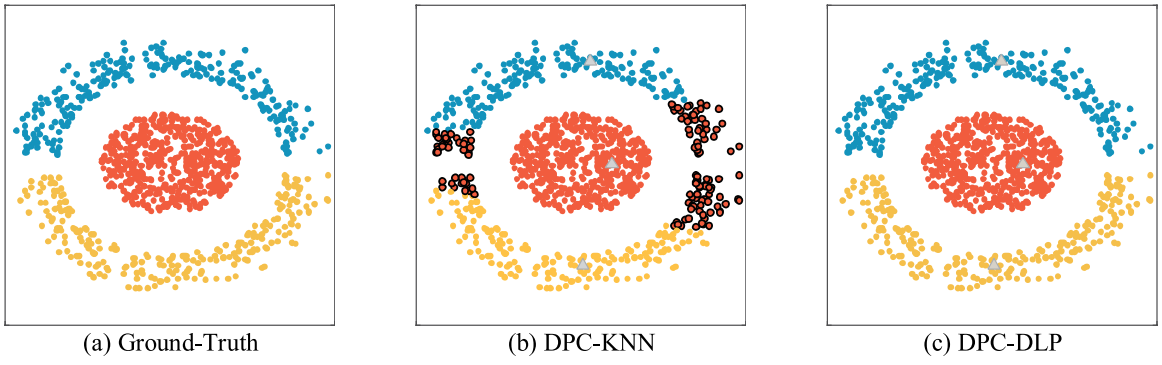


Fig. 5. (a) Ground-truth clusters and the results of (b) DPC-KNN and (c) the proposed method (DPC-DLP) clustering methods on the *CMC* dataset.

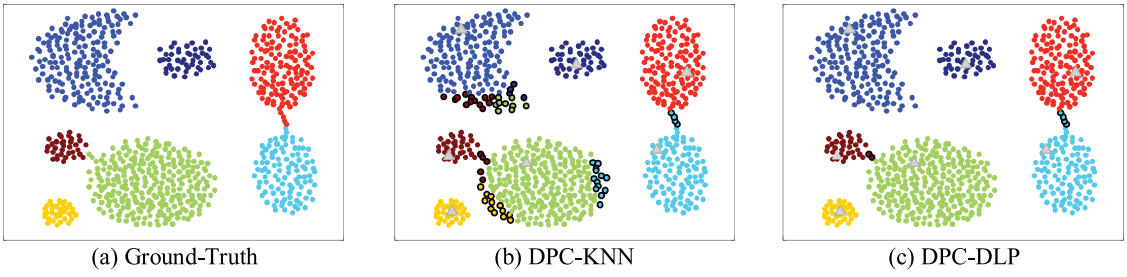


Fig. 6. (a) Ground-truth clusters and the results of (b) DPC-KNN and (c) the proposed method (DPC-DLP) clustering methods on the *Aggregation* dataset.

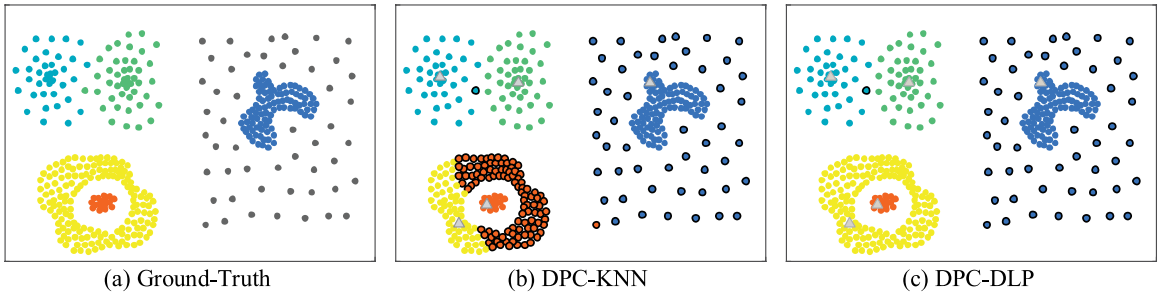


Fig. 7. (a) Ground-truth clusters and the results of (b) DPC-KNN and (c) the proposed method (DPC-DLP) clustering methods on the *Compound* dataset.

Table 3
Comparison of results in terms of F-measure. The best results are boldfaced and the second best ones are marked with stars and the number in brackets shows *p* parameter.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
Iris	0.8196	0.8668(0.10)	0.9004(0.14) *	**0.9478(0.10)**
Parkinson	0.7707*	0.6495(0.02)	0.7322(0.12)	**0.8005(0.16)**
Sonar	0.5027	0.5279(0.04)	0.5440(0.48) *	**0.5740(0.04)**
Seeds	0.8106	0.8177(0.26)	0.8396(0.08) *	**0.8495(0.04)**
Thyroid	**0.8061**	0.6369(0.005)	0.6369(0.005)	0.7743(0.005) *
Ecoli	0.6211	0.7858(0.012)	0.7858(0.012)	**0.8008(0.012)**
WDBC	0.7860*	0.6653 (0.002)	0. 7280(0.01)	**0. 8441(0.001)**
Diabetes	0.6125*	0.6016(0.001)	0.6016(0.001)	**0.6333(0.009)**
Vehicle	0.3075	0.3470(0.004)	0.3548(0.004) *	**0.3965(0.004)**

Table 4
Comparison of results in terms of Rand Index (RI). The best results are boldfaced and the second best ones are marked with stars and the number in brackets shows *p* parameter.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
Iris	0.8797	0.9124(0.1)	0.9341(0.14)*	**0.9656(0.10)**
Parkinson	0.6269	0.5929(0.02)	0.6487(0.12)*	**0.7039(0.16)**
Sonar	0.5032	0.5222(0.04)	0.5340(0.48) *	**0.5647(0.04)**
Seeds	0.8743	0.8795(0.26)	0.8939(0.08)*	**0.9004(0.04)**
Thyroid	**0.7907**	0.6064(0.005)	0.6064(0.005)	0.7011(0.005)*
Ecoli	0.8199	0.8681(0.012)	0.8681(0.012)*	**0.877(0.012)**
WDBC	0.7478*	0. 6507(0.002)	0. 6526(0.01)	**0. 8279(0.001)**
Diabetes	0.5498	0.5830(0.001)	0.5830(0.001)	**0.5963(0.009)**
Vehicle	0.6532	0.6579(0.004)*	0.6573(0.004)	**0.6807(0.004)**

4.3.2. Results on real-world datasets

Several experiments were also performed to evaluate the performance of the proposed method on the real-world datasets and the results are reported in Tables (3-5). Table 3 compares the proposed method with IDPC, DPC-KNN, and FCM in terms of F-measure evaluation metric. For those of DPC-based methods such as DPC-DLP, IDPC, and DPC-KNN, the best value of the percentage of data (i.e. p) is reported. This parameter is used to compute the local densities. In this table, the best results are boldfaced and the second-best results are indicated by the star (*) notation. The results reveal that in all cases except the Thyroid dataset, the proposed method achieved the best F1 results.

Table 4, reports the comparison results in terms of Rand Index (RI) evaluation metrics. The results are reported similar pattern in which in most cases the proposed method achieved the best results. The experiments were also performed comparing the methods based on the Adjusted Rand Index (ARI). From the above results, it is clear that the performance of the DPC-DLP is much better than the other state-of-the-art clustering methods.

4.3.3. Sensitivity analysis

The proposed method consists of several adjustable parameters include the number of neighbors in the local structure graph (k), correlation rate (α) and the percentage of data (p). In this paper, k is used to construct the local structure graph and it is set to = 10 . It means that in this graph, each node is connected to k number of its nearest neighbors. The correlation rate is used to evaluate the effect of correlation values and it is used in Eq. (11). In the experiments, this parameter us set to $\alpha = 0.01$. In DPC-based methods, the parameter p is used to choose a portion of data to compute the local densities. Several experiments were performed to sensitivity analysis the effect of this parameter on the performance of the DPC-KNN, IDPC and DPC-DLP methods. Fig. 9 tests the quality of clustering results in terms of the accuracy measure while the p parameter varies from 0.001 to 0.28. The results show that the proposed method achieved higher accuracy results in most cases compared to IDPC and DPC-KNN for different values of the p parameter. The experiments also revealed that the results of the proposed method were not impressed by the different values of the parameter. In other words, the proposed method has lower sensitivity to this parameter compared to the others.

4.4. Runtime

In this section, the methods are compared based on their time performances and their results are reported in Table 6. These results show the running times (in seconds) of the methods while performed on the real-world datasets. The results show that in most case the proposed method requires higher computational resources compared to the other methods. This is mainly due to the fact that the time required by DPC-DLP, DPC-KNN and IDPC depends on computing the local densities, the distances between points and the δ values, while the proposed method also needs additional computational resources for updating matrix fusion (Eq. (11)).

4.5. Scalability analysis

Scalability generally refers to the ability of a method to work on large-scale data. In other words, scalability means that as the size of data become larger, its performance improves correspondingly. Due to the rapid development of online data, the scalability of a method should be properly analyzed. To this aim, several experiments are performed on nine high dimensional datasets to show the scalability of the proposed method on large-scale datasets. The details of these datasets are provided in Table 2. The size of these datasets is varied from 400 to 9298 instances, their dimensions are in the range of [7 6400] and the number of classes is in range of [3 40]. Note that these datasets cover different research fields including; signal processing and computational biology. Tables (7–9) report respectively the RI, ARI, and NMI results of the proposed method compared to IDPC, DPC-KNN, and FCM methods. It can be observed from the results that, in most cases, the proposed method achieves better results. Particularly, DPC-DLP has considerable performance in the case of a large number of clusters.

4.6. DPC-DLP analysis

This section aims to analyse the behavior of the proposed method in clustering data and to show how the labels are propagated. To this end, the first four iterations of the proposed method while applying on the *Compound* dataset is shown in Fig. 8. Fig. 8(a) shows that the proposed method correctly classifies simple clusters in its first iteration (i.e. Orange and Cyan points – top-left clusters), while it may not classify ring shape clusters (i.e. bottom-left cluster). By using the local structure of data points and correlation between labels in the next iterations, the proposed method gradually propagate true labels along the ring. The details are shown in Fig. 8(b-d). It can be seen from this figure that the data points belong to the ring shape cluster finally takes the same label. In other words, in each iteration, the proposed label propagation replaces the label of misclassifying nodes by true ones using updated kernel matrix.

- Furthermore, this section aims to report the following interesting patterns extracted from the reported results: The results reveal that there is a trade-off between the execution time and the clustering performance. Tables (3–6) indicates that the proposed method achieves higher accuracy by spending a little more time compared to the other non-iterative methods. Consequently, it can be concluded that the proposed graph-based

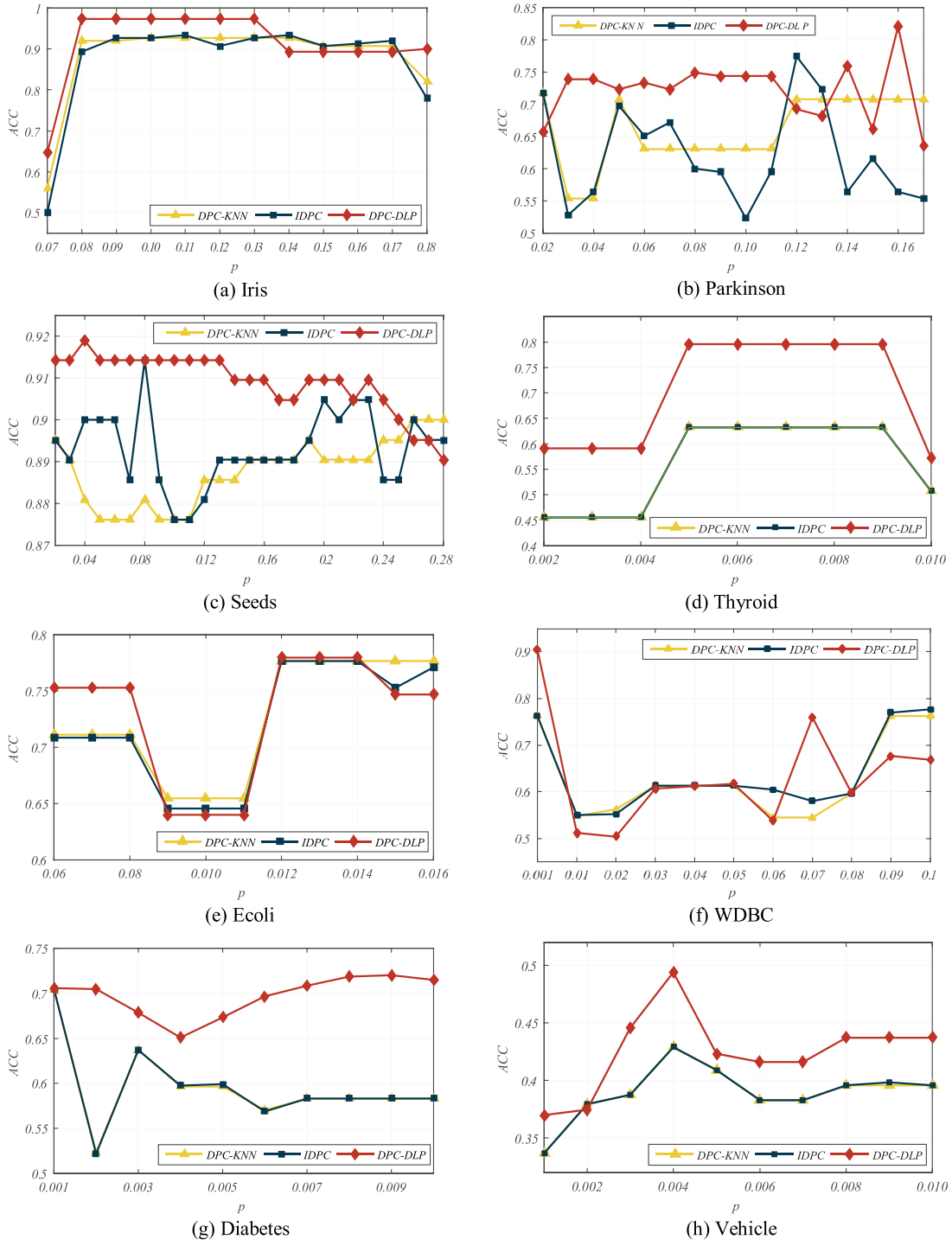


Fig. 8. Performing method on sample data. DPC-DLP achieves suitable results after four iterations.

label propagation results in proper label assignment so as to form final clusters, but it requires a little more computational time.

- Figs. (4-7) clearly reveal that the proposed method has a better performance while clustering tail points and some of the border points between clusters. Therefore the proposed method has better performance in clustering complex data. the proposed label propagation refines the assigned labels in the next iterations. In other words, while a point takes a wrong label in the earlier iterations, it will be refined in the next iterations. Note that the other DPC-based extensions operate only in a single-iteration and do not have any label refinement strategy.
- The proposed method uses only an adjustable parameter used in (1) to determine how many neighbors of a specific point are considered in the local density calculation. The value of this parameter has the main effect on the performance of the proposed method. Therefore, it is crucial to assign a proper value to this parameter before running the method. A general way to tuning the value of this parameter is performing an iterative process to test how the changes of the parameter values

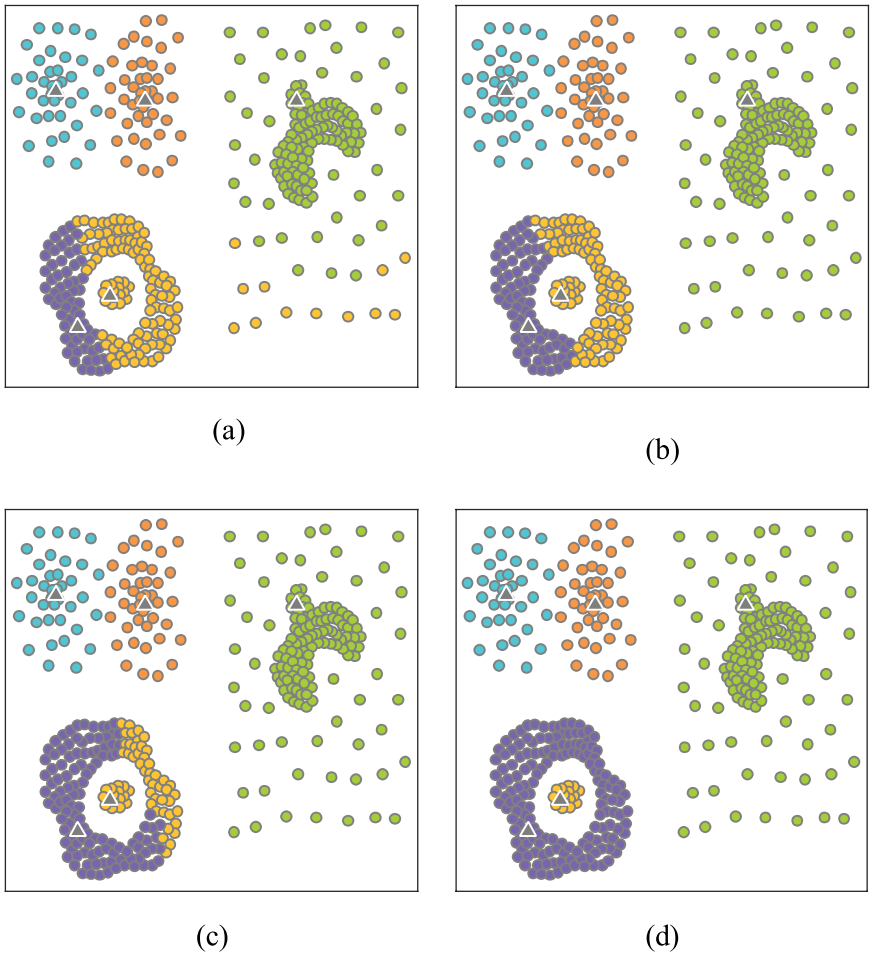


Fig. 9. Results of performing three algorithms with different p on various real-world datasets.

affect the final clustering results. Fig. 9 reports the accuracy of clustering results while this parameter varies from 0.0002 to 0.16 which is a reasonable range for all the datasets. The results report a lower sensitivity of the proposed method to this parameter compared to the others. As using graph-based label propagation methods reduces the sensitivity of the clustering methods to the number of neighbors while assigning labels.

- Tables (3–5) and Tables (7–9) reveal that the proposed method generates higher quality clustering results. The main reason is that the proposed exploits both local structure and potential label correlation in updating the graph structure. It is likely to help the label propagation process in assigning true labels.

Table 5
Comparison of results in terms of Adjusted Rand Index (ARI). The best results are boldfaced and the second best ones are marked with stars and the number in brackets shows *p* parameter.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
Iris	0.7294	0.8015(0.10)	0.8512(0.14)*	**0.9222(0.1)**
Parkinson	0	0.1713(0.02)	0.2248(0.12)*	**0.2766(0.16)**
Sonar	0.0064	0.0443(0.04)	0.0680(0.48)*	**0.1293(0.04)**
Seeds	0.7166	0.7277(0.26)	0.7604(0.08)*	**0.7751(0.04)**
Thyroid	**0.5790**	0.2077(0.005)	0.2077(0.005)	0.3800(0.005)*
Ecoli	0.5059	0.6925(0.012)	0.6925(0.012)	**0.7144(0.012)**
WDBC	0.4862*	0.3003(0.002)	0. 2830(0.01)	**0. 6526(0.001)**
Diabetes	0.0804	0.1659(0.001)	0.1659(0.001)	**0.1844(0.009)**
Vehicle	0.0762	0.1160(0.004)	0.1226(0.004)*	**0.1804(0.004)**

5. Discussion

The proposed DPC-DLP method includes three main steps. The former aims to identify cluster centers. In the second step, cluster backbones are generated and then in the third step, a graph-based label propagation method is used form final clusters by considering cluster membership values. The main property of the original density peaks clustering is using a heuristic mechanism to search through data space to identify the cluster centers. Although the proposed method keeps this property, it employs a novel density metric based on the k-Nearest Neighbors to compute the local densities. The proposed metric is not influenced by the cut-off distance and thus it is more effective than the original DPC. This property makes the proposed method to accurately evaluate the local density of any size datasets. It is worth mentioning that the proposed local density metric depends only on the k nearest neighbors, whilst the original definition relies on the cut-off distance. Note that determining the parameter k is easier than determining the cut-off distance. A general rule to define the value of k is setting it to any value lower than 2% of the total number of points in the dataset.

The most DPC extensions operate in a single-step and thus if they assign a wrong label to a point, it is propagated to the other nodes due to the lack of a label refinement strategy. To address this issue the proposed method uses membership degrees for each assigned label updated through an iterative process. In each iteration, the label propagation boosts the strong labels and weakens the weak labels in order to refine wrong labels. This refinement

Table 6
Runtime of 9 clustering algorithms in seconds on UCI datasets.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
Iris	0.0437 ± 0.0126	0.0300 ± 0.0033	0.0198 ± 0.0092	0.0118 ± 0.003
Parkinson	0.0320 ± 0.0084	0.0290 ± 0.0071	0.0365 ± 0.0070	0.0149 ± 0.0025
Sonar	0.0568 ± 0.0088	0.0324 ± 0.0037	0.0181 ± 0.0033	0.0273 ± 0.0050
Seeds	0.0538 ± 0.0161	0.0420 ± 0.0058	0.0401 ± 0.0069	0.0207 ± 0.0022
Thyroid	0.2211 ± 0.0112	0.0421 ± 0.0038	0.0480 ± 0.0079	0.0230 ± 0.0044
Ecoli	0.8804 ± 0.0493	0.1639 ± 0.0124	0.1553 ± 0.0112	0.0333 ± 0.0047
WDBC	0.0781 ± 0.0091	0.0879 ± 0.0065	0.1265 ± 0.0067	0.1928 ± 0.0152
Diabetes	0.1591 ± 0.0067	0.1107 ± 0.0082	0.1481 ± 0.0142	0.7176 ± 0.0267
Vehicle	0.8370 ± 0.1574	0.2324 ± 0.0037	0.3082 ± 0.0034	0.3439 ± 0.0242

Table 7
Comparison of results in terms of Rand Index (RI). The best results are boldfaced and the second-best ones are marked with stars and the number in brackets shows the number of neighbors.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
ORL	0.5022	**0.9627 (5)**	0.9429 (19)	0.9453 (2)*
Drivface	0.4283	0.5615 (3)	0.6939 (2)*	**0.8164 (65)**
Coil20	0.6465	**0.8547 (2)**	0.8475 (118)	0.8507 (61)*
Yeast	0.7162	0.6485 (1)	0.7220 (88)*	**0.7257 (32)**
Mfeat	**0.8877**	0.8377 (3)*	0.7507 (1)	0.7867 (1)
Segment	0.8792	0.8374 (5)	**0.8859 (51)**	0.8849 (34)*
Abalone	**0.8684**	0.8267 (1)	0.8484 (300) *	0.8302 (45)
Waveform	0.6600	0.7072 (22)	0.7754 (7) *	**0.7794 (7)**
USPS	0.5585	**0.8737 (15)**	0.8435 (1)*	0.8350 (1)

Table 8
Comparison of results in terms of Adjusted Rand Index (ARI). The best results are boldfaced and the second-best ones are marked with stars and the number in brackets shows the number of neighbors.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
ORL	0.0217	0.2865 (5)*	0.1409 (47)	**0.3096 (2)**
Drivface	0.0143	0.1233 (3)	0.1939 (2)*	**0.2962 (64)**
Coil20	0.0908	0.1921 (2)*	0.1721 (108)	**0.2303 (22)**
Yeast	0.0950	0.1072 (1)	0.1387 (38)*	**0.1875 (1)**
Mfeat	**0.4181**	0.2860 (8)	0.2517 (1)	0.3507 (1)*
Segment	0.5063	0.4896 (34)	**0.6037 (51)**	0.5857 (34)*
Abalone	0.0352	0.0651 (151)	0.0691 (2)*	**0.0733 (8)**
Waveform	0.2364	0.3542 (23)	0.5086 (7)*	**0.5164 (7)**
USPS	0.1183	0.4253 (14)*	0.2434 (1)	**0.4287 (1)**

Table 9
Comparison of results in terms of normalized mutual information (NMI). The best results are boldfaced and the second-best ones are marked with stars and the number in brackets shows the number of neighbors.

Dataset	Algorithms			
	FCM	DPC-KNN	IDPC	DPC-DLP
ORL	0.2239	0.6784 (5)*	0.5586 (41)	**0.7234(2)**
Drivface	0.0584	0.1169 (110)	0.1653 (2)*	**0.2278 (64)**
Coil20	0.3883	0.4795 (87)*	0.4379 (50)	**0.6044 (15)**
Yeast	0.1739	0.1940 (1)	0.2468 (5)*	**0.3112 (1)**
Mfeat	0.5585*	0.4996 (13)	0.4717 (1)	**0.6049 (1)**
Segment	0.6102	0.6820 (37)	**0.7157 (51)**	0.7133 (20)*
Abalone	0.1605	0.1780 (50)*	0.1778 (4)	**0.2138 (2)**
Waveform	0.3209	0.3768 (35)	0.5007 (7)*	**0.5041 (7)**
USPS	0.2305	0.5066 (5)*	0.3542 (1)	**0.6424 (1)**

strategy results in improving the quality of clustering results, especially for border region points as well as revealing complex manifolds. Similar to any iterative method, the proposed label propagation requires a higher computational cost compared to non-iterative methods. Therefore, one can propose a computational efficient label propagation method.

A major advantage of the proposed method is reducing the data space and keeping the local structure of data with the aim of identifying oblong-shaped clusters more precisely. Also, this property improves the noise robustness of the method to deal with noisy data. Moreover, combining the local structure of data with label space correlation leads to improving the discrimination ability of the proposed method in overlapped regions. Although this combination improves the quality of results, it has computational overhead which limit its applicability to big data analysis.

6. Conclusion

In this paper, a graph-based density peaks clustering method called DPC-DLP is proposed. The proposed method uses a novel dynamic graph-based label propagation strategy in its process by considering the correlation between the instances and the local structure of data. The DPC-DLP consists of three main steps. In the first step, a uniform local density metric is used to identify cluster centers. This metric employs a Gaussian kernel assigning higher density values to those of points located at the core compared to those on the border of clusters. The second step aims to cluster backbones by using the identified centers. To this end, a k-NN graph is constructed for the dataset in which each node is connected to its k nearest neighbors. However, each center and its neighbors form a cluster backbone. In the third step, a novel graph-based label propagation method is used to propagate the labels of cluster cores to the remaining instances. Note that, the cluster backbones generated on the basis of the local structure of data and can roughly retain the shape of extremely complex patterns. Furthermore, using a multi-step label assignment strategy makes the true classification of those instances that located in border and overlapped regions. In other words, the proposed method successfully addressed several issues arising from the clustering algorithm of DPC including its density metric and also the potential issue hidden in its multi-step assignment strategy. To evaluate the effectiveness of the proposed method, several extensive experiments are conducted on both synthetic and real-world datasets and the results prove the effectiveness and robustness of the proposed method.

There are several future directions for improving the proposed method. First, to obtain the similarity matrix, DPC-DLP and most of DPC extensions, calculation of the distance between all pairs is required, which limits them to be run on large datasets. To overcome this limitation, the idea of the grid can be used in the proposed method. Using this idea, the data space is divided into serval cells and the cost is reduced with the number of grid cells. The second future direction is to use kernel density estimation methods for computing local densities which do not need to define any

parameter. Although the proposed method has a high stability with different values of k, DPC-DLP, it also requires identifying of this parameter before starting the method. Third, the proposed method has some limitation on clustering complex data, especially in gene expression application. One way to enhance its ability in identifying complex clusters is to use the idea of micro-clusters. In this idea, the data is first divided into a large number of micro-clusters and then merge high correlated ones until natural clusters are discovered. Fourth, another future direction is to improve the computation complexity of the proposed method facing with large-scale datasets. A general way to this aim is to propose non-iterative label propagation methods by using the advantages of sparse graphs. Finally, one can propose an incremental DPC-DLP to cluster streaming data.

Authors contributions

Conceptualization	S.A.Seydi, A.Lotfi and P.Moradi provide the main idea of the proposed method. The idea was proof checked by N.N.Qader.
Methodology	The models, methodology and experiments were designed by S.A.Seydi, A. Lotfi and P.Moradi.
Validation	The accuracy of results was checked by P. Moradi and N.N.Qader.
Software	S.A.Seydi implemented the methods using MATLAB software and carried out the experiments.
Writing – Original Draft	The original draft and response to the reviewer document were originally prepared by A.Lotfi and P.Moradi.
Writing – Review & Editing	The manuscript and the response to the reviewer documents were edited by both P.Moradi and N.N.Qader
Visualization	The pseudo-code and Fig. 1 were provided by A. Lotfi. The other figures were provided by S.A.Seydi and A.Lotfi
Supervision	The whole project was supervised by P.Moradi.

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record , *28*, 49–60.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10*, 191–203.

Bie, R., Mehmood, R., Ruan, S., Sun, Y., & Dawood, H. (2016). Adaptive fuzzy clustering by fast search and find of density peaks. *Personal and Ubiquitous Computing, 20*, 785–793.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering, 60*, 208–221.

Castellanos-Garzón, J. A., García, C. A., Novais, P., & Díaz, F. (2013). A visual analytics framework for cluster analysis of DNA microarray data. *Expert Systems with Applications, 40*, 758–774.

de Andrades, R. K., Dorn, M., Farenzena, D. S., & Lamb, L. C. (2013). A cluster-DEE-based strategy to empower protein design. *Expert Systems with Applications, 40*, 5210–5218.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.

Ding, J., He, X., Yuan, J., & Jiang, B. (2017). Automatic clustering based on density peak detection using generalized extreme value distribution. *Soft Computing*, 1–20.

Du, M., Ding, S., & Jia, H. (2016). Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems, 99*, 135–145.

Du, M., Ding, S., & Xue, Y. (2017a). A novel density peaks clustering algorithm for mixed data. *Pattern Recognition Letters, 97*, 46–53.

Du, M., Ding, S., & Xue, Y. (2017b). A robust density peaks clustering algorithm using fuzzy neighborhood. *International Journal of Machine Learning and Cybernetics*, 1–10.

Ester, M., Kriegel, H.-P., Sander, R., & Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231).

Haghtalab, S., Xanthopoulos, P., & Madani, K. (2015). A robust unsupervised consensus control chart pattern recognition framework. *Expert Systems with Applications, 42*, 6767–6776.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st Pacific-Asia conference on knowledge discovery and data mining, (PAKDD)* (pp. 21–34).

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*, 651–666.

Jebara, T., Wang, J., & Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 441–448). ACM.

Jia, S., Tang, G., Zhu, J., & Li, Q. (2016). A novel ranking-based clustering approach for hyperspectral band selection. *IEEE Transactions on Geoscience and Remote Sensing, 54*, 88–102.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32*, 241–254.

Kozma, R., Rosa, J. L. G., & Piazentin, D. R. M. (2013). Cognitive clustering algorithm for efficient cybersecurity applications. In *The 2013 international joint conference on neural networks (IJCNN)* (pp. 1–8).

Liang, Z., & Chen, P. (2016). Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering. *Pattern Recognition Letters, 73*, 52–59.

Lichman, M. (2013). *UCI machine learning repository*. Irvine: University of California, School of Information and Computer Sciences.

Liu, P., Zhou, D., & Wu, N. (2007). VDBSCAN: Varied density based spatial clustering of applications with noise. In *2007 International conference on service systems and service management* (pp. 1–4). IEEE.

Lotfi, A., Seyedi, S. A., & Moradi, P. (2016). An improved density peaks method for data clustering. In *6th International conference on computer and knowledge engineering (ICCKE)* (pp. 263–268). IEEE.

Sharma, Lovely, & Ramya, K. (2013). A review on density based clustering algorithms for very large datasets. *International Journal of Emerging T echnology and Advanced Engineering, 3*.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability: Vol. 1* (pp. 281–297).

McGarry, K. (2013). Discovery of functional protein groups by clustering community links and integration of ontological knowledge. *Expert Systems with Applications, 40*, 5101–5112.

Mehmood, R., Zhang, G., Bie, R., Dawood, H., & Ahmad, H. (2016). Clustering by fast search and find of density peaks via heat diffusion. *Neurocomputing, 208*, 210–217.

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2*, 86–97.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2*, 559–572.

Ram, A., Jalal, S., Jalal, A. S., & Kumar, M. (2010). A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications, 3*, 1–4.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*, 846–850.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science, 344*, 1492–1496.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*, 2323–2326.

Souvenir, R., & Pless, R. (2005). Manifold clustering. In *Tenth IEEE international conference on computer vision (ICCV'05) Volume 1: 1* (pp. 648–653).

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*, 2319–2323.

Wang, B., Tu, Z., & Tsotsos, J. K. (2013). Dynamic label propagation for semi-supervised multi-class multi-label classification. In *Proceedings of the IEEE international conference on computer vision (ICCV'13)* (pp. 425–432).

Wang, G., & Song, Q. (2016). Automatic clustering via outward statistical testing on density metrics. *IEEE Transactions on Knowledge and Data Engineering, 28*, 1971–1985.

Wu, Z., & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*, 1101–1113.

Xie, J., Gao, H., Xie, W., Liu, X., & Grant, P. W. (2016). Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. *Information Sciences, 354*, 19–40.

Xu, C., & Su, Z. (2015). Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics, 31*, 1974–1980.

Xu, J., Wang, G., & Deng, W. (2016). DenPEHC: Density peak based efficient hierarchical clustering. *Information Sciences, 373*, 200–218.

Xu, X., Ester, M., Kriegel, H.-P., & Sander, J. (1998). A distribution-based clustering algorithm for mining in large spatial databases. In *14th International conference on data engineering* (pp. 324–331). IEEE.

Xu, X., Ju, Y., Liang, Y., & He, P. (2015). Manifold density peaks clustering algorithm. In *Third international conference on advanced cloud and big data* (pp. 311–318).

Yang, X., Bai, X., Latecki, L. J., & Tu, Z. (2008). Improving shape retrieval by learning graph transduction. In *European conference on computer vision* (pp. 788–801). Springer.

Yankov, D., & Keogh, E. (2006). Manifold clustering of shapes. In *Sixth international conference on data mining (ICDM'06)* (pp. 1167–1171).

Yaohui, L., Zhengming, M., & Fang, Y. (2017). Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowledge-Based Systems, 133*, 208–220.

Yu, K., Yu, S., & Tresp, V. (2006). Soft clustering on graphs. In *Advances in neural information processing systems (NIPS)* (pp. 1553–1560).

Yue, S., Wang, J., Tao, G., & Wang, H. (2010). An unsupervised grid-based approach for clustering analysis. *Science China Information Sciences, 53*, 1345–1357.

Zhang, Y., Chen, S., & Yu, G. (2016). Efficient distributed density peaks for clustering large data sets in MapReduce. *IEEE Transactions on Knowledge and Data Engineering, 28*, 3218–3230.