Neural Computing and Applications manuscript No.
(will be inserted by the editor)

McDPC: Multi-center Density Peak Clustering

Yizhang Wang · Di Wang · Xiaofeng Zhang · Wei Pang · Chunyan Miao · Ah-Hwee Tan · You Zhou

Received: date / Accepted: date

Yizhang Wang
College of Computer Science and Technology, Jilin University, Changchun, China
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China

Di Wang
Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly,
Nanyang Technological University, Singapore, Singapore
NTU-WeBank Joint Research Centre on FinTech,
Nanyang Technological University, Singapore, Singapore

Xiaofeng Zhang
Department of Computer Science, Harbin Institute of Technology (Shenzhen), Shenzhen, China

Wei Pang
School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK

Chunyan Miao
Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly,
Nanyang Technological University, Singapore, Singapore
NTU-WeBank Joint Research Centre on FinTech,
Nanyang Technological University, Singapore, Singapore
School of Computer Science and Engineering,
Nanyang Technological University, Singapore, Singapore

Ah-Hwee Tan
Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly,
Nanyang Technological University, Singapore, Singapore
School of Computer Science and Engineering,
Nanyang Technological University, Singapore, Singapore

You Zhou*
College of Computer Science and Technology, Jilin University, Changchun, China
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China
Tel.: +8613644311431

Abstract Density peak clustering (DPC) is a recently developed density-based clustering algorithm that achieves competitive performance in a non-iterative manner. DPC is capable of effectively handling clusters with single density peak (single center), i.e., based on DPC's hypothesis, one and only one data point is chosen as the center of any cluster. However, DPC may fail to identify clusters with multiple density peaks (multi-centers) and may not be able to identify natural clusters whose centers have relatively lower local density. To address these limitations, we propose a novel clustering algorithm based on a hierarchical approach, named Multi-center Density Peak Clustering (McDPC). Firstly, based on a widely adopted hypothesis that the potential cluster centers are relatively far away from each other. McDPC obtains centers of the initial micro-clusters (named representative data points) whose minimum distance to the other higher-density data points are relatively larger. Secondly, the representative data points are autonomously categorized into different density levels. Finally, McDPC deals with micro-clusters at each level and if necessary, merges the micro-clusters at a specific level into one cluster to identify multi-center clusters. To evaluate the effectiveness of our proposed McDPC algorithm, we conduct experiments on both synthetic and real-world datasets and benchmark the performance of McDPC against other state-of-the-art clustering algorithms. We also apply McDPC to perform image segmentation and facial recognition to further demonstrate its capability in dealing with real-world applications. The experimental results show that our method achieves promising performance.

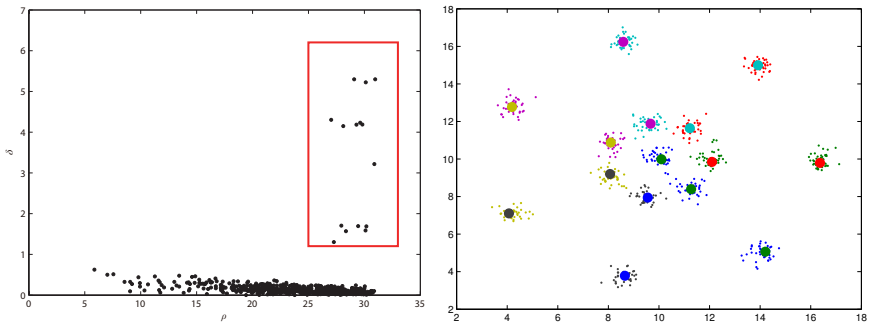
Keywords Density peak clustering · multi-center cluster · image segmentation

1 Introduction

Density-based clustering has been widely adopted in the literature [1,2,3]. Moreover, it recently attracted an increasing amount of attention in data mining and pattern recognition [4,5]. In the family of density-based clustering methods, DBSCAN (Density Based Spatial Clustering of Applications with Noise) is probably the most well-known one. DBSCAN works well in the identification of clusters of arbitrary shapes [6,7]. In 2014, another well-known density-based clustering method named DPC was proposed [8]. The hypothesis adopted by DPC is simple but effective that cluster centers should have high density and be far away from one another [9]. In short, DPC provides a powerful tool for analyzing complex intrinsic data structures, i.e., local density.

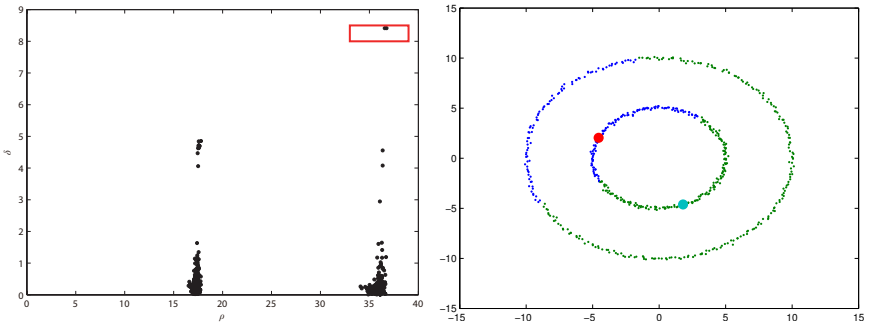
In DPC, users are required to choose cluster centers manually on the two-dimensional space (known as decision graph) defined by the computed local density (known as parameter ρ in DPC) and the minimum distance between the underlying data point and other higher-density data points (known as parameter δ in DPC). As a rule of thumb, the data points in the upper right corner of a decision graph (see Section 2 for more technical details) should be chosen as cluster centers (see Fig. 1). As shown in Fig. 1b, DPC performs well because each natural cluster has only one single center (see Fig. 1a, each data point in the red rectangle is chosen

*Corresponding Author
E-mail: zyou@jlu.edu.cn



(a) Decision graph of dataset R15 [10] generated by DPC

(b) DPC-R15 ($pct = 5$)



(c) Decision graph of dataset 2cirlce [11] generated by DPC

(d) DPC-2cirlce ($pct = 5$)

Fig. 1: The decision graphs and clustering results generated by DPC on two synthetic datasets. Data points in the red rectangle in Figs. 1a and 1c are chosen as cluster centers, the enlarged points represent the selected centers in Figs. 1b and 1d. Parameter *pct* is predefined by the user (see Section 2 for more technical details).

as a center). However, DPC obtains inferior results as shown in Fig. 1d because (i) each natural cluster has multiple centers (multi-centers) and (ii) DPC ignores the outer circle cluster whose centers are all located in the lower local density region.

To address the aforementioned issues, we propose a novel clustering method named Multi-center Density Peak Clustering (McDPC). In the initial clustering stage, McDPC first considers the minimum distance between each individual data point and other higher-density data points to obtain all the micro-clusters of a dataset. Then based on the generated decision graph, all data points can be autonomously categorized into different levels according to their local densities. At certain density levels (except the level with the highest local density), McDPC merges all the micro-clusters in their respective density level into one cluster so as to prevent natural clusters with lower densities being left out. At the highest local density level, if the minimum distance between the representative data points and

other higher-density data points is relatively small (controlled by parameter δ in DPC), these representative data points are considered as being significantly close to each other. Therefore, all the micro-clusters at this level should be merged into one cluster to correctly identify a single cluster with multi-centers.

After introducing the technical details of our proposed McDPC algorithm, we further evaluate its performance on six synthetic datasets and six real-world ones. Moreover, we also apply McDPC to perform image segmentation and facial recognition tasks. Experimental results show that McDPC performs better than the other state-of-the-art clustering methods.

The rest of this paper is organized as follows. In Section 2, we introduce the dynamics of DPC and briefly review other related work. In Section 3, we present the proposed McDPC algorithm with technical details. In Section 4, we report the experimental results on publicly available datasets with discussions. Finally, in Section 5, we conclude the paper and propose future work.

2 Related Work

Density peak clustering (DPC) algorithm follows the following two hypotheses [8]:

(i) Cluster centers are surrounded by data points of lower density.
(ii) Cluster centers are far away from each other.

Specifically, for a dataset $D = (x_1, x_2, ..., x_i, ..., x_N)$, the similarity matrix S is defined as $S(i,j) = \|x_i - x_j\|$, $i > j$, where $\|\cdot\|$ measures the distance between data points x_i and x_j (in this paper, Euclidean distance is adopted by all clustering methods). Subsequently, S is sorted in descending order to form a set: $s = \{s_1, ..., s_M\}$, where M denotes the length of vector s (the value of M is derived from N, see more technical details presented in [8]). The local density ρ_i of data point x_i is then defined as follows:

$$\rho_i = \sum_j e^{-(\frac{\|x_i - x_j\|}{d_c})^2}, \tag{1}$$

$$d_c = s_{rnd(pct\%*M)}, \tag{2}$$

where d_c denotes the cutoff distance, pct denotes a user-specified parameter that decides which element of s is selected for the assignment of d_c, and $rnd(o)$ returns the nearest integer value of o.

Moreover, δ_i, which denotes the minimum distance between data point x_i and other higher-density data points, is defined as follows:

$$\delta_i = \min_{j:\rho_j > \rho_i} S(i,j). \tag{3}$$

The values of ρ_i and δ_i are then used to generate a two-dimensional decision graph, which helps the user to manually select cluster centers. As a rule of thumb, data points with relatively higher ρ and δ values should be selected as cluster centers. Subsequently, the remaining data points are assigned to the respective nearest higher density cluster centers.

To the best of our knowledge, there are two kinds of strategies proposed in the literature to extend the DPC algorithm. Firstly, the performance of DPC is greatly

affected by the threshold parameter, i.e., local density d_c, which needs to be fine tuned with respect to different datasets. To deal with the parameter setting issue, Ding *et al.* developed an automatic DPC algorithm based on generalized extreme value distribution [12]. Liu *et al.* incorporated the idea of K-nearest neighbors to compute the global parameter d_c to achieve adaptive clustering [13]. Secondly, when the cluster centers are wrongly chosen, other data points will be subsequently misassigned. To better identify the cluster centers, Xie *et al.* employed fuzzy weighted K-nearest neighbors to make data points allocation more robust [14]. Du *et al.* improved the clustering results (especially for high-dimensional data sets) using K-nearest neighbors and principal component analysis [15]. Xu *et al.* proposed a grid granulation framework to enable the clustering of large-scale and high-dimensional datasets with a hierarchical strategy [16]. Wang *et al.* applied an improved density peak based clustering method to social circle detection [17]. By only considering the density within a local neighborhood region, Parmar *et al.* propose a residual error-based density peak clustering algorithm to better identify overlapping clusters [18]. In addition, density peak clustering algorithm has been recently applied in many research fields. Lin *et al.* used it to diagnose the fault in photovoltaic array [19]. Tu *et al.* adopted it for hyperspectral anomaly detection [20]. Ding *et al.* proposed an entropy-based density peaks clustering algorithm for mixed type data [21]. Guo *et al.* used an improved density peaks clustering algorithm to analysis drug-target data [22].

In this paper, we delineate our novel clustering technique, which adopts a different approach from the afore-reviewed methods. Specifically, if a natural cluster has multiple centers or its center has lower local density, DPC may fail to obtain satisfactory results. Our proposed McDPC addresses the aforementioned limitations by analyzing the decision graph in a novel autonomous manner. The technical details of McDPC are introduced in the following section.

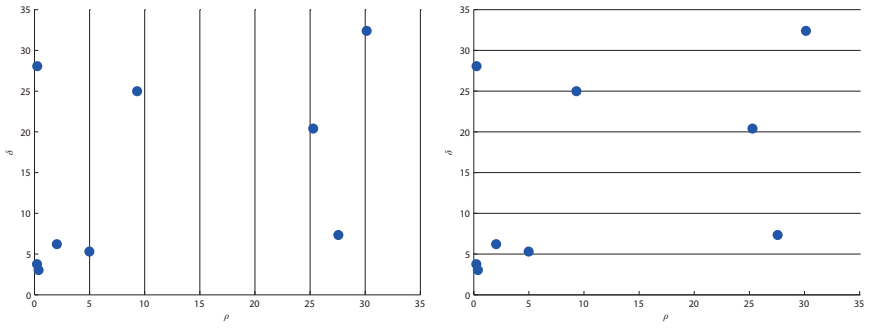
3 Multi-center Density Peak Clustering

In this section, we present our proposed clustering method, named Multi-center Density Peak Clustering (McDPC). First, we introduce the necessary definitions of McDPC. We then introduce the technical details on the two major procedures of McDPC, namely (i) autonomously finding all the micro-clusters, whose centers are named representative data points and (ii) properly grouping these micro-clusters to obtain the final clustering results.

3.1 Preliminary Definitions

After obtaining the decision graph (following the same procedures used in DPC), we further use an interval length θ to equally divide the horizontal axis (which represents local density ρ_i of data point x_i) of the derived decision graph into X bins. We name this horizontal division process ρ-cut. Moreover, we use PX to denote the obtained horizontal bins that $PX = \{PX_1, \cdots, PX_X\}$. Specifically, the value of X is computed as follows:

$$X = \left\lfloor \frac{max(\rho) - min(\rho)}{\theta} \right\rfloor + 1, \tag{4}$$



(a) The ρ-cut of a decision graph (b) The δ-cut of a decision graph

Fig. 2: Illustrations of ρ-cut and δ-cut using a randomly generated dataset. (a) Identification of X bins. (b) Identification of Y bins.

where $max(\rho)$ denotes the maximum local density found in the underlying dataset, $min(\rho)$ denotes the minimum local density found in the underlying dataset, and θ denotes the user-defined value for interval length. An example is shown in Fig. 2a, to illustrate the relationship between θ and X. If we set θ to 5, then according to (4), X equals to 7.

Similarly, we use an interval length γ to equally divide the vertical axis (which represents δ_i of data point x_i) of the derived decision graph into Y bins. We name this vertical division process δ-cut. Similarly, we have $PY = \{PY_1, \cdots, PY_Y\}$. The value of Y is computed as follows:

$$Y = \left\lfloor \frac{max(\delta) - min(\delta)}{\gamma} \right\rfloor + 1, \tag{5}$$

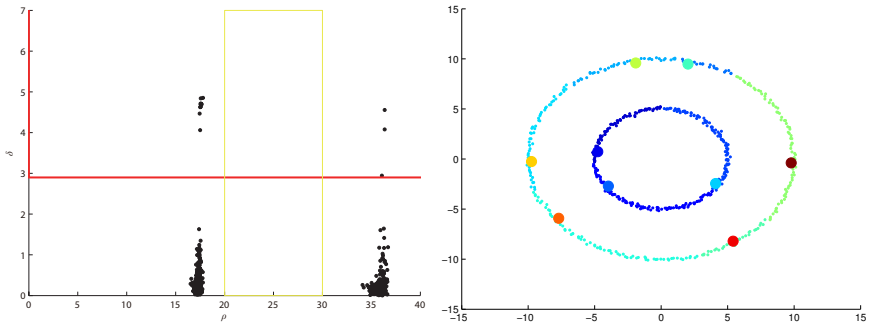
where $max(\delta)$ denotes the maximum δ value found in the underlying dataset and $min(\delta)$ denotes the minimum δ value found in the underlying dataset. An example is shown in Fig. 2b, to illustrate the relation between δ and Y. If we set γ to 5, then according to (5), Y equals to 7.

Based on the introductions of ρ-cut and δ-cut, we further present the following definitions:

(i) *X-break*: A region comprising two or more contiguous bins PX_i in the x-axis with the absence of any data point from within is named an X-break. As shown in Fig. 2a, the $[10, 25]_\rho$ region is found as an X-break. Note that there may exist more than one X-break in a decision graph.

(ii) *Density level*: The X-break (if exist) naturally separates data points into different regions along the horizontal axis. We assign each region with a density level equals to the averaged density among all the data points within the corresponding region. As shown in Fig. 2a, the averaged density $\bar{\rho}_{[0,10]} = 2.86$ in the $[0, 10]_\rho$ region and $\bar{\rho}_{[25,30]} = 27.66$ in the $[25, 30]_\rho$ region.

(iii) *Maximum density level*: The density level of the right most region in a decision graph is defined as the maximum density level. As shown in Fig. 2a, the maximum density level is 27.66, which is the density level of the rightmost $[25, 30]_\rho$ region.



(a) Representative data points identified in dataset 2circles ($pct = 5, \lambda = 2.9$)

(b) Correspondingly generated micro-clusters

Fig. 3: Illustrations on the identification of representative data points and the corresponding micro-clusters.

(iv) *Y-break*: Similar to X-break, a region comprising two or more contiguous bins PY_i in the y-axis with the absence of any data point from within is named a Y-break. As shown in Fig. 2b, the $[10, 20]_\delta$ region is found as a Y-break. Note that there may exist more than one Y-break in a decision graph.

(v) *δ level*: Similar to density level, the Y-break (if exist) naturally separates data points into different regions along the vertical axis. We assign each region with a δ level equals to the averaged δ value among all the data points within the corresponding region. As shown in Fig. 2b, the averaged δ $\bar{\delta}_{[0,10]} = 4.59$ in the $[0, 10]_\delta$ region and $\bar{\delta}_{[20,35]} = 26.28$ in the $[20, 35]_\delta$ region.

With the help of these definitions, we present the detailed dynamics of McDPC in the following subsection.

3.2 Identification of Micro-clusters and Their Representative Data Points

Data points with higher δ values may be far away from others in the derived decision graph and thus, have high possibility to be selected as centers. Therefore, we name the data points with higher δ value *representative data points* and tentatively use them to obtain all the micro-clusters. We use a straightforward method to identify representative data points, which can be described as follows. Data point x_i is identified as a representative point if and only if $\delta_i \geq \lambda$, where λ is a user-defined threshold. For example, as shown in Fig. 3a, if λ is set to 2.9, all data points fall above the red horizontal line are identified as representative data points. Once representative data points are obtained, each remaining data point is assigned to the same cluster of its nearest neighbor of higher density, which has already obtained its corresponding cluster index.

3.3 Clustering based on Different Density Levels

As aforementioned in Section 1 that DPC provides a powerful tool for analyzing complex intrinsic data structures, i.e., local density. In McDPC, we also use local density to better identify the potential cluster centers. Specifically, the different density levels are autonomously identified by applying ρ-cut. Subsequently, the representative data points are identified, which may reside in different density levels, and are further used to obtain all the micro-clusters. At this stage, the problem of grouping all data points in a underlying dataset into different clusters has been transformed into grouping the micro-clusters. To this end, we propose a micro-cluster grouping method, which autonomously perform the corresponding actions based on the density levels. The autonomous micro-cluster grouping method is introduced as follows:

(i) Because representative data points at a lower density level (not at the maximum density level) have similar and low local densities, the corresponding micro-clusters should be merged into one cluster.
(ii) At the maximum density level, if a natural cluster has multiple centers, it may be easily separated into multiple clusters. Thus, we use δ-cut to autonomously identify clusters with multi-centers.
  (ii.a) If all the identified representative data points are at the same δ level, they all should be selected as cluster centers, because they have similar and higher δ and ρ values.
  (ii.b) On the contrary, if they spread across different δ levels, the corresponding micro-clusters should be merged with others that all the micro-clusters in the same region are merged into one single cluster. The intuition of such a grouping approach is that lower δ values actually mean that the minimum distance between the underlying representative data points and others with higher density is small, i.e., the corresponding micro-clusters are thus not distinguishable.

To further illustrate the scenario described in (ii.b), we use dataset 2circles as an example (see Fig. 3a). In this example, θ is set to 5, then the $[20, 30]_\rho$ region (represented as the yellow rectangle in Fig. 3a) is found as the X-break. As such, the representative data points are divided into two density levels. According to (ii.b), micro-clusters in each local density level are merged into one cluster, respectively. In the end, McDPC generates two clusters for dataset 2circles (see Fig. 3b).

As a form of compensation to autonomy, McDPC employs relatively more numbers of parameters as compared to DPC. In total, McDPC takes four parameters, namely γ, θ, λ and pct. Parameter γ and θ are used to perform ρ-cut and δ-cut, respectively, λ is the threshold used to identify micro-clusters, and pct is used to generate the decision graph. The overall procedures of McDPC are shown in Algorithm 1.

4 Performance Evaluations on Synthetic and Real-World Datasets

In this section, we use both synthetic and real-world datasets to evaluate the effectiveness of McDPC. The performance of McDPC is compared against three

Algorithm 1 Multi-center density peak clustering

Input: input dataset D, parameters pct, θ, γ and λ
Output: assigned cluster indices

compute ρ and δ values of each data point;
obtain representative data points (RDP) and corresponding micro-clusters (see Section 3.2);
perform ρ-cut to distinguish density levels and subsequently identify X-break(s);
$m \leftarrow$ number of density levels;
for $j = 1 : m$ **do**
    if $j == m$ **then**
        $MRDP \leftarrow$ RDP in maximum density level;
        perform δ-cut to distinguish δ levels and subsequently identify Y-break;
        if $MRDP$ spread across different δ levels **then**
            all micro-clusters in the jth density level are merged into one cluster;
        else
            each micro-cluster in the jth density level is retained as an individual cluster;
        end if
    else
        corresponding micro-clusters in the jth density level are merged into one cluster;
    end if
end for

Table 1: Property of both synthetic and real-world datasets

Dataset	*#Samples*	*#Dimensions*	*#Natural Clusters*
Jain	373	2	2
Aggregation	788	2	7
2circles	600	2	2
2circles_noise	634	2	3
Pathbased	300	2	2
Halfkernel	1000	2	2
Pima	768	8	2
Ecoli	336	7	3
Sonar	208	60	2
Iris	150	4	3
Seeds	210	7	3
Wine	178	13	3

state-of-the-art clustering algorithms, namely clustering by find of density peak (DPC) [8], Affinity Propagation (AP) [23, 24] and Density Based Spatial Clustering of Applications with Noise (DBSCAN) [6]. We adopt F-measure (FM) [25], adjusted rand index (ARI) [26], rand index (RI) [26] normalized mutual information (NMI) [27] as the performance evaluation metrics. Values of metrics range are from -1.0 to 1.0, wherein a larger value denotes better performance. All the six synthetic datasets used in this paper (listed in the first six rows in Table 1) can be downloaded from our online repository[1]. Moreover, all the six real-world datasets used in this paper (listed in the last six rows in Table 1) can be downloaded from UCI machine learning repository[2]. The property of all the twelve datasets is summarized in Table 1.

[1] URL: https://github.com/mlyizhang/Multi-center-DPC.git
[2] URL: http://archive.ics.uci.edu/ml/index.php

Table 2: Performance benchmark on synthetic datasets

Datasets	Algorithm	*Par*	*FM*	*ARI*	*RI*	*NMI*
Jain	AP	50	0.67	0.27	0.63	0.34
	DBSCAN	2.9/20	**1.00**	**1.00**	**1.00**	**1.00**
	DPC	40	**1.00**	**1.00**	**1.00**	**1.00**
	McDPC	0.1/2/3.35/2	**1.00**	**1.00**	**1.00**	**1.00**
Aggregation	AP	30	0.79	0.73	0.91	0.82
	DBSCAN	2.5/4	0.86	0.81	0.93	0.89
	DPC	4	**1.00**	**1.00**	**1.00**	**1.00**
	McDPC	0.5/0.1/2.9/4	**1.00**	**1.00**	**1.00**	**1.00**
2circles	AP	20	0.43	0.12	0.56	0.17
	DBSCAN	3/3	**1.00**	**1.00**	**1.00**	**1.00**
	DPC	20	0.67	0.01	0.51	0.09
	McDPC	0.2/2/3/2	**1.00**	**1.00**	**1.00**	**1.00**
2circles_noise	AP	20	0.30	-0.00	0.52	0.14
	DBSCAN	0.85/5	0.99	0.99	0.99	0.98
	DPC	5	0.61	0.09	0.53	0.17
	McDPC	0.1/2/3/2	**1.00**	**1.00**	**1.00**	**1.00**
Pathbased	AP	30	0.65	0.40	0.70	0.49
	DBSCAN	2/5	0.73	0.54	0.76	0.70
	DPC	5	0.64	0.41	0.71	0.50
	McDPC	0.12/0.8/3.5/0.5	**1.00**	**1.00**	**1.00**	**1.00**
Halfkernel	AP	21	0.42	0.01	0.51	0.07
	DBSCAN	3/4	**1.00**	**1.00**	**1.00**	**1.00**
	DPC	10	0.72	0.40	0.70	0.44
	McDPC	0.5/5/7/4	**1.00**	**1.00**	**1.00**	**1.00**

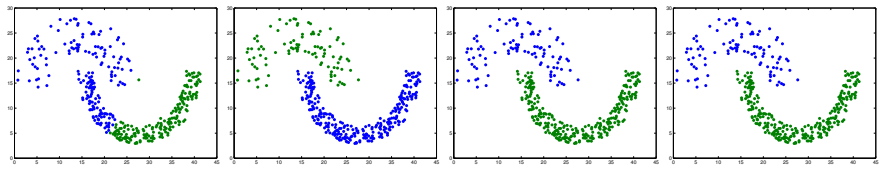
4.1 Experimental Results on Synthetic Datasets

We present the experimental results on the six synthetic datasets in Table 2 and visualize the clustering results in Figs. 4-9, respectively. The parameter values of all clustering algorithms have been tuned by trial-and-error and the best results are reported. As shown in Table 2, McDPC achieves perfect results on all six synthetic datasets.

Please note that in Table 2, *Par* denotes the parameter(s) being used by the respective clustering algorithm. Specifically, AP uses *preference* (median value of similarity matrix), DBSCAN uses *Eps* (radius of a cluster) and *MinPts* (minimum number of data points in a cluster), DPC uses *pct* (see Section 2), and McDPC uses γ, θ, λ and *pct* (see Section 3). The parameter values used by each individual algorithm to obtain the corresponding results exactly follow the sequence of parameters afore-introduced.

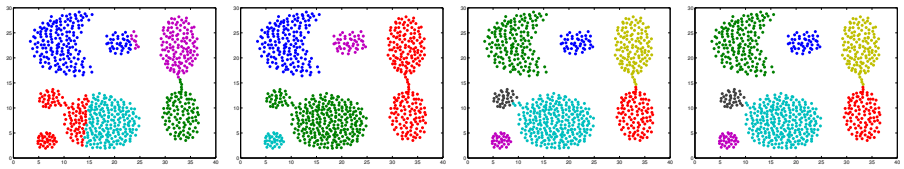
As shown in Figs. 4-9, AP does not perform well on non-convex clusters. For clusters with single centers, both DPC and McDPC perform well (see Figs. 4 and 5). More encouragingly, McDPC works well on datasets wherein natural clusters have multiple centers. Furthermore, McDPC correctly identifies clusters in the lower local density region (see Figs. 6-9). In contrast, DPC fails to identify such clusters.

To further test the performance of McDPC, in the following subsection, we present its clustering results on real-world datasets.



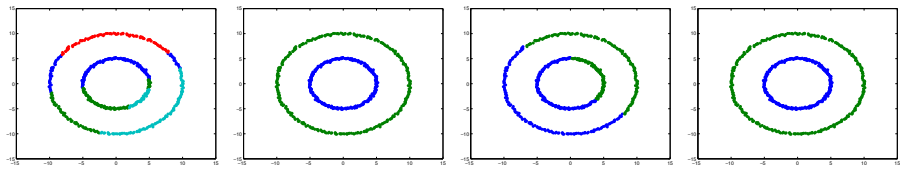
(a) AP (b) DBSCAN (c) DPC (d) McDPC

Fig. 4: Clustering results on dataset Jain.



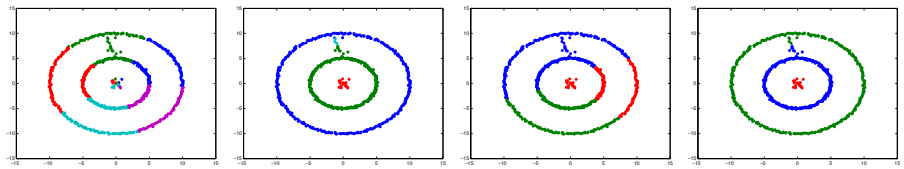
(a) AP (b) DBSCAN (c) DPC (d) McDPC

Fig. 5: Clustering results on dataset Aggregation.



(a) AP (b) DBSCAN (c) DPC (d) McDPC

Fig. 6: Clustering results on dataset 2circles.

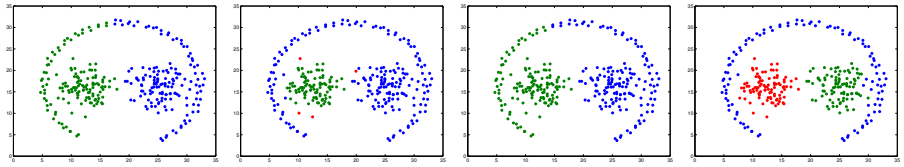


(a) AP (b) DBSCAN (c) DPC (d) McDPC

Fig. 7: Clustering results on dataset 2circles_noise.

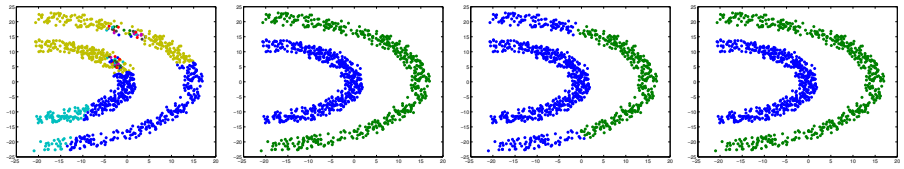
4.2 Experimental Results on Real-World datasets

We present the experimental results on the six real-world datasets in Table 3. As shown, FM scores obtained by McDPC are always equal to or greater than DPC for all datasets. Due to the variety of different datasets, no single clustering algorithm outperforms the others in all six datasets. McDPC gets the best performance in



(a) AP (b) DBSCAN (c) DPC (d) McDPC

Fig. 8: Clustering results on dataset Pathbased.



(a) AP (b) DBSCAN (c) DPC (d) McDPC

Fig. 9: Clustering results on dataset Halfkernel.

Table 3: Performance benchmark on real-world datasets

Datasets	Algorithm	*Par*	*FM*	*ARI*	*RI*	*NMI*
Pima	AP	50	0.59	**0.14**	**0.57**	**0.09**
	DBSCAN	1.4/4	**0.74**	0.00	0.55	0.00
	DPC	3	**0.74**	0.00	0.55	0.00
	McDPC	0.02/1/0.6/5	**0.74**	0.00	0.55	0.00
Ecoli	AP	21	0.80	0.71	0.87	0.67
	DBSCAN	0.1/1	0.56	0.41	0.81	0.51
	DPC	2	0.79	0.68	0.86	0.64
	McDPC	0.008/0.002/0.3/0.3	**0.83**	**0.75**	**0.90**	**0.71**
Sonar	AP	20	0.52	-0.00	**0.50**	**0.00**
	DBSCAN	9.2/4	**0.71**	**0.00**	**0.50**	**0.00**
	DPC	23	0.66	**0.00**	**0.50**	**0.00**
	McDPC	0.01/3/1.5/0.2	**0.71**	**0.00**	**0.50**	**0.00**
Iris	AP	1	0.52	0.33	0.76	0.59
	DBSCAN	1/2	0.77	0.57	0.78	0.73
	DPC	0.8	0.67	0.43	0.43	0.63
	McDPC	0.03/0.5/1/0.8	**0.92**	**0.89**	**0.95**	**0.87**
Seeds	AP	25	**0.80**	**0.70**	**0.87**	0.67
	DBSCAN	4/5	0.57	0.00	0.33	0.00
	DPC	2	**0.80**	**0.70**	**0.87**	**0.70**
	McDPC	0.2/0.01/2/2	**0.80**	**0.70**	**0.87**	**0.70**
Wine	AP	25	0.58	0.36	0.71	**0.43**
	DBSCAN	100/2	0.58	0.00	0.34	0.00
	DPC	0.1	**0.60**	**0.39**	**0.72**	**0.43**
	McDPC	0.01/0.1/250/0.2	**0.60**	**0.39**	**0.72**	**0.43**

most cases, which suggests that McDPC is more generic to deal with datasets of different intrinsic structure.

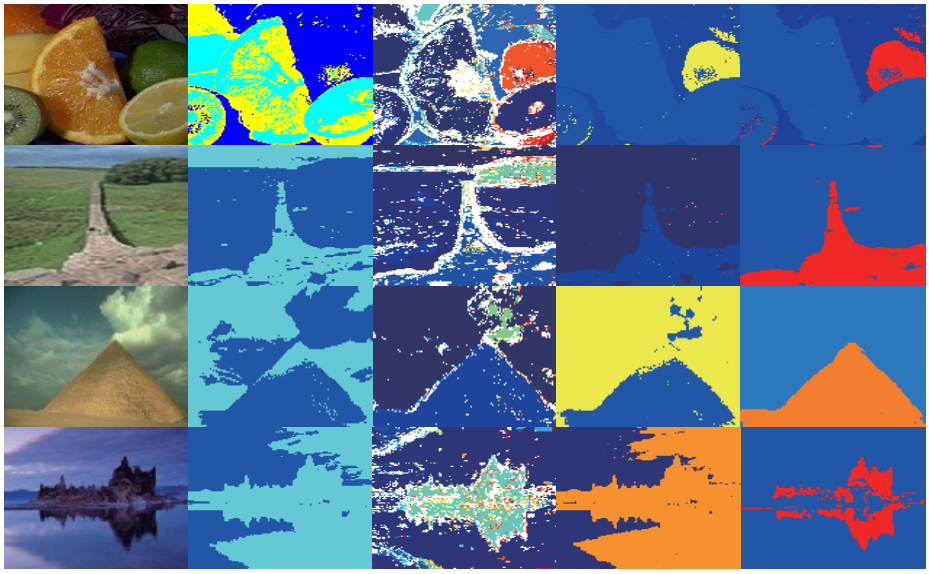


Fig. 10: From left column to right column: the original image, the segmentation results obtained by AP, DBSCAN, DPC and McDPC, respectively.

4.3 Experimental Results on Color Image Segmentation

Image segmentation has always been one of the most important research topics in the field of machine vision or pattern recognition in general [28,29,30,31]. To further compare the performance of McDPC with others, we select the image of Fruits from University of Eastern Finland Test Image Datasets[3], and the images of Road, Hill and Water from Berkeley Segmentation Dataset and Benchmark[4]. The size of all image is 128 $\times$ 128 and RGB values of every pixel are used for clustering. The segmentation results are shown in Fig. 10. For color images Fruits and Road, DPC and McDPC get the same segmentation results, which are better than those of AP and DBSCAN. For color images Hill and Water, McDPC obtains the best performance among all. It is encouraging to see that McDPC outperforms the other benchmarking models in this image segmentation experiment.

4.4 Experimental Results on Facial Recognition

In this subsection, we apply McDPC to perform facial recognition. Specifically, the Olivetti Face dataset[5] is used. This dataset is a well recognised benchmark for facial recognition tasks and it consists of 400 facial images (all on greyscale, 112 $\times$ 92 pixels). Each person has 10 images and each image has 10,304 features. Adopting the same experimental setup introduced in [8], we use the first 100

[3] URL: http://cs.uef.fi/sipu/images/
[4] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/
[5] URL: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Table 4: Performance benchmark on facial recognition

	AP (2.1)	DBSCAN (0.85/2)	DPC (d_c=0.07 [8])	McDPC (1/0.001/1.05/1)
FM	0.6744	0.6352	0.6651	**0.7643**
ARI	0.6424	0.5918	0.6023	**0.7405**
RI	0.9420	0.9263	0.9127	**0.9596**
NMI	0.7949	0.7979	0.8264	**0.8565**

Fig. 11: Clusters obtained by McDPC on the Olivetti Face dataset.

images to evaluate the performance of McDPC. The similarity between two images is computed by complex wavelet structural similarity (CW-SSIM) [32]. We take CW-SSIM as the facial features. The parameters of DPC are set following the values reported in [8]. As shown in Table 4, McDPC outperforms AP, DBSCAN and DPC again in this facial recognition experiment. Furthermore, we illustrate the clustering results in Fig. 11, wherein different colors represent different clusters.

Table 5: FM of McDPC and DPC when $pct = \{2, 3, 4, 5, 6\}$

Dataset	McDPC	DPC
Jain	**0.8346 ± 0.0958**	0.6694± 0.0988
Pathbased	**0.6818** ± 0.0534	0.5886 ± **0.0127**
2circles_noise	**0.8767** ± 0.1263	0.4934 ± **0.0089**
Pima	**0.7380 ± 0.0000**	0.7023 ± 0.0378
Sonar	**0.7070 ± 0.0000**	0.5145 ± **0.0000**
Iris	**0.7748** ± 0.0295	0.7715 ± **0.0000**

Table 6: Performance of McDPC on different parameter values

Dataset	FM	[range of γ]	FM	[range of θ]	FM	[range of λ]
Jain	0.86±0.08	[0.1,0.5]	1.00±0.00	[1.8,2.2]	1.00±0.00	[3.32,3.39]
Pathbased	1.00±0.00	[0.1,1.1]	1.00±0.00	[0.1,1.5]	1.00±0.00	[3.36,3.70]
2circles_noise	1.00±0.13	[0.1,0.5]	1.00±0.00	[0.5,2.0]	1.00±0.00	[2.89,3.18]
Pima	0.74±0.00	[0.01,0.05]	0.74±0.00	[1.0,50]	0.74±0.00	[0.46,0.63]
Sonar	0.71 ±0.00	[0.005,0.02]	0.71±0.00	[0.1,0.5]	0.71±0.00	[0.1,0.5]
Iris	0.83±0.08	[0.02,0.06]	0.92±0.00	[0.1,0.5]	0.92±0.00	[0.94,1.10]

4.5 Sensitivity Tests on Various Parameters of McDPC

As afore-introduced in Section 3, McDPC employs four parameters, namely γ, θ, λ and pct. In this subsection, we show that although McDPC employs relatively more number of parameters than the benchmarking models, e.g., DPC, McDPC's parameters are insensitive, or in other words, its performance is stable, within a reasonable value range. Specifically, we present the sensitivity test results on all four parameters used in McDPC and further provide general guidance on how to determine the parameter values.

First of all, we start from pct, which directly generates the decision graph in both DPC and McDPC. We apply difference parameter values, i.e., $pct = \{2, 3, 4, 5, 6\}$, to both algorithms on six datasets (three synthetic and three real-world) and report the results in Table 5, wherein better results (larger FM value and smaller standard deviation) are highlighted in bold. It is clearly shown that McDPC always performs better than DPC on FM values and they two do not differ much in terms of stability. As in the rest of the experiments reported in this subsection, we fix the other three parameter values in McDPC while testing the sensitivity of one parameter.

For parameters γ, θ and λ, the value ranges should be separately evaluated as they are all dependant on the different nature of the datasets. Therefore, in Table 6, we list the range of parameter values being tested alongside with the obtained FM values. Please note that for all datasets, we ran five experiments with the values equally interpolating the parameter range, and averaged the results. It is encouraging to see that the performance of McDPC is quite stable, because it only has small variations on the change of γ and no variations on θ and λ. Moreover, the performance of McDPC under different parameter testing is always better than that of DPC, as shown in Fig. 12.

Parameter γ is used to obtain Y-break (see (5)), which highly affects whether certain micro-clusters should be merged into one or retained as individual clusters

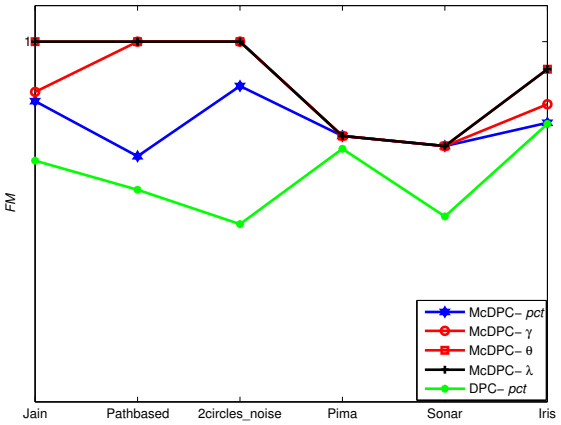


Fig. 12: Performance comparison between McDPC (under different parameter testing) and DPC.

(see Steps 8 to 13 in Algorithm 1). To properly select the value of γ, one needs to first examine whether there exist any natural gap(s) in the decision graph along the δ axis and then select accordingly.

Parameter θ is used to distinguish density levels and obtain X-break (see (4)). Specifically, the choice of θ should identify the obvious gap(s) exist in the decision graph along the ρ axis. In practise, if there does not exist any obvious gap, we can set θ to a small value.

Parameter λ is used to identify the representative data points (centers of micro-clusters) among those with high δ values (see Section 3.2). If a relatively large gap exists along the δ axis (see Fig. 3a), λ should be selected from the value range of the gap. As such, although being an important predefined threshold, the value of λ may not lead to any changes in performance, as shown in Table 6.

In summary, although McDPC employs three more number of parameters than DPC, these three parameters can be easily assigned based on the afore-introduced heuristics to obtain highly competitive results. As shown in Table 6 and Fig. 12, the parameter values of McDPC are insensitive within reasonable parameter ranges and McDPC always performs better than DPC on all datasets used for sensitivity testing.

4.6 Time Complexity Analysis of McDPC

The time complexity of computing parameters ρ and δ is $O(n^2)$, the time complexity of clustering based on different density levels is $O(n^2)$. Thus, the time complexity of McDPC is $O(n^2)$, which is close to DPC. As shown in Table 7, the time complexity of McDPC is not higher than other algorithms.

Table 7: Time Complexity of AP, DBSCAN, DPC and McDPC

AP	DBSCAN	DPC	McDPC
$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$

5 Conclusion

In this paper, we propose a novel density peak clustering algorithm, named Multi-center Density Peak Clustering (McDPC). McDPC overcomes the two major limitations of DPC: (i) not being able to identify clusters with multi-centers and (ii) not being able to detect clusters in the lower local density level. Specifically, McDPC applies ρ-cut to identify clusters whose centers have lower local densities and further applies δ-cut to identify clusters with multiple density peaks. Experimental results on six synthetic datasets and six real-world ones show that McDPC performs well comparing with other state-of-the-art clustering algorithms. More encouragingly, McDPC outperforms all the other benchmarking models on both the image segmentation and facial recognition experiments.

Going forward, we plan to develop an unsupervised method to extract deep features for the use of McDPC, and autonomous methods to determine the parameter values of McDPC, for color image clustering tasks.

Acknowledgements This research is supported by the National Natural Science Foundation of China (61772227,61572227), the Science & Technology Development Foundation of Jilin Province (20180201045GX) and the Social Science Foundation of Education Department of Jilin Province (JJKH20181315SK). This research is also supported, in part, by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-GC-2019-003), the Singapore Ministry of Health under its National Innovation Challenge on Active and Confident Ageing (NIC Project No. MOH/NIC/COG04/2017), and the Joint NTU-WeBank Research Centre on Fintech, Nanyang Technological University, Singapore.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Hans-Peter Kriegel, Martin Pfeifle, Hierarchical Density-Based Clustering of Uncertain Data, IEEE International Conference on Data Mining, Volume, 1-4 (2005)
2. Hong Chang, Dit Yan Yeung, Robust path-based spectral clustering, Pattern Recognition, 41, 191-203 (2008)
3. Manish Kashyap, Mahua Bhattacharya, A density invariant approach to clustering, Neural Computing and Applications, 28, 1695-1713 (2017)
4. Chamundeswari G., Varma Partha Saradhi, Satyanaraya Ch, Spatial Data Clustering: A Review, International Journal of Advanced Research in Computer Science, 5, 62-63 (2014)
5. Liang Bai, Xueqi Cheng, Jiye Liang, Huawei Shen, Yike Guo, Fast density clustering strategies based on the k-means algorithm, Pattern Recognition, 71, 375-386 (2017)

6. Ester Martin, Kriegel Hans-Peter, Xu Xiaowei, A density-based algorithm for discovering clusters in large spatial databases with noise, International Conference on Knowledge Discovery and Data Mining, 226-231 (1996)
7. P. Viswanath, R. Pinkesh, l-DBSCAN : A Fast Hybrid Density Based Clustering Method, Pattern Recognition, 1, 912-915 (2006)
8. Rodriguez Alex, Laio Alessandro, Clustering by fast search and find of density peaks, Science, 344, 1492–1496 (2014)
9. Mingjing Du, Shifei Ding, Yu Xue, A robust density peaks clustering algorithm using fuzzy neighborhood, International Journal of Machine Learning and Cybernetics, 1-10 (2017) 5‘
10. Veenman Cor J., Reinders Marcel J. T., Backer Eric, A Maximum Variance Cluster Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 1273-1280 (2002)
11. Yizhang Wang, Wei Pang, You Zhou, Density propagation based adaptive multi-density clustering algorithm, Plos One, 13, 1-13 (2018)
12. Jiajun Ding, Xiongxiong He, Junqing Yuan, Bo Jiang, Automatic clustering based on density peak detection using generalized extreme value distribution, Soft Computing, 22, 2777–2796 (2018)
13. Yaohui Liu, Zhengming Ma, Fang Yu, Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy, Knowledge-Based Systems, 133, 208-220 (2017)
14. Juanying Xie, Hongchao Gao, Weixin Xie, Xiaohui Liu, Philip W. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors, Information Sciences, 354, 19-40 (2016)
15. Mingjing Du, Shifei Ding, Hongjie Jia, Study on density peaks clustering based on K-nearest neighbors and principal component analysis, Knowledge-Based Systems, 99, 135-145 (2016)
16. Ji Xu, Guoyin Wang, Weihui Deng, DenPEHC: Density peak based efficient hierarchical clustering, Information Sciences, 373, 200-218 (2016)
17. Mengmeng Wang, Wanli Zuo, Ying Wang, An improved density peaks-based clustering method for social circle discovery in social networks, Neurocomputing, 179, 219-227 (2016)
18. M. Parmar, D. Wang, X. Zhang, A.-H. Tan, C. Miao, J. Jiang, Y. Zhou, REDPC: A residual error-based density peak clustering algorithm, Neurocomputing, 348, 82-96 (2019)
19. Peijie Lin, Yaohai Lin, Zhicong Chen, Lijun Wu, Lingchen Chen, Shuying Cheng, A density peak-based clustering approach for fault diagnosis of photovoltaic arrays, International Journal of Photoenergy, 2017, 1-14, (2017)
20. Bing Tu, Xianchang Yang, Nanying Li, Chengle Zhou, Danbing He, Hyperspectral Anomaly Detection via Density Peak Clustering, Pattern Recognition Letters, 1-10, (2019)
21. Shifei Ding, Mingjing Du, Tongfeng Sun, Xiao Xu, Yu Xue, An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood, Knowledge-Based Systems,113, 294-313 (2017).
22. Maozu Guo, Donghua Yu, Guojun Liu, Xiaoyan Liu, Shuang Cheng, Drug-target interaction data cluster analysis based on improving the density peaks clustering algorithm, Intelligent Data Analysis, 23, 1335-1353 (2019)
23. Frey, B. J., Dueck, D, Clustering by passing messages between data points, Science, 315, 972-973 (2007)
24. Givoni, I. E., Frey, B. J., A binary variable model for affinity propagation, Neural Computation, 21, 1589-1600 (2009)
25. David Martin Powers, Evaluation: From Precision, Recall and F-Measure To ROC, Informedness, Markedness and Correlation, Journal of Machine Learning Technologies, 2, 2229-3981 (2011)
26. Nguyen Xuan Vin, Julien Epps, James Bailey, Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance, The Journal of Machine Learning Research, 11, 2837-2854 (2010)
27. Haijun Zhang, Han Guo, Xinghao Wang, Yuzhu Ji, Q. M. Jonathan Wu, Clothescounter: A framework for star-oriented clothes mining from videos, Neurocomputing, 1-10, (2019)
28. Yong Shi, Zhensong Chen, Zhiquan Qi, Fan Meng, Limeng Cui, A novel clustering-based image segmentation via density peaks algorithm with mid-level feature, Neural Computing and Applications, 28, 29-39 (2017)
29. Yanhui Guo, Rong Xia, Abdulkadir Sengur, Kemal Polat, A novel image segmentation approach based on neutrosophic c-means clustering and indeterminacy filtering, Neural Computing and Applications, 28, 3009-3019 (2017)

30. Shanwen Zhang, Zhuhong You, Xiaowei Wu, Plant disease leaf image segmentation based on superpixel clustering and EM algorithm, Neural Computing and Applications, 31, 1225-1232 (2019)
31. Haijun Zhang, Shuang Wang, Xiaofei Xu, Tommy W. S. Chow, Q. M. Jonathan Wu, Tree2Vector: Learning a Vectorial Representation for Tree-Structured Data, IEEE Transactions on Neural Networks and Learning Systems,11,5304-5318 (2018)
32. Samaria, F. S, Harter, A. C, Parameterisation of a stochastic model for human face identification, Proceedings of IEEE Workshop on Applications of Computer Vision, 22, 138-142 (1994)