12-2017

A novel density peak clustering algorithm based on squared residual error

Milan PARMAR

Di WANG

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Chunyan MIAO

Jianhua JIANG

See next page for additional authors

A Novel Density Peak Clustering Algorithm based on Squared Residual Error

Milan Parmar[†‡], Di Wang[§], Ah-Hwee Tan[§¶], Chunyan Miao[§¶], Jianhua Jiang[‡], You Zhou[†*]
[†]College of Computer Science and Technology
Jilin University
Changchun, China
[‡]School of Management Science and Information Engineering
Jilin University of Finance and Economics
Changchun, China
[§]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly
[¶]School of Computer Science and Engineering
Nanyang Technological University
Singapore
[*]Corresponding Author
milaanparmar9@hotmail.com, {wangdi, asahtan, ascymiao}@ntu.edu.sg, jjh@jlufe.edu.cn, zyou@jlu.edu.cn

***Abstract*—The density peak clustering (DPC) algorithm is designed to quickly identify intricate-shaped clusters with high dimensionality by finding high-density peaks in a non-iterative manner and using only one threshold parameter. However, DPC has certain limitations in processing low-density data points because it only takes the global data density distribution into account. As such, DPC may confine in forming low-density data clusters, or in other words, DPC may fail in detecting anomalies and borderline points. In this paper, we analyze the limitations of DPC and propose a novel density peak clustering algorithm to better handle low-density clustering tasks. Specifically, our algorithm provides a better decision graph comparing to DPC for the determination of cluster centroids. Experimental results show that our algorithm outperforms DPC and other clustering algorithms on the benchmarking datasets.**

***Index Terms*—clustering, density peak clustering, squared residual error, low-density data points**

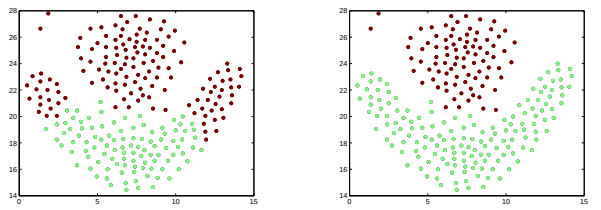
I. Introduction

Clustering algorithms aim to analyze data by discovering their underlying structure and organize them into separate categories according to their characteristics expressed as internal homogeneity and external bifurcation without priori-knowledge. Successful applications of clustering techniques are evident in various domains, such as pattern recognition [1] [2], bioinformatics [3], disease diagnosis [4], risk analysis [5] [6], etc. Moreover, some emerging topics such as big data [7], virtual reality [8], IoT [9], etc., also avail from clustering methods. In general, clustering methods can be broadly categorized into five groups based on their dynamics: partitioning [10], hierarchical [11] [12], density-based [13], model-based [14], and grid-based [15].

This research is supported in part by the Science & Technology Development Foundation of Jilin Province under grant No. 20160101259JC and the National Science Fund Project of China No. 61772227. This research is also supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative.

Density-based clustering methods have been widely used to form arbitrary shape clusters by detecting high-density regions in the high dimensional data space. Fundamentally, the region with high density, or a set of densely connected data points, in the data space is treated as a cluster. Density-based spatial clustering of applications with noise (DBSCAN) [16] is probably the most well-known density-based clustering algorithm engendered from the basic notion of the local density, which creates arbitrary-shaped clusters. Recently, density-based clustering methods have attracted more attention since Rodriguez and Liao proposed their density peak clustering (DPC) algorithm [17] in 2014. The desirable features of DPC include detection of non-spherical clusters without specifying the number of clusters, few number of control parameters, and autonomous identification of cluster centroids for varying cluster sizes and within-cluster density.

However, DPC has its limitations. Alike DBSCAN, DPC may fail to capture thin clusters by using its decision graph (see Section II), i.e., it does not perform well on anomaly detection. Data distribution within clusters has to be carefully examined to detect anomalies, mainly because the presence of anomalies is a clear sign of erroneous conditions that may lead to significant performance degradation [18]. As shown in Fig. 1, DPC generates groups of data points by identifying clusters with maximum density, but does not handle well the uneven distribution in individual clusters (also pointed out in [19]), e.g., the two anomalies (at top left corner) are always considered as part of a larger cluster regardless of different C_d values, where C_d denotes the user specified cutoff disctance (see Section II). In such cases, it is difficult for DPC to pick up all the outliers with varying C_d values and it may not be able to find clusters of small sizes or consisting of borderline points and outliers (relatively speaking) only. Furthermore, when the dimensionality of the underlying dataset increases, the well-known "curse of dimensionality" problem [20] will



(a) $C_d = 0.9301$ (b) $C_d = 1.3124$

(c) $C_d = 2.1213$ (d) $C_d = 3.2381$

Fig. 1. Visualizations of clusters identified in the Flame dataset by DPC with different C_d parameter values.

exacerbate. Hence, in this paper, we propose a novel density-based clustering algorithm to detect anomalies that cannot be identified using existing methods. As a result, the proposed algorithm can better identify and handle various types of anomalies manifested in different patterns.

To correctly and efficiently identify the anomalies and consequently finalize the cluster formation, we rely on using the concept of ***halo points*** to unravel low-density points in the following two ways (see Section III-C for more technical details): **(i)** *halo points identification*: a set of low-density points are considered as *halo points*, and **(ii)** *halo points decision*: *halo points* can be categorized into outliers and borderline points that they are either merged into certain existing clusters or used to form new clusters. To better deal with *halo points* and hence increase the clustering performance, in this paper, we propose an effective density-based clustering algorithm based on squared residual error (e^2) [21].

The main contributions of our proposed clustering method are listed as follows:

1) We incorporate the squared residual error theory to enable the discovery of anomalies and borderline points by identifying the *halo points*.
2) The decision graph derived by our proposed clustering method can better identify the cluster centroids and aggregate clusters.
3) *Halo points* make it easier to isolate anomalies from borderline points.

We apply our proposed algorithm on four synthetic datasets and four UCI datasets for performance evaluations. We also apply K-Means [22], affinity propagation (AP) [11] and DPC on the same datasets for comparisons. Experimental results show that our algorithm achieves the best performance on most datasets (specifically, best on seven out of eight datasets and the second best on the remaining dataset).

The rest of the paper is organized as follows. We briefly introduce the dynamics of DPC in Section II. We present our proposed clustering method based on squared residual error in Section III. We report the experimental results with comparisons and discussions in Section IV. We draw the conclusion and propose future work in Section V.

II. Related Work

We present the dynamics, pros and cons of DPC in this literature review section. In a nutshell, DPC generates clusters by assigning data points to the same cluster of its nearest neighbor with higher density. Moreover, DPC uses the decision graph approach to identify cluster centroids. A decision graph is derived based on the following two fundamental properties of each data point $\boldsymbol{x_i}$: **(i)** local density $\boldsymbol{\rho_i}$ and **(ii)** individual distance of each data point from points of higher density $\boldsymbol{\delta_i}$.

Assume a dataset consists of $\boldsymbol{X_{P\times M}} = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_P}]^{\boldsymbol{T}}$, where $\boldsymbol{x_i} = [\boldsymbol{x_{1i}}, \boldsymbol{x_{2i}}, ..., \boldsymbol{x_{Mi}}]$ is a vector with $\boldsymbol{M}$ attributes and $\boldsymbol{P}$ is the total number of data points. The distance between two data points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is computed as follows:

$$\boldsymbol{d_{ij}} = || \boldsymbol{x_i} - \boldsymbol{x_j} ||. \tag{1}$$

The local density of a data point $\boldsymbol{x_i}$, denoted as $\boldsymbol{\rho_i}$ and known as the hard threshold [17], is then defined as:

$$\boldsymbol{\rho_i} = \sum_j \boldsymbol{\chi}(\boldsymbol{x}) \cdot (\boldsymbol{d_{ij}} - \boldsymbol{C_d}), \tag{2}$$

where $\boldsymbol{\chi}(\boldsymbol{x}) = \boldsymbol{1}$, if $\boldsymbol{x} < \boldsymbol{0}$, and $\boldsymbol{C_d}$ is the cutoff distance that user specified to control the weight degradation rate. The determination of $\boldsymbol{C_d}$ is actually the assignment of the average number of neighbors that each data point has. Specifically, $\boldsymbol{\rho_i}$ is defined as the number of data points that have shorter distance than $\boldsymbol{C_d}$ and are adjacent to $\boldsymbol{x_i}$.

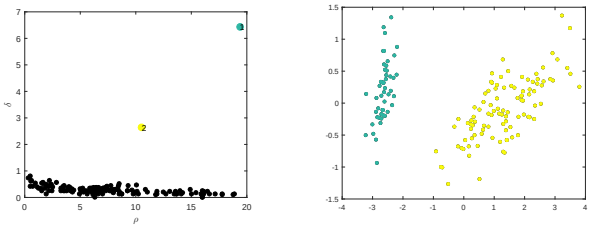
Another way of local density computation known as the soft threshold [17] is defined as follows:

$$\boldsymbol{\rho_i} = \sum_j \boldsymbol{exp}\left(-\frac{\boldsymbol{d_{ij}^2}}{\boldsymbol{C_d^2}}\right). \tag{3}$$

$\boldsymbol{\delta_i}$ is defined as the shortest distance from any other data point that has a higher density value than $\boldsymbol{x_i}$. If $\boldsymbol{x_i}$ has the highest density value, $\boldsymbol{\delta_i}$ is assigned to the longest distance to any other data point. Specifically, $\boldsymbol{\delta_i}$ is computed as follows:

$$\boldsymbol{\delta_i} = \begin{cases} \boldsymbol{\min\limits_{j:\rho_j>\rho_i}} \boldsymbol{d_{ij}}, & \text{if } \exists\, j \text{ s.t. } \boldsymbol{\rho_j} > \boldsymbol{\rho_i}, \\ \boldsymbol{\max\limits_{j}} \boldsymbol{d_{ij}}, & \text{otherwise.} \end{cases} \tag{4}$$

DPC finds a border region for each cluster, where the region is defined as the set of points assigned to that cluster but within certain distance (i.e., $\boldsymbol{C_d}$) from the data points belonging to another cluster. Subsequently, DPC finds the data point of the highest density within the border region of that cluster and denotes its density as $\boldsymbol{\rho_b}$. The data points of the cluster whose density is higher than $\boldsymbol{\rho_b}$ are considered as part of the



(a) Iris decision graph by DPC (b) Iris cluster formation by DPC

Fig. 2. Determination of cluster centroids and the resulting cluster formation based on the decision graph generated by DPC on the Iris dataset.

cluster core and others are considered as part of the cluster halo (suitable to be considered as noise) [17].

The performance of DPC is highly sensitive to the identification of the cluster centroids [17]. Cluster centroids with high local density $\boldsymbol{\rho}$ and high $\boldsymbol{\delta}$ can be easily identified in the decision graph (see Fig. 2(a)). Nevertheless, it is difficult for DPC to identify cluster centroids with low $\boldsymbol{\rho}$ and high $\boldsymbol{\delta}$, or with high $\boldsymbol{\rho}$ and low $\boldsymbol{\delta}$.

III. REDPC: Residual Error-based Density Peak Clustering Algorithm

In this section, we present our proposed clustering algorithm named Residual Error-based Density Peak Clustering (REDPC), which inherits the strengths of centroid detection from DPC [17], distance measure from residual error theory [21], and density-connectivity from DBSCAN [16].

The dynamics of REDPC are designed according to the following two bases:

1) A cluster is formed when its centroid is surrounded by only the data points with higher residual error.
2) A data point can be assigned to the cluster when there is another data point with higher $\boldsymbol{\delta}$ (see the following subsection) and lower residual error.

The overall REDPC procedure consists of the following four stages and each stage is elaborated in the following subsections, respectively.

1) *Preprocessing*: compute the squared residual error between data points and compute $\boldsymbol{\delta}$.
2) *Initial assignments*: generate the decision graph based on residual errors, identify centroids, and assign data points with their respective cluster label.
3) *Halo identification*: identify halo points (consists of borderline points and anomalies).
4) *Final refinements*: detect and isolate anomalies from *halo points* and output the final clustering results (with anomalies represented using special symbols).

A. Preprocessing

Unlike DPC [17], we incorporate the residual error approach instead of relying on local density between data points, because residual error constructs a more informative decision graph in the later stage, which may lead to better clustering performance. Specifically, the squared residual error ($\boldsymbol{e_{ij}^2}$) of a data point $\boldsymbol{x_i}$ to its neighbor $\boldsymbol{x_j}$ is determined by the distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ and the neighborhood size:

$$\boldsymbol{e_{ij}^2 = \frac{|| x_i - x_j ||^2}{| N_i |^2}}, \tag{5}$$

where $|| \cdot ||$ denotes Euclidean distance, $\boldsymbol{N}$ is a predefined parameter, which defines the neighborhood size, and $\boldsymbol{| N_i |}$ denotes the number of data points in $\boldsymbol{N_i}$.

Similar to DPC, a cut-off residual $\boldsymbol{C_d}$ value is predefined and later in Section III-C, $\boldsymbol{C_d}$ is used to identify *halo points*.

$\boldsymbol{\delta_i}$ denotes the minimum distance of data point $\boldsymbol{x_i}$ to another data point with lower residual error. $\boldsymbol{\delta_i}$ is computed as follows:

$$\boldsymbol{\delta_i} = \begin{cases} \boldsymbol{\min\limits_{j:(e_{ji}^2)<(e_{ij}^2)}} || \boldsymbol{x_i - x_j} ||, & \text{if } \exists\, j \text{ s.t. } \boldsymbol{(e_{ji}^2) < (e_{ij}^2)}, \\ \boldsymbol{\max\limits_{j:(e_{ji}^2)<(e_{ij}^2)}} || \boldsymbol{x_i - x_j} ||, & \text{otherwise.} \end{cases} \tag{6}$$

B. Initial Assignments

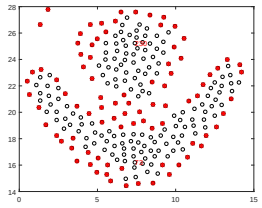
A decision graph is plotted based on residual error $\boldsymbol{e^2}$ between data points and $\boldsymbol{\delta}$. First, each centroid of the cluster is identified by its relatively low $\boldsymbol{e_{ij}^2}$ and high $\boldsymbol{\delta_i}$. Secondly, each data point is assigned to the same cluster as its neighbor with low residual error and high $\boldsymbol{\delta}$. As such, cluster labels are initially assigned (see Algorithm 1).

Algorithm 1 Cluster aggregation algorithm.

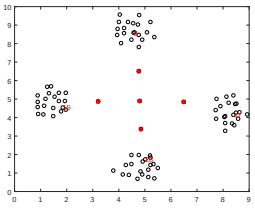
Require: X (Centroids), sortd_$\boldsymbol{e_{ij}^2}$ (residual error vector of point $\boldsymbol{i}$) sorted in ascending order
Ensure: *Cl* (aggregated clusters)
  for $i \leftarrow$ 1:*Total(X)* **do**
    Cl(i) $\leftarrow$ 1:*X(i)*
  end for
  for $j \leftarrow$ 1:n **do**
    if *Cl(sortd_*$\boldsymbol{e_{ji}^2}$*)* $\neq$ *"label_not_assigned"* **then**
      Cl(sortd_*$\boldsymbol{e_{ji}^2}$)* $\leftarrow$ *Cl(NNneigh(sortd_*$\boldsymbol{e_{ji}^2}$*)*
    end if
  end for

C. Halo Identification

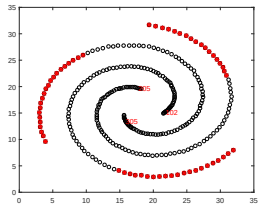
After cluster aggregation, we further identify the halo points if the number of identified centroids is greater than one. First, we compute the average residual error *avg_*$\boldsymbol{e^2}$ between each data point $\boldsymbol{x_i}$ in cluster *Cl(i)* and each data point $\boldsymbol{x_j}$ in cluster *Cl(j)* if the residual error between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is less than $\boldsymbol{C_d}$. Secondly, we define the border residual error of every point belongs to the same cluster *border_*$\boldsymbol{e^2}$*(Cl(i))* with the value of *avg_*$\boldsymbol{e^2}$, if and only if *avg_*$\boldsymbol{e^2} < 1$. Finally, all the *halo points* of the clusters are identified if the individual residual error of each data point in the cluster exceeds the corresponding cluster border residual value, i.e., *Cl(i)* $>$ *border_*$\boldsymbol{e^2}$*(Cl(i))* (see Algorithm 2). All the halo points of all the clusters are collected in a *haloset* wherein each point is labeled with its assigned *clusterId*. As exemplified in Fig. 3, halo points



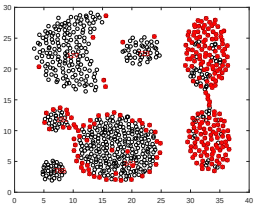
(a) Flame



(b) D2



(c) Spiral



(d) Aggregation

Fig. 3. Halo points detected in Flame, D2 (see Section IV), Spiral and Aggregation datasets.

identified in various datasets are represented with circles in red color.

Algorithm 2 *Halo points* identification.

Require: *Cl* (aggregated clusters)
Ensure: *haloset* (vector of halo points)
  if *Total(X)* $>$ 1
  *border_*e^2 $\leftarrow$ *ones*(1, *Total(X)*)
    for *i* $\leftarrow$ 1:*n*-1 **do**
      for *j* $\leftarrow$ *i*+1:*n* **do**
        if *Cl(i)* $\cong$ *Cl(j)* && *DM(i, j)*<= C_d
          *avg_*e^2 = ($e^2(i)$+$e^2(j)$)/2
          if *avg_*e^2 $<$ border_e^2*(Cl(i))* **then**
            border_*e^2(Cl(i))* $\leftarrow$ *avg_*e^2
          end if
          if *avg_*e^2 $<$ *border_*e^2*(Cl(j))* **then**
            border_*e^2(Cl(j))* $\leftarrow$ *avg_*e^2
          end if
        end if
      end for
    end for
    if *Cl(i)* $>$ *border_*e^2*(Cl(i))* **then**
      halo(i)$\leftarrow$ 0 (*halo point identified not belong to any class (0)*)
    end if
  end if
  haloset = find(halo(:)==0) % put all halo points in haloset

D. Final Refinements

During anomaly detection, a halo point with high residual error and high δ (these threshold values are auto-derived, see Algorithm 3) is recognized as an anomaly. All the anomalies are collected in the *anoset*. The final clustering results are then plotted in a way to highlight those identified anomalies (see Fig. 5(d)).

TABLE I
THE PROPERTIES OF THE SYNTHETIC AND UCI DATASETS

Datasets	Parameters		
	No. of Points	*No. of Dimensions*	*No. of Clusters*
Flame	240	2	2
Aggregation	788	2	7
Spiral	312	2	3
Iris	150	4	3
Seeds	210	7	3
Wine	178	13	3
Glass	214	9	6
D2	85	2	4

Algorithm 3 Anomaly identification.

Require: *haloset* (vector of halo points)
Ensure: *anoset* (vector of anomalies points)
  Set thresold limit for e^2 *and* δ *for anomaly detection*
  *limit_*e^2 $\leftarrow$ *mean(*e^2*)+ sortd(*e^2*)**0.8
  *limit_*δ $\leftarrow$ *max(*δ*)+min(*δ*)*/2
  for *i* $\leftarrow$ 1:*haloset* **do** *% for every halo point*
    if *Cl(i)* $>$ *limit_*e^2 && δ_i $<$ *limit_*δ **then**
      ano(i) $\leftarrow$ 0 *% anomaly identified*
    end if
  end for
  anoset = find(ano(:)==0) % put all anomalies in anoset

It is of great importance to distinguish the anomalies from normal data points and reasonable outliers because anomalies highly likely represent the abnormal patterns or malicious activities in real-world scenarios. For example, unusual road traffic patterns may suggest nearby accidents or emergencies, unusual credit card transactions may indicate identity theft, unusual computer network loads should alert the cyber security division, etc.

IV. EXPERIMENTAL RESULTS

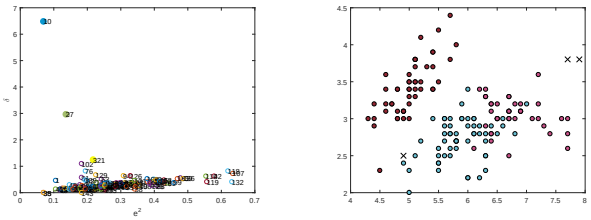
To test the feasibility and validate the robustness of REDPC, we compare its performance with K-Means [22], AP [11], and DPC [17] on three widely-used synthetic clustering datasets, namely Flame, Aggression and Spiral, four UCI datasets, namely Iris, Seeds, Wine and Glass, and one own-defined dataset D2[1]. The properties of all eight datasets are listed in Table I.

In this paper, we use F-score to measure the accuracy of the clustering results. The performance comparisons among all the benchmarking models are reported in Table II. It is encouraging to find that REDPC achieves the highest F-score on seven out of eight datasets. Although REDPC only achieves the second best on Seed, the difference between the winner is as small as $\mathbf{0.8068 - 0.8065 = 0.0003}$ or $\mathbf{0.03\%}$.

[1]The D2 dataset (with cluster labels) is available online: https://www.dropbox.com/s/899xltgq3gg09bg/D2_with_label.csv?dl=0

TABLE II
F-SCORE ON EIGHT BENCHMARKING DATASETS

Dataset	K-Means	AP	DPC	*REDPC*
Iris	0.8208	0.4851	0.7715	**0.8404**
Seeds	**0.8068**	0.3877	0.8026	0.8065
Wine	0.5835	0.3142	**0.5892**	**0.5892**
Glass	0.5052	0.2874	0.5418	**0.5542**
Spiral	0.3277	0.2853	**1**	**1**
Flame	0.7364	0.2874	**1**	**1**
Aggregation	0.7725	0.3429	**1**	**1**
D2	0.4333	0.4332	**1**	**1**



(a) Iris decision graph by REDPC (b) Cluster formation by REDPC

Fig. 4. Determination of cluster centroids and the resulting cluster formation based on the decision graph generated by REDPC on the Iris dataset.

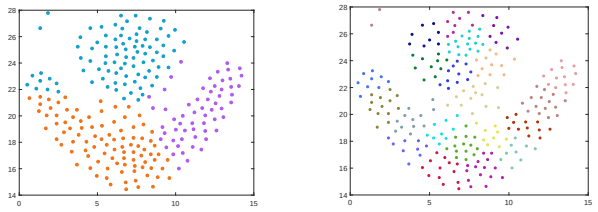
Other than REDPC always performs better or equally good when compared to DPC, we find that it is much easier to identify cluster centroids by using the decision graph derived by REDPC than that by DPC. It is shown in Fig. 2(a) that the third cluster centroid is difficult to be identified merely based on $\boldsymbol{\rho}$ and $\boldsymbol{\delta}$. However, as shown in Fig. 4(a), the identification of the third cluster centroid is easier based on $\boldsymbol{e^2}$ and $\boldsymbol{\delta}$. More importantly, DPC does not perform well when there are ascertaining anomalies whose distance to higher density points is less than $\boldsymbol{C_d}$. On the other hand, REDPC uses $\boldsymbol{e^2}$ as one of the identification criteria, which reduces the dependency of $\boldsymbol{C_d}$. This is the main reason why REDPC outperforms DPC on all the UCI datasets.

To illustrate the capability of REDPC in anomaly detection, we present the clustering results of applying all the benchmarking clustering methods on the Flame dataset in Fig. 5. Comparing Fig. 5(d) to the rest of the subfigures, it is clearly shown that only REDPC successfully identifies the anomalous data points in the top left corner (although both DPC and REDPC achieve 100% $\boldsymbol{F}$-score).

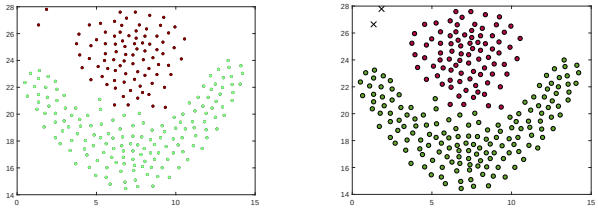
To illustrate the performance of REDPC on datasets with different density distributions, we present the clustering results of applying all the benchmarking clustering methods on the Aggregation dataset in Fig. 6. Fig. 6(d) shows that REDPC can perfectly handle clusters of different sizes with boundaries in close proximity.

V. Conclusion

In this paper, we propose a novel density peak type of clustering method named REDPC by using squared residual error to better identify cluster centroids. The experimental results on both synthetic and real-world UCI datasets demonstrate that REDPC outperforms DPC and other clustering algorithms.

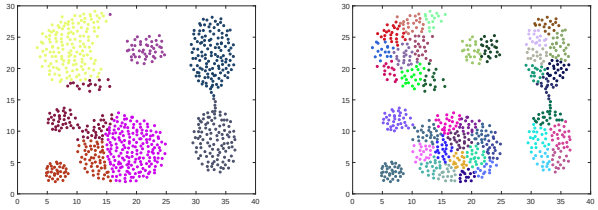


(a) $\boldsymbol{K}$-Means, $\boldsymbol{K}$ = 3 (b) AP, preference = -5.9261

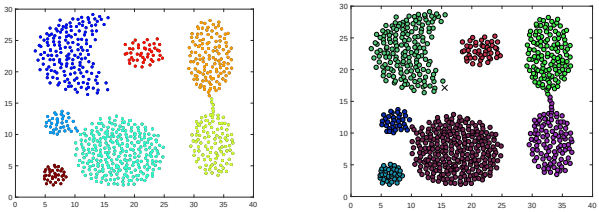


(c) DPC, $\boldsymbol{C_d}$ = 1.6008 (d) REDPC, $\boldsymbol{N_i}$ = 5, $\boldsymbol{C_d}$ = 1.4577 (#clusters = 2, #anomalies = 2)

Fig. 5. An illustration of anomaly detection on the Flame dataset.



(a) $\boldsymbol{K}$-Means, $\boldsymbol{K}$ = 7 (b) AP, preference = -16.5170



(c) DPC, $\boldsymbol{C_d}$ = 2.3162 (d) REDPC, $\boldsymbol{N_i}$ = 5, $\boldsymbol{C_d}$ = 1.8601

Fig. 6. Cluster formation on the Aggregation dataset.

Going forward, we will improve the proposed clustering algorithm for more autonomy in parameter value determinations, refinement in the clustering dynamics for better performance, and applications on more complex and challenging datasets.

References

[1] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2138–2150, 2014.

[2] J. Wen, D. Zhang, Y. Cheung, H. Liu, and X. You, "A batch rival penalized expectation-maximization algorithm for gaussian mixture clustering with automatic model selection." *Computational and Mathematical Methods in Medicine*, vol. 2012, p. 425730, 2012.

[3] C. Xu and Z. Su, "Identification of cell types from single-cell transcriptomes using a novel clustering method," *Bioinformatics*, vol. 31, no. 12, pp. 1974–1980, 2015.

[4] D. Wang, C. Quek, and G. S. Ng, "Ovarian cancer diagnosis using a hybrid intelligent system with simple yet convincing rules," *Applied Soft Computing*, vol. 20, pp. 25–39, 2014.

[5] G. Kou, Y. Peng, and G. Wang, "Evaluation of clustering algorithms for financial risk analysis using MCDM methods," *Information Sciences*, vol. 275, pp. 1–12, 2014.

[6] D. Wang, C. Quek, and G. S. Ng, "Bank failure prediction using an accurate and interpretable neural fuzzy inference system," *AI Communications*, vol. 29, no. 4, pp. 477–495, 2016.

[7] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.

[8] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, "Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey," *arXiv preprint arXiv:1303.2292*, 2013.

[9] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L. T. Yang *et al.*, "Data mining for internet of things: A survey." *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[10] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "Survey of multiobjective evolutionary algorithms for data mining: Part II," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 20–35, 2014.

[11] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[12] D. Wang and A.-H. Tan, "Self-regulated incremental clustering with focused preferences," in *Proceedings of International Joint Conference on Neural Networks*. IEEE, 2016, pp. 1297–1304.

[13] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, 2016.

[14] T. Chen, N. L. Zhang, T. Liu, K. M. Poon, and Y. Wang, "Model-based multidimensional clustering of categorical data," *Artificial Intelligence*, vol. 176, no. 1, pp. 2246–2269, 2012.

[15] M. Parikh and T. Varma, "Survey on different grid based clustering algorithms," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 2, no. 2, 2014.

[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[17] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, p. 15, 2009.

[19] M. Wang, W. Zuo, and Y. Wang, "An improved density peaks-based clustering method for social circle discovery in social networks," *Neurocomputing*, vol. 179, pp. 219–227, 2016.

[20] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[21] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[22] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.