Annals of Mathematics and Artificial Intelligence (2022) 90:855–872
https://doi.org/10.1007/s10472-021-09744-4

Neighborhood density information in clustering

Mujahid N. Syed[1]

Accepted: 6 April 2021 / Published online: 27 May 2021

Abstract
Density Based Clustering (DBC) methods are capable of identifying arbitrary shaped data clusters in the presence of noise. DBC methods are based on the notion of local neighborhood density estimation. A major drawback of DBC methods is their poor performance in high-dimensions. In this work, a novel DBC method that performs well in high-dimensions is presented. The novelty of the proposed method can be summed up as follows: a hybrid first-second order optimization algorithm for identifying high-density data points; an adaptive scan radius for identifying reachable points. Theoretical results on the validity of the proposed method are presented in this work. The effectiveness and efficiency of the proposed approach are illustrated via rigorous experimental evaluations. The proposed method is compared with the well known DBC methods on synthetic and real data from the literature. Both internal and external cluster validation measures are used to evaluate the performance of the proposed method.

Keywords Data clustering · Nonlinear optimization · Density estimation

Mathematics Subject Classification 2010 16:90XX · 27:90C30 · 62H30:1:91C20

1 Introduction

Arguably, the most popular and oldest data clustering algorithm is the K-means [1–3]. K-means algorithm partitions the given data points into exactly K spherical clusters. Although, K-means algorithm is predominantly known in the data clustering community, it has the following inherent disadvantages: The value of K must be supplied as an input to the algorithm. The algorithm's cluster definition is limited to the spherical shapes. The algorithm cannot differentiate actual data points from the noise (unrelated/erroneous data points) in the data. Nevertheless, there has been many enhancements proposed in the literature ([3, 4]) of clustering that tries to overcome the above disadvantages.

✉ Mujahid N. Syed
snumujahid@gmail.com; smujahid@kfupm.edu.sa

[1] Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Kingdom of Saudi Arabia

Specifically, Density Based Clustering (DBC) methods overcome the above disadvantages [5, 6]. One of the seminal DBC methods is Density Based Spatial Clustering of Applications with Noise (DBSCAN) [7]. DBSCAN groups a set of contiguous points with sufficient neighborhood density, termed as reachable points, into one cluster (see [7] for the detailed description). Points that lie in low neighborhood density regions are marked as noise. The definition of reachable points depends upon two critical parameters, namely: scan radius (ϵ) and minimum number of points to form a cluster point ($MinPts$).

A discussion on the performance of DBSCAN and its sensitivity w.r.t the two parameters is presented in [8]. To the best of our knowledge, there exists no automated mechanism to estimate the scan radius in higher dimensions. Although the radius estimation in two or three dimensions is manageable via manual, graphical and empirical approaches like nearest neighbor approach [9]; its estimation in the higher dimensions is still an open question for DBSCAN. In addition to that, the non-adaptable nature of the scan radius in the presence of low dimensional clusters embedded in higher dimensions and varying densities results in the poor performance of DBSCAN.

In this paper, an optimization based DBC method is proposed. The method can identify arbitrarily shaped separable clusters in the higher dimensions. By separable clusters, it is assumed that the clusters are disconnected or non-overlapping. The key novelty of the method is that it provides a scan radius estimation mechanism in any dimension. The rest of the paper is organized as follows. Section 2 reviews the related work related to DBC. Section 3 introduces the concept of neighborhood density based information for convex clusters. In Section 4 the concept is extended for the case of arbitrarily shaped separable clusters, and an algorithm that implements the enhancement is presented. Numerical experiments that illustrate the performance of the proposed algorithm on 2 & 3 dimensional data sets are presented in Section 5. Section 6 depicts the performance of the proposed algorithm in the higher dimensions (≥ 3). Finally, Section 7 concludes the paper by highlighting advantages and disadvantages of the proposed algorithm.

2 Related work on DBC

At Knowledge Discovery and Data Mining (KDD) conference in 2014, DBSCAN received "the test of time" award. Since its inception, numerous DBC methods were inspired on the rationale of DBSCAN. Some of the well know extensions or related works of DBSCAN are presented as follows. Generalized DBSCAN (GDBSCAN) method is one of the earliest extension of DBSCAN that generalizes the definition of neighborhood and distance measure [10]. The generalization proposed in GDBSCAN facilitates clustering of data points and spatial objects according to their (spatial and non-spatial) attributes. Ordering Points To Identify the Clustering Structure (OPTICS) method is based on the ideas similar to DBSCAN [11]. However, OPTICS is capable of discovering clusters of varying density.

A comprehensive list of works on DBC is out of scope for this paper. However, few density based well known methods that highlights the usage of density criterion in cluster identification is surveyed. A generalized projection based clustering method known as arbitrarily ORiented projected CLUSter generation (ORCLUS) was proposed in [12]. For a given high dimensional data, ORCLUS finds clusters in lower dimensional subspace via projection mechanism. Mixture model based approaches presented in [13, 14] illustrate the usage of Gaussian density approach. In [15] a density based clustering was proposed that can handle multiple levels of cluster densities. The approach was based on nearest neighbor and shared nearest neighbor density information of data points. A nonparametric estimation

of density in identifying clusters was implemented in [16, 17]. Moreover, the idea of assigning high density points that have large distances among them as cluster centers was depicted in [18]. A hybrid approach involving K-means and expectation maximization concepts can be seen in [19]. In addition to that, see [20, 21] and the references there in for an overview of the density based approaches in clustering. In the next section, a density function whose maxima corresponds to density peaks is proposed.

3 Neighborhood density information & convex clusters

Without loss of generality, assume that there are N data points in n dimensions. Let $\mathbf{y}_p = [y_{p,1}, \ldots, y_{p,n}]^T$ be a column vector representing the p^{th} data point, $\forall\ p = 1, \ldots, N$. Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ be the matrix representing the data. Assume that the N data points belong to some K unknown but separable clusters. In order to build the concept, it is assumed that the clusters are convex, and can be represented by cluster centers. Let $\mathbf{x}$ be one of the cluster center, and p be any data point. The density information provided by the i^{th} coordinate of the cluster center, about the i^{th} coordinate of the data point w.r.t some window ($\sigma > 0$) can be measured by the Parzen window estimate $k_\sigma() : \mathbb{R} \mapsto \mathbb{R}$, where $k_\sigma()$ is a profile function. The cumulative density information between the cluster center and the data point can be defined as the product of individual coordinate information. The neighborhood density information of the data provided by the cluster center can be defined as:

$$\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma) = \frac{1}{N} \sum_{p=1}^{N} \left(\prod_{i=1}^{n} k_\sigma(y_{pi}, x_i) \right), \tag{1}$$

where $\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma)$ is the average information of data $\mathbf{Y}$ at the cluster center estimate $\mathbf{x}$ based on the profile function $k_\sigma()$. The information function given in Equation (1) is a generic function that appears very often in the density based analysis.

A profile function with appropriate properties should be selected to quantify the measure. One of the criteria to select the profile function depends upon the cluster structure (spherical, hyperplane, etc.). Properties like symmetry, smoothness, robustness and concavity, which provides mathematical advantage, should also be considered in selecting the profile function. In this work, it is assumed that the profile function satisfies the following properties: (1) $0 \leq k_\sigma(y_{pi}, x_i) \leq 1 \quad \forall\ \sigma,\ i,\ p$, and (2) If $|y_{pi} - x_i| \leq |y_{qi} - x_i|$, then $k_\sigma(y_{pi}, x_i) \leq k_\sigma(y_{qi}, x_i) \quad \forall\ \sigma,\ i,\ p,\ q$; where $i = 1, \ldots, n$ and $p, q = 1, \ldots, N$.

The former is a normalization property that results in $\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma) \in [0, 1]$. A value of 0 implies no density information, and a value of 1 implies the maximum information. The latter inequality states that the cluster center will have higher information about a point closer to it, than compared to a farther point. In this work, a Gaussian kernel function is used as the profile function, i.e.

$$\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma) = \frac{1}{N} \sum_{p=1}^{N} \left(\prod_{i=1}^{n} e^{\frac{-(y_{pi} - x_i)^2}{2\sigma^2}} \right). \tag{2}$$

The above function not only satisfies the above two properties, but also simultaneously covers the three desirable properties: smoothness, robustness and concavity. That is, the resulting $\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma)$ function with the Gaussian kernel is infinitely differentiable, hence it is smooth. The shape of the kernel function provides robustness. Concavity is not achieved in the absolute sense. Nevertheless, it can be achieved under certain localization restrictions. Following theorems state and prove the robustness and concavity of the neighborhood information function.

Theorem 3.1 *Let* $\Delta = max\{|y_{pi} - y_{qi}| \mid i = 1, \ldots, n;\ p, q = 1, \ldots, N;$ *and* $p < q\}$. *Let* $conv(Y)$ *be the convex hull of* Y. *If* $\sigma > \sqrt{n}\Delta$, *then* $\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma)$ *is concave over* $\mathbf{x} \in conv(Y)$.

Proof for Theorem 3.1: WLOG, let $F_{\sigma,\mathbf{Y}}(\mathbf{x}) = -\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma)$. It is equivalent to show that $F_{\sigma,\mathbf{Y}}(\mathbf{x})$ is convex when $\sigma \geq \sqrt{n}\Delta$. The elements of gradient of $F_{\sigma,\mathbf{Y}}(\mathbf{x})$ are:

$$\nabla F_{\sigma,\mathbf{Y}}(\mathbf{x})_j = -\frac{1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} \frac{(y_{pj}-x_j)}{\sigma^2} \quad \forall\ j = 1, \ldots, n. \tag{3}$$

The elements of Hessian of $F_{\sigma,\mathbf{Y}}(\mathbf{x})$ are:

$$\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jr} = \frac{-1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} \frac{(y_{pj}-x_j)}{\sigma^2} \frac{(y_{pr}-x_r)}{\sigma^2} \quad \forall\ j \neq r = 1, \ldots, n. \tag{4a}$$

$$\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jj} = \frac{1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} \left(\frac{\sigma^2 - (y_{pj}-x_j)^2}{\sigma^4} \right) \quad \forall\ j = 1, \ldots, n. \tag{4b}$$

Consider the following:

$$|\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jr}| = \frac{1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} \frac{|y_{pj}-x_j|}{\sigma^2} \frac{|y_{pr}-x_r|}{\sigma^2} \quad \forall\ j \neq r = 1, \ldots, n. \tag{5}$$

Since $\mathbf{x} \in conv(Y)$ and from the definition of Δ, the following inequalities hold:

$$|\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jr}| \leq \frac{1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} \frac{\Delta}{\sigma^2} \frac{\Delta}{\sigma^2} \quad \forall\ j \neq r = 1, \ldots, n. \tag{6}$$

and

$$\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jj} \geq \frac{1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} \left(\frac{1}{\sigma^2} - \frac{\Delta^2}{\sigma^4} \right) \quad \forall\ j = 1, \ldots, n. \tag{7}$$

Since $\sigma \geq \sqrt{n}\Delta$ for $j = 1, \ldots, n$, it implies that :

$$\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jj} > 0. \tag{8}$$

Now consider the following:

$$\begin{aligned} &\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jj} - \sum_{r=1, r\neq j}^{n} |\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jr}| \\ &\geq \frac{1}{N} \sum_{p=1}^{N} \prod_{i=1}^{n} e^{\frac{-(y_{pj}-x_i)^2}{2\sigma^2}} \frac{1}{\sigma^2} \left(1 - \frac{n\Delta^2}{\sigma^2}\right) \end{aligned} \tag{9}$$

Using $\sigma \geq \sqrt{n}\Delta$ for $j = 1, \ldots, n$ in the above equation, we get:

$$\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jj} - \sum_{r=1, r\neq j}^{n} |\nabla^2 F_{\sigma,\mathbf{Y}}(\mathbf{x})_{jr}| \geq 0. \tag{10}$$

Thus, using result from Equations (8) and (10), and from the Gershgorin disc theorem it can be stated that the Hessian is PSD, which implies that $F_{\sigma,\mathbf{Y}}(\mathbf{x})$ is convex.

□

Theorem 3.2 *Let $\sigma > 0$ be the kernel width. A cluster center* $\mathbf{x}$ *with kernel width σ provides β level information about a point p if and only if the Euclidean distance between the point and the cluster center is equal to $\sigma\sqrt{2\ln(\frac{1}{\beta})}$, where $\beta \in (0, 1]$ is the information level.*

Proof for Theorem 3.2: The proof follows immediately from the following observation:

$$\mathcal{I}(\mathbf{x}, \mathbf{y}_p, \sigma) = \prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}} = e^{\frac{-||\mathbf{y}_p-\mathbf{x}||^2}{2\sigma^2}}. \tag{11}$$

Thus, $\mathcal{I}(\mathbf{x}, \mathbf{y}_p, \sigma) = \beta$ if and only if $||\mathbf{y}_p - \mathbf{x}|| = \sigma\sqrt{2\ln(\frac{1}{\beta})}$. □

Remark 3.3 Let $\sigma > 0$ be the kernel width, and $\alpha \longrightarrow 0$ be a resolution parameter. A cluster center $\mathbf{x}$ at kernel width σ does not provide any information about a point $\overline{\mathbf{x}}$ if and only if the Euclidean distance between the point and the cluster center is greater than $\sigma\sqrt{2\ln(\frac{1}{\alpha})}$.

Proof for Remark 3.3: The proof follows immediately from Theorem 3.2. That is: $\mathcal{I}(\mathbf{x}, \mathbf{y}_p, \sigma) \leq \alpha$ if and only if $||\mathbf{y}_p - \mathbf{x}|| \geq \sigma\sqrt{2\ln(\frac{1}{\alpha})}$. □

Remark 3.4 Let C be a set of points that belong to one cluster, and $\Delta_C = max\{|y_{pi} - y_{qi}| \mid i = 1, \ldots, n;\ p, q \in C\}$. Let $\alpha \longrightarrow 0$ be a resolution parameter. If all the points that do not belong to C are $\sigma\sqrt{2\ln(\frac{1}{\alpha})}$ far away from C's cluster center and $\sigma > \sqrt{n}\Delta_C$, then $\mathcal{I}(\mathbf{x}, \mathbf{Y}, \sigma)$ is concave over $\mathbf{x} \in conv(C)$.

Proof for Remark 3.4: W.L.O.G, let r be an arbitrary cluster. Let $C_r \subseteq \mathbf{Y}$ be the set of points that belongs to the cluster. The proof follows immediately from the following representation:

$$\begin{aligned}\mathcal{I}(\mathbf{x}_1, \mathbf{Y}, \sigma) &= \frac{1}{N}\sum_{p\in C_r}\left(\prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}}\right) \\ &\quad + \frac{1}{N}\sum_{p\notin C_r}\left(\prod_{i=1}^{n} e^{\frac{-(y_{pi}-x_i)^2}{2\sigma^2}}\right).\end{aligned} \tag{12}$$

From Remark 3.3 the second term in the above representation can be discarded. Now, any point in the cluster belongs to the convex hull of the cluster points. Thus, from Theorem 3.1 the result follows. □

Remark 3.3 depicts the robustness of the neighborhood function. In the presence of noise, the robustness property assists in obtaining better clustering results by ignoring the noisy data points. Furthermore, in any clustering method, the points from one cluster will typically have no effect on the other clusters. Thus, robustness property is very suitable not only for the noisy data, but also for the typical clustering problems. Remark 3.4 highlights that if the search for a cluster center is started from a point closer to the cluster center, then the information function behaves as a concave function. This in turn implies that the second order optimization methods can be used to speed up the search process. Next, the extension of the above results to non-convex cluster shapes will be presented.

4 Proposed algorithm for non-convex clusters

If a contiguous group of points is considered as a single cluster, then the results from Section 3 can be extended to handle the non-convex clusters. The following parts of this section describes the proposed Neighborhood Information Cluster Estimation (NICE) algorithm:

4.1 NICE main phases

Zoom-in phase The key task in this phase is to identify a point with high local density. It is assumed that a neighborhood of data points with high local density will have high probability to be in a cluster. A spherical neighborhood is assumed here, and its size is proportional to the kernel width σ. At the beginning of this phase, neighborhood size is very large ($\propto \Delta$). As the algorithm proceeds, the neighborhood size is reduced until it reaches a threshold (τ_1). The idea is to search for the neighborhood starting from global level, and terminating at local level (zooming in). This phase exploits the properties presented in Remarks 3.3 and 3.4 to identify a densest neighborhood. The center of such neighborhood is considered as attractor (a densest point).

Zoom-out phase The key tasks of this phase is to identify prime neighborhood W around the attractor, and the corresponding prime kernel width δ^*. The neighborhood is searched from the attractor by increasing the kernel width from τ_1 up to τ_2 (zooming out). The prime neighborhood is the smallest spherical neighborhood around the attractor containing at least η_2 data points at ξ_1 information level. The non-existence of the prime neighborhood terminates the algorithm. On the contrary, if the prime neighborhood exists, then the explore phase is executed.

Explore phase The key task of this phase is to identify all the points that belong to a cluster. The idea is to identify all the reachable points from the prime neighborhood using DBSCAN. Identifying reachable points from any point require the information of scan radius (r_k) and minimum points (η_3). The scan radius is estimated using ξ_2 and δ^* based on Remark 3.3. Once the scan radius is estimated, data points that are reachable from all the points inside the prime neighborhood are collected into contiguous set Z_k.

Update phase If the contiguous set has enough cardinality (η_1), then the set is archived as a cluster. The data points belonging to the contiguous set are removed from the input data, and the algorithm is repeated on the updated input data. If the contiguous set does not have enough cardinality, then the algorithm terminates. This termination criterion eliminates search over low density points, which are potentially noise.

4.2 NICE implementation

The proposed implementation of the NICE algorithm identifies the clusters sequentially. Algorithm 1 presents the details of the proposed idea. In the algorithm, l represents the iteration counter for each cluster and it is re-initialized to zero before the search of a new cluster. Similarly, any previous notation with index (l) implies that its usage is related to the current cluster only. For example: $\mathbf{x}^{(l)}$ is the estimated densest point at iteration l for the current cluster.

Algorithm 1: The proposed algorithm.

input : $\mathbf{Y} \in \mathbb{R}^{n \times N}$
output: k^*, $\mathbf{Y}_t^* \quad \forall t = 1, \ldots, k^*$

Initialize: $S = \mathbf{Y}$, $k = 1$;
while $S \neq \emptyset$ **do**
  Zoom In Phase:
  Set: $l = 0$, $\Delta = max\{|y_{pi} - y_{qi}| \mid \forall i = 1, \ldots, n;\ \mathbf{y}_p,\ \mathbf{y}_q\ \in S\}$;
  Set: $\mathbf{x}^{(l)} = \text{mean}(S)$, $\sigma^{(l)} = \sqrt{n}\Delta$;
  repeat
    $\sigma^{(l+1)} = \mu_1\, \sigma^{(l)}$;
    $\mathbf{g}_{(l+1)} = -\nabla \mathcal{I}(\mathbf{x}^{(l)}, S, \sigma^{(l+1)})$;
    $\mathbf{H}_{(l+1)} = -\nabla^2 \mathcal{I}(\mathbf{x}^{(l)}, S, \sigma^{(l+1)})$;
    if $(\mathbf{g}_{(l+1)}^T [\mathbf{H}_{(l+1)}]^{-1} \mathbf{g}_{(l+1)} \leq 0)$ *or* $(|[\mathbf{H}_{(l+1)}]^{-1} \mathbf{g}_{(l+1)}| > \xi_3 \sqrt{2} \sigma_{(l+1)})$ **then**
      $\mathbf{x}^{(l+1)}$ = first-order($\mathbf{x}^{(l)}$);
    else
      $\mathbf{x}^{(l+1)}$ = second-order($\mathbf{x}^{(l)}$);
    end
    $l = l + 1$;
  until $(\sigma^{(l)} \geq \tau_1)$;
  Zoom Out Phase:
  Set: $\mathbf{z} = \mathbf{x}^{(l-1)}$, $d_p = ||\mathbf{y}_p - \mathbf{z}|| \ \forall \mathbf{y}_p \in S$;
  Set: $v = 0$, $\delta^{(v)} = \tau_1$, $W = \emptyset$;
  while $(|W| \leq \eta_2)$ *and* $(\delta^{(v)} \leq \tau_2)$ **do**
    $W = \{\mathbf{y}_p \in S : d_p \leq \delta^{(v)} \sqrt{-2 \ln(\xi_1)}\}$;
    $\delta^{(v+1)} = \mu_2 \delta^{(v)}$;
    $v = v + 1$;
  end
  Explore Phase:
  Set: $r_k = 0$, $Z_k = \emptyset$, $\delta^* = \delta^{(v-1)}$;
  if $|W| \geq \eta_2$ **then**
    $r_k = \delta^* \sqrt{-2 \ln(\xi_2)}$;
    Z_k = all-reachable($\mathbf{z}$, r_k, η_3);
  end
  Update Phase:
  if $(|Z_k| \geq \eta_1)$ *and* $((|S| \geq \eta_1)$ *or* $(k = 1))$ **then**
    $\mathbf{Y}_k = Z_k$;
    $S = S \backslash Z_k$;
    $k = k + 1$;
  else
    Return: $k^* = k$, $\mathbf{Y}_t \quad \forall t = 1, \ldots, k^*$;
    Stop;
  end
end

In Algorithm 1, when the neighborhood information function is locally concave at $\mathbf{x}^{(l)}$, a second order search method will be executed. On the other hand, if the function is not locally concave, then a first order search method is executed. The sign of $(\mathbf{g}^T[\mathbf{H}]^{-1}\mathbf{g})$ is a quick test for the local concavity of the proposed function. Further more, the condition $(|[\mathbf{H}]^{-1}\mathbf{g}| < \sqrt{2}\sigma)$ ensures that the newton step does not go beyond the current kernel width. Based on the configuration of the given data points (specifically due to symmetry), the Hessian matrix $\mathbf{H}$ can become indefinite at $\mathbf{x}^{(l)}$. For example: in Fig. 1, the data points are shown on the x-axis, and the profile of the neighborhood information function is drawn for various values of the kernel width along the y-axis. For σ_i, $i = 1, 2, 3$ the neighborhood information function is locally concave at $x^{(i)}$, for all $i = 1, 2, 3$. However, when the kernel width is reduced from σ_3 to σ_4, the function becomes locally indefinite at $x^{(4)}$. Furthermore, for lower values of the kernel width (σ_5 and σ_6), the function becomes locally convex (at $x^{(5)}$ and $x^{(6)}$ respectively).

Functions 'second-order($\mathbf{x}^{(l)}$)' executes one step of newton search starting from initial solution $\mathbf{x}^{(l)}$. The purpose of using only one step is to keep $\mathbf{x}^{(l)}$ in locally concave region of the information function. Therefore, the improvement from first newton step is enough for the above purpose, and speeds up the search process. On the other hand, function 'first-order($\mathbf{x}^{(l)}$)' will execute the gradient search starting from initial solution $\mathbf{x}^{(l)}$. The gradient search algorithm will terminate when the Hessian matrix becomes locally negative definite. The purpose of the gradient search algorithm is to drive away the current solution to a nearby locally concave region. Finally, function 'all-reachable($\mathbf{z}$, r_k, η_3)' executes a search for reachable points from the densest point $\mathbf{z}$ with search radius r_k, and minimum number of points η_3. This function is similar to DBSCAN's reachable point search function. In the next sections, numerical performance of the proposed algorithm is presented.

5 Low-dimensional performance analysis

There are myriad of clustering algorithms in the literature, and our objective is not to conduct a comprehensive comparison. Few popular algorithms that withstood the test of

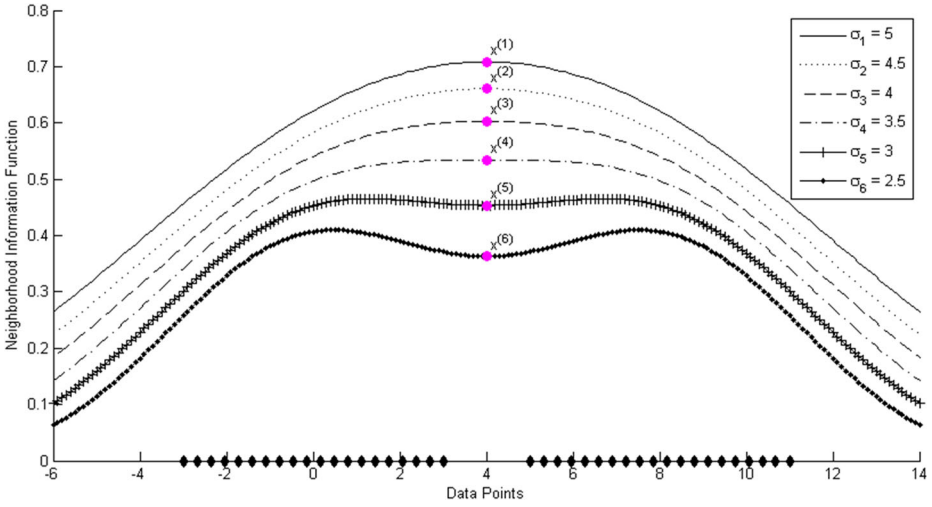

Fig. 1 Neighborhood Function Profiles

time are considered here. Specifically, the aim of these experiments is to compare the proposed approach with DBSCAN. Through the experiments, we are experimentally evaluating the significance of the proposed enhancement mechanism to DBSCAN. Synthetic data sets from the literature are used for assessing the performance of the proposed approach. In Experiment-1, the model selection is done using internal cluster validation measures. Specifically, 5 high values of ξ_1 w.r.t Silhouette measure are first selected. Then the value of ξ_1 among the top 5 values that corresponds to highest value of Calinski-Harabasz measure is selected. If all there is a tie, then S_{Dbw} measure is used for breaking ties. The cluster quality is depicted and evaluated graphically.

5.1 Experiment-1

Objective The goal of this experiment is to test the performance of the proposed algorithm on data sets defined over two or three dimensions.

Data Demographics Fig. 2 represents the data demographics. The data is taken from the following sources [22] [23].

Experimental Setup Following parameter values are set for the proposed algorithm: $\mu_1 = 0.95$, $\mu_2 = 1.01$, $\tau_1 = 0.1$, $\tau_2 = \Delta$, $\eta_1 = 10$, $\eta_2 = 10$, $\eta_3 = 3$ and $\xi_2 = 10^{-6}$. The value of parameter ξ_1 is iteratively searched over the interval $(0,\ 1)$, and using the proposed multiple internal indices the best value for ξ_1 is selected. For K-means and K-medoids algorithm, the value of K is searched iteratively over the interval $[1, 20]$ with increment of 1. The value of K that minimizes the internal index S_{Dbw} is selected as the best value for K. Similarly, for DBSCAN, the value of $MinPts$ is set to 3 (similar to $\eta_3 = 3$). However, the value of ϵ is searched iteratively over the interval $[0.1, 2]$ with increment of 0.1.

Results & Discussion The results of the proposed algorithm is compared with K-means, K-medoids, and DBSCAN algorithm (see Table 1). The cluster assignments by the proposed algorithm are illustrated using different colors in Fig. 2. As expected, K-means and K-medoids algorithm performed poorly in the presence of non-convex clusters. They also perform poorly in the data sets containing different inter and intra cluster densities. DBSCAN algorithm performs well in 4 out of 9 cases. The reason for its poor performance is the selected range of parameter ϵ. Although, there are empirical studies that estimates the value of ϵ, the studies cannot be used in the higher dimensions. Similar to parameters K and ϵ, parameter ξ_1 can be considered as the key parameter of the proposed algorithm. The range of ξ_1 is always in the interval $(0,\ 1)$ for any data set in any dimension. This is perhaps, the strength of the proposed approach over the other clustering approaches. From the experimentation, it has been observed that the other parameters are easy to set based on data and abstract subjective knowledge. Parameters like μ_1, μ_2, τ_1, and ξ_2 are related to the precision and speed of the algorithm. Value of τ_2 is always fixed to Δ. Parameters η_1, η_2, and η_3 need subjective knowledge. Similar to the estimation of $MinPts$, parallel studies can be developed for estimation of the three parameters. The choice of the internal indices selected for the experiment is due to the following reason: It is well known that different internal indices capture different cluster properties. Thus, the choice of internal indices does effect the solution. Based on our trial experiments, the proposed mechanism of using multiple internal indices works well for all the experiments.

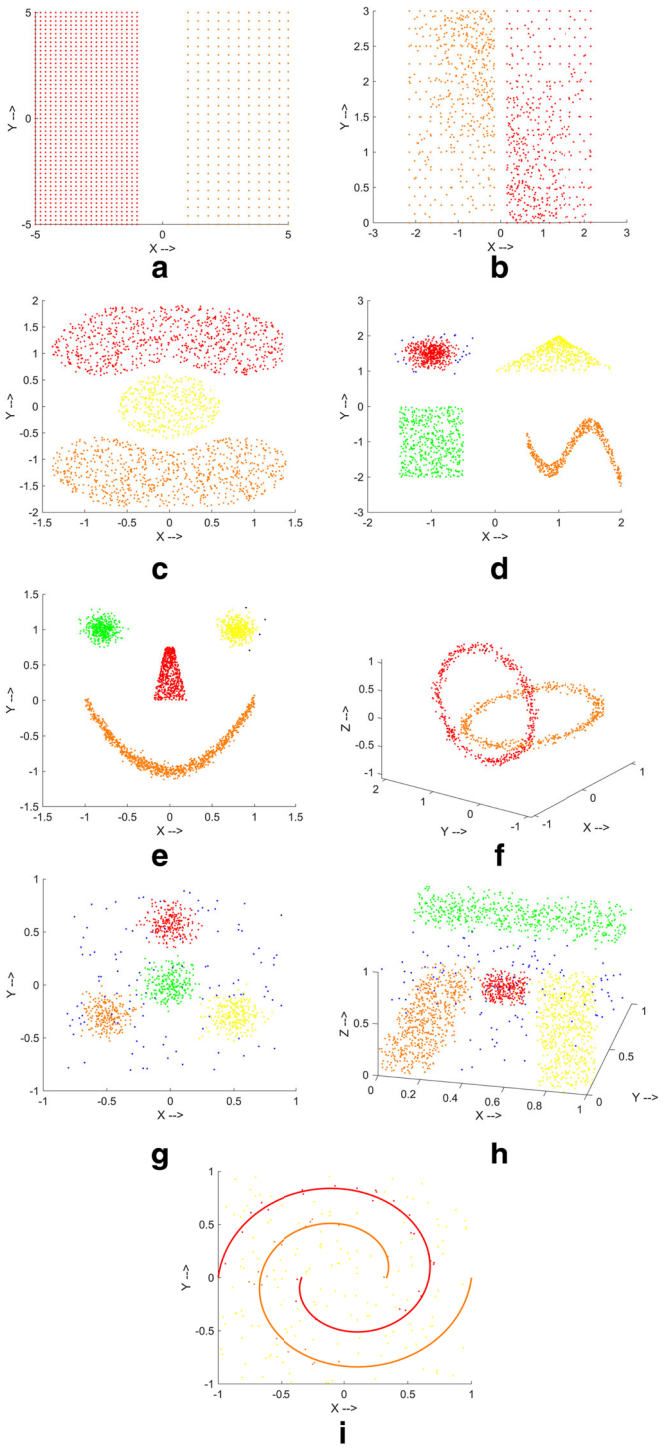

Fig. 2 Data Demographics & Results of the Proposed Approach

Table 1 Experiment-1 data and results

Data Sets	Actual # Clusters	P	n	noise %	source	NICE # Clusters	K-means # Clusters	K-mediods # Clusters	DBSCAN # Clusters
Figure 2a	2	1375	2	0	Generated	**2**	**2**	3	218
Figure 2b	2	1016	2	0	FCP	**2**	1	1	8
Figure 2c	3	2000	2	0	mlbench	**3**	2	2	**3**
Figure 2d	4	2000	2	0	mlbench	**4**	**4**	12	**4**
Figure 2e	4	2500	2	0	mlbench	**4**	5	1	**4**
Figure 2f	2	1000	3	10	FCP	**2**	10	19	4
Figure 2g	5	2000	3	10	mlbench	**5**	4	14	7
Figure 2h	5	2000	3	10	mlbench	**5**	**5**	6	16
Figure 2i	3	2000	2	10	mlbench	**3**	2	**3**	1

6 High-dimensional performance analysis

One of the bottlenecks in many clustering algorithms is their extendability to the higher dimensions. The goal in the following experiments is to test the performance of the proposed algorithm in clustering high-dimensional data. The experiments in this section focus on the ability of the algorithm to handle high dimensional data, low dimensional embedded clusters and symmetry. In these experiments, the model selection is done using multiple internal cluster validation measures (similar to Experiment-1). Specifically, 5 high values of ξ_1 w.r.t Silhouette measure are first selected. Then the value of ξ_1 among the top 5 values that corresponds to highest value of Calinski-Harabasz measure is selected. The cluster quality is depicted and evaluated via external cluster validation measures like: precision, recall, Rand, Jaccard and Folkes Mallows indices.

6.1 Experiment-2

Objective The objective of this experiment is to study the performance of the proposed algorithm in identifying low-dimensional clusters embedded in the high dimensional data.

Data Demographics In this experiment, synthetic data sets from dimensions 9 to 23 are considered. The data consists of several low-dimensional separable convex clusters in higher dimensions. Specifically, a data set of dimension n ($n \geq 3$) will have n clusters, where the cluster dimension ranges from 1 to n. That is, in a data set of dimension n there will be exactly one cluster of dimension i, where $i = 1, \ldots, n$. Cluster i will have its center at $[1 + 4(i - 1)]\mathbf{e}$, where $\mathbf{e} \in \mathbb{R}^n$ is the vector of all ones. The data points in a cluster are normally distributed around its cluster center with a variance of 1 unit. Figure 3 illustrate the orientation of the cluster for $n = 3$ data set. The demographics of the data sets are given in Table 2.

Experimental Setup For the proposed algorithm and DBSCAN, the experimental setup is similar to Experiment-1. For K-means and K-medoids algorithm, the value of K is searched iteratively over the interval $[3, 25]$ with increment of 1. The value of the key parameter (or the model selection) is done via combination of Silhouette measure and Calinski-Harabasz measure as described earlier.

Results & Discussion Although the clusters are convex and separable, the main difficulty in the above datasets is presence of lower dimensional clusters. A visual illustration of the clusters provide information related to the goodness of cluster assignment. However, illustration beyond 2 or 3 dimensions is impractical. Thus, cluster assignment in this experiment is measured using internal and external cluster measures. Table 2 presents the summary of the results on internal cluster measures obtained from the three algorithms. The best values of Calinski-Harabasz cluster measure for each data set is highlighted in bold fonts. In addition to that, the number of clusters closest to the actual number of cluster are highlighted in bold fonts. K-means algorithm was not able to detect the right number of clusters 5 out of 6 times. K-medoids on the other hand, was not able to detect the right number of clusters 6 out of 6 times. DBSCAN performs poorly in higher dimensions. Nevertheless, the proposed algorithm worked well in the presence of low-dimensional clusters in the high-dimensional data sets. For D8, although DBSCAN reports the right number of clusters, it has low value in Calinski-Harabasz measure compared to NICE. This shows that the number of clusters may not necessarily imply goodness of cluster separation. In addition to that, the goodness of clusters obtained from NICE method, were evaluated w.r.t external indices. Table 3

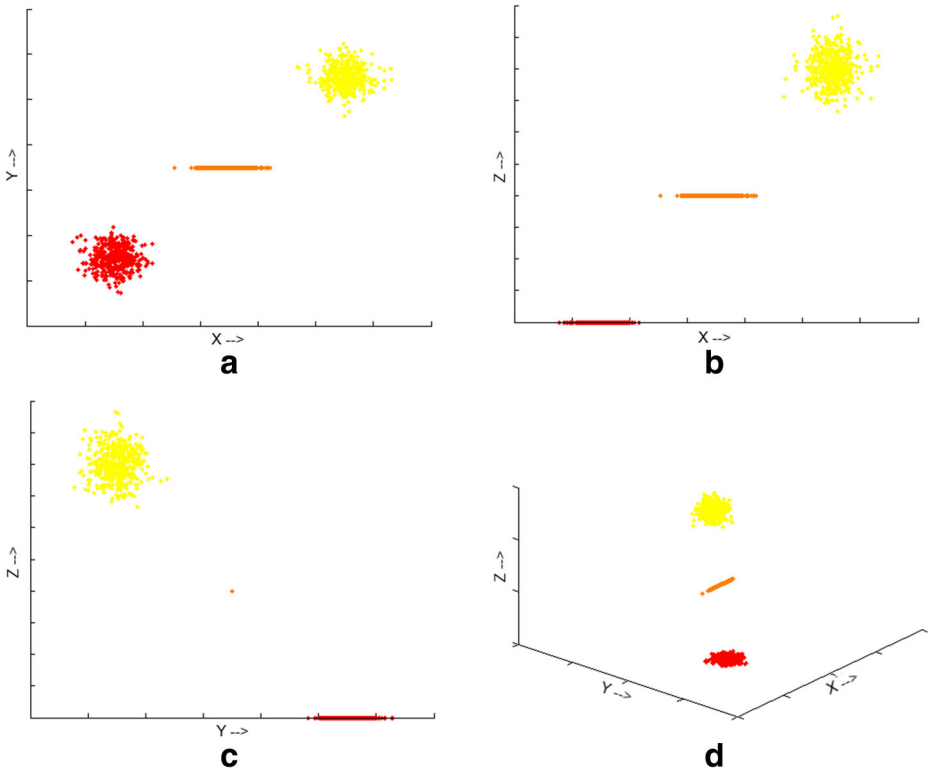

Fig. 3 Experiment-2 Orthogonal Views for $n = 3$

depicts the values of various well known external cluster measures. For all the indices except Hubert, a value closer to 1 implies good cluster separation. For Hubert index, a value closer to zero implies good cluster separation. For all the datasets, in Table 3, there is at least one value equal to 1 indicating that cluster assignment is perfect w.r.t one of the measures.

6.2 Experiment-3

Objective The objective of this experiment is to analyze the effect of symmetry and high dimensions on the performance of the proposed algorithm.

Data Demographics In this experiment, 6 high dimensional data sets available from the literature is considered [24]. Each data set contains 16 symmetric separable convex clusters and 1024 points. The demographics of the data sets are given in Table 4.

Experimental Setup The experimental setup is similar to that of Experiment-2 for all the algorithms.

Results & Discussion Table 4 presents the summary of the results obtained from the three algorithms. Values in Table 4 are highlighted in bold fonts, similar to Table 2. Although the

Table 2 Experiment-2 data and results

Data Sets	D4	D5	D6	D7	D8	D9	D10	D11
n	9	11	13	15	17	19	21	23
P	3833	4577	5127	5824	6592	7381	8146	8983
Actual # Clusters	9	11	13	15	17	19	21	23
K-means # Clusters	10	8	15	10	14	18	24	17
K-means Silhouette	0.7258	0.7665	0.7213	0.7135	0.7951	0.6886	0.6966	0.6685
K-means Calinski Harabasz	321540.4	48007.99	598962.1	72914.79	135001.9	160402.6	1595265	152656.7
K-mediods # Clusters	6	5	8	7	6	7	9	8
K-mediods Silhouette	0.7327	0.6424	0.7379	0.6692	0.6017	0.6096	0.6596	0.6245
K-mediods Calinski Harabasz	27250.75	22911.8	32828.66	32067.12	51748.54	44370.59	52168.65	48286.35
DBSCAN # Clusters	**9**	**11**	**13**	**15**	**17**	21	31	15
DBSCAN Silhouette	0.8781	0.7343	0.7494	0.7668	0.7778	0.6846	0.5631	0.6089
DBSCAN Calinski Harabasz	**355832.7**	**480966.5**	28322.77	95684.84	25523.71	6696.031	2316.836	40.5663
NICE # Clusters	**9**	**11**	**13**	**15**	16	**19**	**21**	**23**
NICE Silhouette	0.8478	0.7247	0.8797	0.8515	0.8602	0.862	0.8162	0.8399
NICE Calinski Harabasz	316804.6	395943.1	**687991.2**	**875761.2**	**275379.1**	**1453799**	**1730185**	**1981242**

Table 3 Experiment-2 NICE results with external indices

Data sets	D4	D5	D6	D7	D8	D9	D10	D11
NICE Precision	**1**	**1**	**1**	**1**	0.8759	**1**	**1**	**1**
NICE Recall	0.9981	0.9991	**1**	0.9976	**1**	0.9993	0.9986	0.9988
NICE Rand	0.9998	0.9999	**1**	0.9998	0.9915	**1**	0.9999	0.9999
NICE Hubert	-0.0003	-0.0002	**0**	**0**	-0.0001	**0**	**0**	**0**
NICE Jaccard	0.9981	0.9991	**1**	0.9976	0.8759	0.9993	0.9986	0.9988
NICE Folkes Mallows	0.9991	0.9995	**1**	0.9988	0.9359	0.9997	0.9993	0.9994
NICE Avg time (sec.)	13.13	21.26	28.89	41.49	58.12	81.74	106.43	136.57

Table 4 Experiment-3 data and results

Data sets	S1	S2	S3	S4	S5	S6
n	32	64	128	256	512	1024
P	1024	1024	1024	1024	1024	1024
Actual Clusters	16	16	16	16	16	16
K-means # Clusters	**16**	11	10	11	9	14
K-means Silhouette	0.684	0.7767	0.7084	0.8268	0.7227	0.7566
K-means Calinski Harabasz	764.0664	269.6436	194.6662	230.1584	160.5625	325.6987
K-mediods # Clusters	10	12	12	14	14	14
K-mediods Silhouette	0.705	0.7475	0.8006	0.7363	0.8878	0.9201
K-mediods Calinski Harabasz	245.2719	273.1301	297.3412	309.3855	503.4606	531.9228
DBSCAN # Clusters	10	0	2	0	2	0
DBSCAN Silhouette	0.8462	0.8462	0.6357	0.6357	0.6439	0.6439
DBSCAN Calinski Harabasz	4.1968	4.1968	11.9487	11.9487	3.6804	3.6804
NICE # Clusters	17	**16**	**16**	**16**	**17**	**16**
NICE Silhouette	0.8396	0.8564	0.8664	0.8681	0.8718	0.9909
NICE Calinski Harabasz	**2286.92**	**2777.432**	**27423.35**	**9990.46**	**2622.202**	**718469.8**

Table 5 Experiment-3 NICE results with external indices

Data sets	S1	S2	S3	S4	S5	S6
NICE Precision	0.9874	**0.9992**	**0.9999**	**0.9994**	**0.9905**	**1**
NICE Recall	0.9442	0.9845	**0.9906**	0.9865	0.9508	**1**
NICE Rand	**0.9958**	**0.9990**	**0.9994**	**0.9991**	**0.9964**	**1**
NICE Hubert	0.1335	−0.0998	−0.0934	−0.0978	−0.136	−0.0836
NICE Jaccard	0.9329	0.9837	**0.9905**	0.9859	0.9422	**1**
NICE Folkes Mallows	0.9655	**0.9918**	**0.9952**	**0.9929**	0.9704	**1**
NICE Avg time (sec.)	3.43	4.33	6.48	15.67	43.99	174.82

clusters were convex without any noise, K-means and K-medoids were not able to detect the right number of clusters. This is due to the high dimensions. In addition to that, for DBSCAN the range of ϵ did not work for all the data sets. Although one can argue that the range can be altered manually, no automated estimation for the range is available in the literature. The proposed algorithm was able to identify right number of clusters, and in couple of cases it overestimated the number. For S1, although k-means reports the right number of clusters, it has low value in Calinski-Harabasz measure compared to NICE. This shows that the number of clusters may not necessarily imply goodness of cluster separation. In addition to that, the goodness of clusters obtained from NICE method, were evaluated w.r.t external indices. Table 5 depicts the values of various well known external cluster measures. For all the indices except Hubert, a value closer to 1 implies good cluster separation. For Hubert index, a value closer to zero implies good cluster separation. For all the datasets, in Table 5, there is at least one value above 0.99 indicating that cluster separation is almost perfect w.r.t one of the measures.

7 Conclusions

An optimization based enhancement to DBSCAN is proposed in this paper. The primary parameter of the proposed algorithm, ξ_1, is always in the interval $(0,\ 1)$. The interval range is absolute, and will not change w.r.t data characteristics or data dimensions. Thus, estimating scan radius from ξ_1 is the key novelty of the proposed algorithm. Furthermore, the usage of second-order mechanism in the zoom-in phase speeds up the densest point search. Termination criterion ($|Z_k| \geq \eta_1$) of the proposed algorithm eliminates the search over the noisy data. Based on the illustrated experiments, it can be seen that the other parameters are easy to estimate based on the abstract knowledge of the given data. From the numerical experimentation, it can be concluded that the algorithm performs well in separable (convex or non-convex) clusters, with or without noise scenarios. To sum, it will be a good competitor for the existing clustering algorithms, and may provide alternative insights in the cluster analysis.

Acknowledgements The author would like to acknowledge the research support provided by King Fahd University of Petroleum & Minerals (KFUPM).

References

1. MacQueen, J. et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, pp. 281–297 (1967)
2. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)
3. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
4. Gan, G., Ma, C., Wu, J.: Data clustering: theory, algorithms, and applications, vol. 20. Siam (2007)
5. Yang, M.-S.: A survey of fuzzy clustering. Math. Comput. Modell. **18**(11), 1–16 (1993)
6. Kriegel, H.-P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1**(3), 231–240 (2011)
7. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, pp. 226–231 (1996)
8. Campello, R.J.GB., Moulavi, D., Zimek, A., Sander, J.: A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. Data Min. Knowl. Disc. **27**(3), 344–371 (2013)
9. Mount, D.M.: http://www.cs.umd.edu/~mount/ANN/ (2010)
10. Sander, J., Ester, M., Kriegel, H.-P., Xu, X.: Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data Min. Knowl. Discov. **2**(2), 169–194 (1998)
11. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: Optics: ordering points to identify the clustering structure. In: ACM Sigmod record. ACM, pp. 49–60 (1999)
12. Aggarwal, C.C., Yu, P.S.: Finding generalized projected clusters in high dimensional spaces. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. ACM, pp. 70–81 (2000)
13. Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. J. Amer. Stat. Assoc. **97**(458), 611–631 (2002)
14. Spurek, P., Tabor, J., Byrski, K.: Active function cross-entropy clustering. Expert Syst. Appl. **72**, 49–66 (2017)
15. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of the 2003 SIAM international conference on data mining. SIAM, pp. 47–58 (2003)
16. Azzalini, A., Torelli, N.: Clustering via nonparametric density estimation. Stat. Comput. **17**(1), 71–80 (2007)
17. Azzalini, A., Menardi, G., et al.: Clustering via nonparametric density estimation: The r package pdfcluster. J. Stat. Softw. **57**(11), 1–26 (2014)
18. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
19. Tabor, J., Spurek, P.: Cross-entropy clustering. Pattern Recogn. **47**(9), 3046–3059 (2014)
20. Sander, J.: Density-based clustering, pp. 270–273. Springer US, Boston (2010). https://doi.org/10.1007/978-0-387-30164-8_211
21. Celebi, M.E.: Partitional clustering algorithms. Springer (2014)
22. Ultsch, A.: Clustering with som: U*c. In: Proceedings of the 5th Workshop on Self-Organizing Maps, vol. 2, pp. 75–82 (2005)
23. Leisch, F., Dimitriadou, E.: Package 'mlbench' (2013)
24. Franti, P., Virmajoki, O., Hautamaki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1875–1881 (2006)