Depth based local center clustering

SIYI WANG

2022/10/21

1 Background

1.1 Data depth

Ordering is an important concept in analyzing data. In one dimensional case, the arrangement of real numbers provides a natural ordering. However, in multivariate case, there is no natural ordering of vectors. Data depth provides a non-parametric approach to give a central-outward ordering of data.

Assume a data set $\mathbf{X}$ follows a distribution F in a d dimensional real space, $\mathbb{R}^d$, a depth function for $\mathbf{X}$ is defined as

$$D(\bullet, F) : \mathbb{R}^d \to [0, 1] : \boldsymbol{z} \mapsto D(\boldsymbol{z} \mid F),$$

where $\boldsymbol{z} \in \mathbb{R}^d$. The following restrictions should, in theory, be satisfied by data depth [1]:

1. Affine invariance,
2. Null at infinity,
3. Monotone on rays,
4. Upper semi-continuous.

In real applications, the inherent distribution F is unknown, and hence depth is calculated by the empirical distribution of $\mathbf{X}$. For a more intuitive notation, depth function is written as $D(\boldsymbol{z} \mid \mathbf{X})$ instead of $D(\boldsymbol{z} \mid F)$ in this report.

1.2 Literature Review

Numerous applications of data depth have been developed since Tukey's 1975 introduction of the first depth function, such as multivariate density estimation [2], tests for multivariate scale difference [3], robust and affine equivariant estimations of location [4], classification [5, 6, 7] and so on.

Much research has been done using data depth in classification, while less attention is given to the problem of clustering. Jornsten [5] proposed an algorithm called DDclust, improves the output of other clustering algorithms. It takes as input the result from a clustering algorithm like partition around medoids (PAM) clustering and calculates

the depths of observations regarding each cluster based on the initial partition to see if relocating some observations can achieve a better outcome. In 2016, Jeong et al. [8] proposed a density-based clustering algorithm called data depth based clustering analysis (DBCA), which can be regarded as an alternative version of the well-known DBSCAN algorithm [9]. DBCA modifies Mahalanobis depth to a local version as follows:

$$D_M(\boldsymbol{z} \mid \boldsymbol{x}_i) = \left[1 + (\boldsymbol{z} - \boldsymbol{x}_i)'\hat{\Sigma}^{-1}(\boldsymbol{z} - \boldsymbol{x}_i)\right]^{-1}, \tag{1}$$

where $\boldsymbol{z}$ denotes a point in $\mathbb{R}^d$, $\boldsymbol{x}_i$ is an observation in $\mathbf{X}$, while $\hat{\Sigma}$ is the estimated covariance matrix of data set $\mathbf{X}$. Compared with DBSCAN, DBCA utilizes (1) to measure similarities among observations instead of Euclidean distance. The main advantages of DBCA are affine invariance and robustness to noises.

So far, the notion of depth serves an auxillary role to the more traditional clustering methods, and is not employed directly in examining the multimodality of data. In this report, we propose an innovative clustering algorithm that relies on the center-outward ranks given by data depth, named Depth-based Local center Clustering (DLCC). DLCC determines first the neighbors for each point, and then computes depth values regarding the subset constructed by the point and its neighbors. Data depth can be used to locate centers of those subsets, which are designated as local centers. The method then detect inherent clusters by analyzing the local centers.

2 Algorithm

Data depth describes data from a global perspective. To include the concept of data depth into clustering, it is necessary to examine depth based on some subsets of the entire data set, where we refer to the depth calculated with regard to a subset as *local depth* (with respect to the subset). In DLCC, n subsets built by each observation and its neighbors are examined, where n is the total number of observations in a data set $\mathbf{X}$. Fix a size s of subset, neighbors of each observation are defined as the top $s-1$ similar points. Possible approaches for building similarity matrix and defining neighbors are introduced in Section 2.1. Subset form by the observation $\boldsymbol{x}_i$ and its neighbors is noted as $\boldsymbol{X}_i$. Data depth values are used to define the rank of any observation in $\boldsymbol{X}_i$.

Definition 2.1. *Note $\boldsymbol{x}_j$ is an observation in data set $\boldsymbol{X}_i$. The rank of $\boldsymbol{x}_j$ in $\boldsymbol{X}_i$ is defined as Equation (2).*

$$r_{\boldsymbol{X}_i}(\boldsymbol{x_j}) = \#\left\{\boldsymbol{x}_q : D\left(\boldsymbol{x}_j \mid \boldsymbol{X}_i\right) < D\left(\boldsymbol{x}_q \mid \boldsymbol{X}_i\right), \boldsymbol{x}_q \in \boldsymbol{X}_i\right\} + 1. \tag{2}$$

If $r_{\boldsymbol{X}_i}(\boldsymbol{x_j}) = 1$, $\boldsymbol{x_j}$ is the center of $\boldsymbol{X}_i$. Local centers are defined to be the centers of the subsets $\boldsymbol{X_i}$'s. If the center of $\boldsymbol{X}_i$ is $\boldsymbol{x_i}$, then $\boldsymbol{x_i}$ is said as the strict local center.

Local centers can be interpreted as representative points of $\mathbf{X}$, and the quality of them is crucial in following steps. Hence, before grouping local centers, it is necessary to filter them. For convenience, the local centers after filtering are called filtered centers. The algorithm of filtering local centers is explained in Section 2.2. After filtering, the local centers are grouped. One possible approach is based on the proportion of overlapped observations between neighbors of two local centers, and this proportion is called as the similarity between local centers. Here, a new parameter δ, the similarity threshold, is introduced. Based on δ, two strategies of DLCC named "min" and "max", dealing with different scenarios of clustering problems, are summarized as follows:

Definition 2.2. *Let $\{c_1, \cdots, c_P\}$ be a finite set of filtered centers. The grouping result under two strategies should satisfy following requirements.*

- ***Min strategy**: $\forall c_i$ and c_j in the same group, there is $\mathrm{sim}(c_i, c_j) > \delta$.*
- ***Max strategy**: $\forall\ c_i$ in group $\mathbf{g}$, $\exists c_j$ in the same group s.t. $\mathrm{sim}(c_i, c_j) > \delta$, and for $\forall c_v \notin \mathbf{g}$, $\mathrm{sim}(c_i, c_v) \leq \delta$.*

Some insights into the strategy chosen for grouping local centers are significant in applications. Under min strategy, DLCC is a centroid-based clustering method, which attempts to find centroids of latent clusters from filtered centers. The min strategy assumes the sizes of all latent clusters are similar, and the parameter s under min strategy should be close to the real size of latent clusters. On the other hand, the min strategy may potentially enounter the multiple assignment problem. For instance, there may be a collection of filtered centers $\{c_p\}$ so that the only similarities larger than δ are between c_1 and each of the other centers, while the similarities between the other pairs in the collection are all less than δ. Then c_1 will be assigned into multiple groups by this strategy. In this case, we simply drop c_1 from the grouping consideration due to the ambiguity.

The max strategy shows a resemblance with DBCA. It assumes each cluster can be built by connecting multiple finite convex shapes created by filtered centers. Filtered centers in the same group are connected with a chain of similarities. For example, if two filtered centers c_w and c_y are in the same group, then there must exist a chain of filtered centers, $c_{p_1}, c_{p_2}, \cdots, c_{p_h}$, where $c_{p_1} = c_w$, $c_{p_h} = c_y$, and the similarity between any two adjacent centers in that chain is larger than δ. Generally, the parameters s and δ are comparatively small under the max strategy. Unlike DBCA, which builds clusters with the chain of small ellipses with fixed shape, and the center of each ellipse is each observation, DLCC with the max strategy only considers chains of filtered centers. Moreover, the number of neighbors of filtered centers is fixed rather than defined in a fixed region. DLCC can avoid a major shortcoming of DBCA, which has difficulty in handling clusters with different densities, due to the fixed shape of the ellipses. Similar to density-based clustering methods, max strategy is preferable when the latent clusters are disjoint, and it can handle non-convex and varying size clustering problems.

The number of clusters, denoted K, can then be determined, following the grouping of the filtered centers. Let G be the number of groups obtained from the filtered centers using one of the grouping strategies, then $K \leq G$. A *unique neighbor* of a group is defined as an observation that can only be a neighbor for centers in this particular group. If a certain group has no unique neighbor, then it will be dropped and leads to $K < G$. We obtain the temporary clusters from the unique neighbors of filtered centers in each of the groups. Then, we can define the scores of observations relative to each cluster, and establish cutoff points to determine whether to assign the observations to the cluster they obtain highest score with. In Section 2.3 the clustering procedure is presented in detail.

At this stage, the number of clusters has been determined, and we expect that the majority of observations have been assigned to one of the clusters. There may still be points remaining outside of the clusters. In this situation, the clustering problem now can be regarded as a classification problem. Intuitively, we can utilize classification methods to assign the remaining points to one of the clusters. By default, DDLC adopts the most elementary classifier based on data depth, named maximum depth classifier, i.e., assign the observation to the cluster which it obtains the largest depth value. It should be noted that the choice of classification methods is very flexible here. In fact, any classification method may be considered for this step.

Lastly, self-improvements procedures of DLCC are discussed. Although the clustering problem is converted to a classification problem in the last stage, it has some special features, because the observations that are already labelled may still be updated. Here we introduce a logical parameter "maxdepth", which has the potential to revise clustering results.

Definition 2.3. ***maxdepth****: A logical parameter controls whether the algorithm loops until all observations in the allocated cluster have the highest depth values.*

Let the K clusters constructed so far be denoted by $\mathcal{X}_k$, where $k = 1, 2, \cdots, K$. For any observation x_j, the depth value of it with respect to each cluster is then given by $D(x_j \mid \mathcal{X}_k)$. If the depth value of x_j in current cluster does not equal to $\max_k D(x_j \mid \mathcal{X}_k))$, and if maxdepth is TRUE, then x_j will be relocated. This procedure loops until all observations are assigned to the cluster where they obtain the largest depth value. For min strategy, there is one more step for self-improvements. Generally, for a centroid-based clustering algorithm, each cluster should have just one centroid, while the number of filtered centers can be larger than the number of clusters. Therefore, it is natural to find the deepest filtered center for each cluster, and use them to re-produce clusters until the clustering result remains unchanged.

The entire DLCC algorithm is summarized below:

Step 1: Build a $n \times n$ similarity matrix, and locate $s - 1$ neighbors for each observation.

Step 2: Generate n subsets based on step 1, and find all local centers.

Step 3: Obtain filtered centers.

Step 4: Compute the similarities among filtered centers, and divide them into G groups based on min/max strategy.

Step 5: Determine the number of clusters K ($K \leq G$) based on groupings of filtered centers and scores of observations with regard to clusters. Then build temporary clusters which contain majority observations.

Step 6: Classify the remaining observations using classification methods, such as maximum depth classifier, mixture-model based classification and so on.

Step 7: If "maxdepth" is TRUE, loop until all observations in the allocated cluster have the highest depth values.

Step 8: Under min strategy, re-define centroids $\{c_1, \cdots, c_K\}$ which satisfy $D\left(c_k \mid \mathcal{X}_k\right) = \sup D\left(\boldsymbol{C} \mid \mathcal{X}_k\right)$, where $\boldsymbol{C}$ is the set of filtered centers. Loop from step 5 until the result does not change.

2.1 Similarity matrix building

To build the similarity matrix for searching neighbors of each observation, we use the Mahalanobis depth matrix in DBCA. Sample Mahalanobis depth function is defined as Equation (3) in $\mathbb{R}^d$ space with regard to data set $\mathbf{X}$.

$$D_M(\boldsymbol{z} \mid \mathbf{X}) = \left[1 + (\boldsymbol{z} - \bar{\mathbf{X}})' \hat{\Sigma}^{-1} (\boldsymbol{z} - \bar{\mathbf{X}})\right]^{-1}, D_M(\boldsymbol{z} \mid \mathbf{X}) \in [0, 1], \tag{3}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ and, $\bar{\mathbf{X}}$ and $\hat{\Sigma}$ represent the sample mean vector and estimated covariance matrix respectively. In order to obtain a robust result about neighbors of each observation, the covariance matrix here is estimated with the Minimum Covariance Determinant (MCD) method [10]. It can be seen that the modified equation in DBCA [8]

$$D_M(\boldsymbol{z} \mid \boldsymbol{x}_i) = \left[1 + (\boldsymbol{z} - \boldsymbol{x}_i)' \hat{\Sigma}^{-1} (\boldsymbol{z} - \boldsymbol{x}_i)\right]^{-1}$$

uses observation x_i, $i = 1, 2, \cdots, n$ rather than the mean vector in calculating depth value. That is, the center for calculating Mahalanobis depth is translated to each observation and the depth values of other observations are recorded. Hence, the modified Mahalanobis depth value shows the similarity to the observation in the center. Obviously, the right hand side of (1) can be written as $[1 + d^2_{Mah}(\boldsymbol{z}, \boldsymbol{x}_i)]^{-1}$, where $d_{Mah}(\boldsymbol{z}, \boldsymbol{x}_i)$ is the Mahalanobis distance between $\boldsymbol{z}$ and $\boldsymbol{x}_i$, and the similarity matrix is equivalent to the distance matrix of Mahalanobis distance (larger similarity value means smaller Mahalanobis distance).

2.1.1 Min strategy

Under min strategy, a proper similarity matrix is critical; However, the estimated global covariance matrix may not be suitable enough for observations in different latent clusters. From the strong relationship between Mahalanobis distance and Principal Component

Analysis (PCA), the squared Mahalanobis distance equals the sum of squares of standardised principal component scores [11]. If PCA can use the top few principle components (PCs) to explain the bulk of the variances, then the distance determined in the space constructed by the top PCs will resemble the Mahalanobis distance, and the PCA result can indicate strong correlations exist between variables. In contrast, there is no solid evidence that the Mahalanobis distance is superior to the standard Euclidean distance when there are no apparent correlations between variables. A simple test based on PCA is presented below, to determine whether the estimated global covariance matrix should be utilized to generate the similarity matrix.

Let $\mathbf{v} = \{v_1, v_2, \cdots, v_U, \cdots, v_d\}$, which contains the proportion of variance explained by each PC from high to low, where U indicates the number of PCs that explain over 5% variance. If $U = d$, then U redefined to be $d - 1$. If either Equation (4) or Equation (5) is satisfied, the data set passes the test.

$$v_1 > 0.6, \tag{4}$$

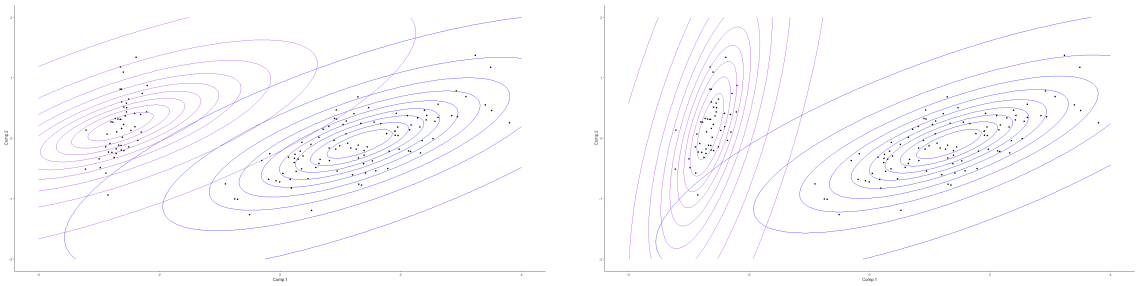
$$\sum_{u=1}^{U} v_u > 0.95. \tag{5}$$

Even if a data set passes the test, a fixed covariance matrix may not be sufficient. Figure 1a shows an example from the iris data. With the fixed covariance matrix, it appears that the purple contour does not correspond to the orientation of the left side observations. Therefore, we investigate rotating the global covariance matrix based on the data distribution and utilize the Gaussian Mixture Model (GMM) concept. The estimated global covariance matrix can be decomposed as Equation (6).

$$\hat{\Sigma} = \lambda \Gamma \Delta \Gamma', \tag{6}$$

where $\lambda = |\hat{\Sigma}|^{1/d}$, Γ is an eigenvector matrix and Δ is a diagonal matrix containing the eigenvalues, and $|\Delta| = 1$ [12]. They represent the size, the orientation and the shape of the covariance matrix. To rotate the estimated covariance matrix, we keep λ and Δ, and only adjust Γ. Then it is similar with a EEV model in Gaussian parsimonious clustering models (GPCM) family. Because λ and Δ are given, the number of free covariance parameters is d less than in an EEV model, and we call it an Alternative EEV (AEEV) model. AEEV model gives two estimated covariance matrices for iris data. Figure 1b shows two similarity

contours with the same shape and size, but different orientations for two selected data points.



(a) The similarity contour plots in two selected points based on the global covariance matrix estimated by MCD method.

(b) The similarity contour plots in two selected points based on the corresponding covariance matrices given by AEEV GMM.

Figure 1: Contour plots of the similarity measurement in the top two PCs of iris data set based on Mahalanobis depth.

If the data set does not pass the test, there are two scenarios. One is that there are significant correlations among variables within each latent cluster, but these correlations cannot be shown globally; the other is that there is no strong correlation among variables even within each latent cluster. Now, we try EEV/EEI model from GPCM family. The advantage of the EEV/EEI model is that it estimates clusters with identical eigenvalues. Therefore, we can conduct the same test using the eigenvalues provided by the EEV/EEI model. If it still fails, we assume the second scenario is correct and simply use Euclidean distance to construct the similarity matrix. To keep the value between 0 and 1 when using the Euclidean distance, we define the (z, x_i)th entry of the similarity matrix as $[1 + d_{Eu}(\boldsymbol{z}, \boldsymbol{x}_i)]^{-1}$, where d_{Eu} is the Euclidean distance. Otherwise, we accept the first scenario and construct the similarity matrix using the covariance matrices provided by the EEV/EEI model.

Those steps can be summarized as trade-offs between GMM and DLCC. If GMM explains too much data, the size of covariance matrices given by GMM will be too small for measuring similarities among observations. Hence, if there is no significant difference in model performance (such as BIC) between GMM results with different numbers of components, we prefer the model with fewer components. Furthermore, if the similarity matrix is constructed from multiple covariance matrices provided by AEEV/EEV/EEI GMM, then similarities between observations are not mutual if they do not belong to the same cluster according to the GMM outcome, i.e., $D_M(\boldsymbol{x}_i \mid \boldsymbol{x}_j) \neq D_M(\boldsymbol{x}_j \mid \boldsymbol{x}_i)$.

Because neighbors of each observation are determined by order statistics, the non-mutual relationship in similarity is acceptable.

2.1.2 Max strategy

In terms of max strategy, s is small, and the number of filtered centers is large. Its clustering outcome is less sensitive than that of the min strategy for the similarity matrix. Furthermore, the mutual relationship for similarity between observations is useful in subsequent analyses of clustering performance under the max strategy. Thus, for the construction of the similarity matrix, we just employ the estimated global covariance matrix.

2.2 Local center selections

Locating and filtering local centers are the core part of DLCC. The chosen depth function is the Mahalanobis depth, where the covariance matrix is estimated using MCD. The point with the greatest depth value in a subset represents the center of that subset. For subsets $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$, n local centers can be found, noted as $\mathbf{a} = \{a_1, a_2, \cdots, a_n\}$. Owing to the robustness of data depth, if certain subsets are comparable, their deepest point may be same. Then we can define the frequency of local center as f, which stands for the times of an observation appears in $\mathbf{a}$.

Some local centers cannot contribute to DLCC. For example, an outlier which locates between two latent clusters can be the deepest point of a subset containing observations from both clusters. To filter local centers, two benchmarks are proposed.

- A local center should locate at the central position of its own neighbors.
- A local center with higher f is more reliable.

The first step of filtering local centers is to pick up local centers satisfying $r_{\boldsymbol{X}_i}(\boldsymbol{x_i}) \leq 2$ (Definition 2.1), which aims to meet the first benchmark. Due to the restricted number of strict local centers, the clustering result will be significantly impacted if the outcome of strict local centers is influenced by randomness from calculations. Therefore, the requirement of a central position is relaxed from $r_{\boldsymbol{X}_i}(\boldsymbol{x_i}) = 1$ to $r_{\boldsymbol{X}_i}(\boldsymbol{x_i}) \leq 2$. For the remaining steps in filtering local centers, there are some differences between min and max strategies.

2.2.1 Min strategy

After the first step filtering, there are now t local centers remaining. Index these local centers in the decreasing order of their frequencies, i.e.

$$f_1 \geq f_2, \cdots, \geq f_t,$$

Then the final result of the filtering step for min strategy will consists of the first $P \leq t$ of these centers $\{c_1, \ldots, c_P\} \subseteq \mathbf{a}$, respectively with frequencies $f_1, \ldots, f_P$. To determine P, we consider the cumulative proportion of neighbors for the local centers down the list, as in (7).

$$\mathcal{P}_R = \frac{\#\{\bigcup_{p=1}^{R} \boldsymbol{X}_{c_p}\}}{n}. \tag{7}$$

Figure 2 plots the example for a simulated data set. Intuitively, if there is a significant jump in the cumulative proportion of a local center and all local centers after that jump contribute little to the cumulative proportion, then the particular local center is a potential cut-off point.

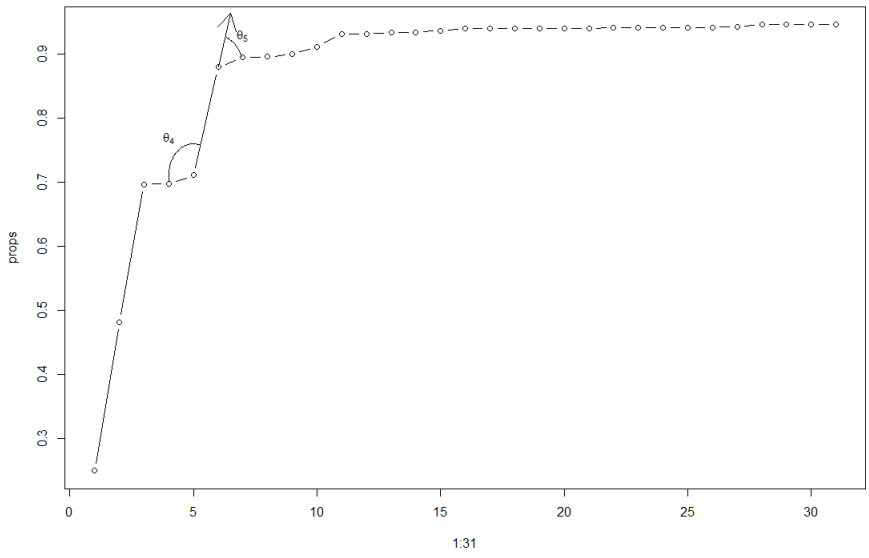


Figure 2: Cumulative proportion for 31 local centers from a simulated data set with 1000 observations.

The steps of searching the last significant jump when $t > 3$ are summarized below:

- Let $\Theta = \{\theta_1, \theta_2, \cdots, \theta_{t-2}\}$ be the set consisting of intersection angles between lines

connecting adjacent points in cumulative proportion plot. Calculate $\tan\Theta$, where $\tan\theta_R = \frac{(\mathcal{P}_{R+1}-\mathcal{P}_R)-(\mathcal{P}_{R+2}-\mathcal{P}_{R+1})}{1+(\mathcal{P}_{R+2}-\mathcal{P}_{R+1})(\mathcal{P}_{R+1}-\mathcal{P}_R)}$.

- If there are θs satisfying $\tan(\theta) < 0$ and $|\tan(\theta)| > \text{Med}(|\tan\Theta|)$, sort those $|\tan(\theta)|$ in decreasing order. Otherwise, sort all $|\tan(\theta)|$ in decreasing order. The permutation of $|\tan(\theta)|$ defined here is written as $(R_1, R_2, \cdots, R_\tau)$, where $\tau = t-2$ in the latter case.

- Set

$$\mathbb{L} = \{|\tan(\theta)|_{R_1} - |\tan(\theta)|_{R_2}, \ldots, |\tan(\theta)|_{R_{\tau-1}} - |\tan(\theta)|_{R_\tau}\}.$$

  For $|\tan(\theta)|_{R_r} - |\tan(\theta)|_{R_{r+1}} = \max(\mathbb{L})$, $P = R_r + 2$ if $\tan\theta_{R_r} < 0$; Else, $P = R_r + 1$.

In the example of Figure 2, $\tan(\theta_4) < 0$, which indicates the slope of the line connecting c_4 and c_5 is smaller than the slope of the line connecting c_5 and c_6. It means the jump right after c_5 is more significant than the jump right before c_4. Moreover, as the number of local centers increases, the jump that may discover a latent cluster will become progressively smaller. If all clusters have been found, then adding other local centers will contribute little to the cumulative proportion. Hence, we locate the cut-off point c_P by finding the largest gap in order statistics of $|\tan(\theta)|$. When $t \leq 3$, the number of latent clusters are assumed to be less or equal to t, P is simply determined by satisfying $\mathcal{P}_P = \min\{\mathcal{P}_R | \mathcal{P}_R > 0.7, R = 1, \cdots, t\}$.

2.2.2 Max strategy

With the same notions of Section 2.2.1, for max strategy, all remaining local centers satisfying $f > 1$ included. Moreover, for any local center a_y with $f = 1$, if $\bigcup_{p=1}^{R} \boldsymbol{X}_{c_p} \cap \boldsymbol{X}_{a_y} = \emptyset$ where R is the current number of filtered centers, it will be selected as well. The reason for this step is that if a cluster is constructed by subsets with non-convex shapes (such as ring and spiral), which contradicts the assumption of the max strategy, then it is possible that no local center in that cluster will remain after filtering because Mahalanobis depth is a convex depth function [1].

2.3 Clustering in DLCC

After unique neighbors for each group of local centers are assigned to temporary clusters, the score of observations to clusters are computed.

2.3.1 Min strategy

For the min strategy, the score accounts for both clustering and assessing if each observation's clustering result is acceptable. Let $\boldsymbol{C}$ denote the set of filtered centers, then

$$\text{score}_{i|k} = \frac{\sup D_M(\mathbf{x_i} \mid \boldsymbol{C}_{g=k}) - \sup D_M(\mathbf{x_i} \mid \boldsymbol{C}_{g\neq k})}{\max\{\sup D_M(\mathbf{x_i} \mid \boldsymbol{C}_{g=k}), \sup D_M(\mathbf{x_i} \mid \boldsymbol{C}_{g\neq k})\}}. \tag{8}$$

Equation (8) is the score function for observation $\mathbf{x_i}$ with respect to temporary cluster k under the min strategy. The score value is between -1 and 1. A positive score illustrates that compared with the nearest filtered center in other groups, the similarity between $\mathbf{x_i}$ and the nearest filtered center in group k is larger.

For each cluster, two score pools are generated. The first is denoted $\mathcal{S}$, which consists of the unique neighbors in this cluster and their scores, while the other, denoted $\hat{\mathcal{S}}_k$, contains the observations whose scores are positive with respect to this cluster, and their scores. Although unique neighbors are relatively trustworthy, some of them may not be reliable enough. A unique neighbor, for instance, can be the $s-1$th similar point of a filtered center in group k, while also being the sth similar point of a filtered center in another group. Observations in $\mathcal{S}_k$ whose scores are smaller than the median score in $\hat{\mathcal{S}}_k$ are shifted to $\hat{\mathcal{S}}_k$ to filter out less dependable points. Then, for scores in $\hat{\mathcal{S}}_k$, a cut-off point will be determined. All observations with scores above the cutoff point will be moved to $\mathcal{S}_k$. To determine the cutoff point, scores in $\mathcal{S}_k$ are sorted in decreasing order as $\{\gamma_1, \gamma_2, \cdots, \gamma_E\}$, where E represents the number of scores in $\mathcal{S}_k$. One candidate of the cut-off point is γ_e which satisfies Equation (9), where $\lfloor * \rfloor$ is the floor function.

$$\gamma_e - \gamma_{e+1} = \max\{\gamma_{\lfloor E/2 \rfloor} - \gamma_{\lfloor E/2 \rfloor+1}, \ldots, \gamma_{E-1} - \gamma_E\}. \tag{9}$$

Another candidate for the cut-off point can be used when $E < s/2$. It is defined as the $1 - \frac{s/2-E}{\#\{\hat{\mathcal{S}}_k\}}$ quantile of scores in $\hat{\mathcal{S}}_k$. This prevents a temporary cluster from having an insufficient amount of observations, and it can ensure that a temporary cluster contains at

least $s/2$ observations. It is necessary to note that, if $E + \#\{\hat{\mathcal{S}}_k\} < s/2$, DLCC under min strategy will fail because it violates the premise that all clusters are of comparable size. The cutoff point is specified as the lowest candidate value. After transferring observations with scores larger than the cutoff point in $\hat{\mathcal{S}}_k$, the temporary cluster k is updated as current observations in $\mathcal{S}_k$.

2.3.2 Max strategy

The score in this section is only for overlapping neighbors between filtered centers in different groups. Let $\boldsymbol{C}_k$ be the set containing filtered centers in group k, then the score is defined as follows:

$$\text{score}_{i|k} = \frac{\#\{c \mid \mathbf{x_i} \in \boldsymbol{X}_c,\ c \in \boldsymbol{C}_k\}}{\#\{\boldsymbol{C}_k\}}. \tag{10}$$

All observations that are both neighbors of filtered centers in different groups are allocated to the cluster with the highest score. Observations that remain unlabeled are not neighbours of any filtered center.

2.3.3 Classification and "maxdepth"

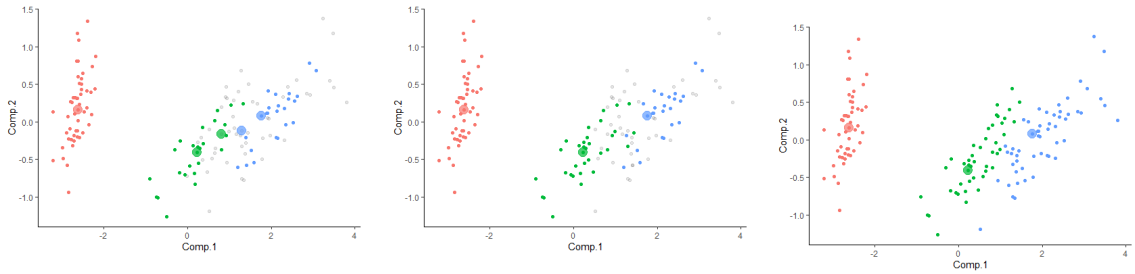
In terms of classification, if the default technique is used, the Mahalanobis depth function will continue to be used in both the maximum depth classifier and the algorithm for maximizing depth values in order to maintain consistency (if "maxdepth" is True). Unlike finding local centers, the covariance matrix now is based on the traditional moment estimation because in theory, temporary clusters are subsets of latent clusters, and all observations in the temporary clusters are selected with confidence after steps in Sections 2.3.1 and 2.3.2. It should not be necessary to further utilize MCD estimation for the covariance matrix. Besides the default algorithm, model-based classification method is also tested in applications.

3 Applications and Results

In this section, real data sets such as Iris, Seed and Wine from UCI machine learning repository [13] and synthetic data sets are used for investigating both the performance and

the logic behind DLCC.

3.1 Min strategy



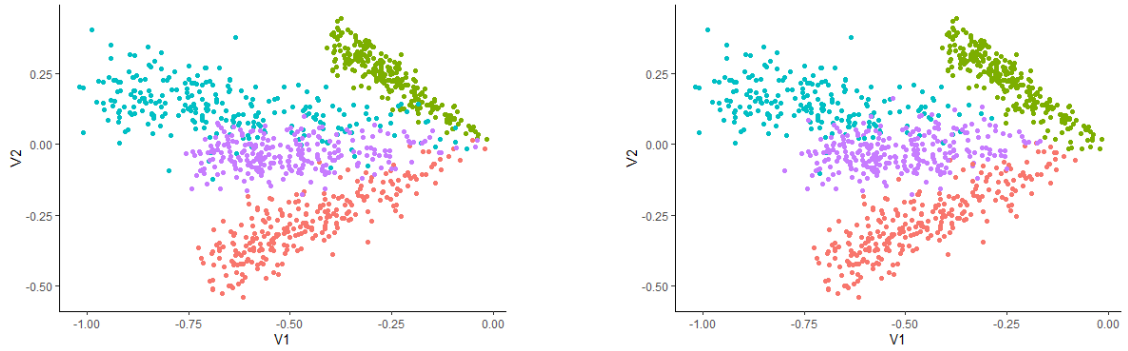
(a) Filtered centers and temporary clusters

(b) Redefined centroids and temporary clusters

(c) Final clusters

Figure 3: Visualizations for DLCC (min strategy) on the iris data set, where the x and y axes are the top two principle components and the color represents the clustering result. Grey points in temporary clusters are unlabelled observations.

Figure 3 shows there are 5 filtered centers on the Iris data set, and they are divided into 3 groups under min strategy. Notice that the number of filtered centers exceeds the number of clusters, the three deepest filtered centers from the current clustering result are selected as the re-defined centroids. In this step, two filtered centers locating at the border of two clusters are dropped in generating temporary clusters. Figure 3 also illustrates that, despite the presence of unlabeled observations in temporary clusters, the classification procedure yields a satisfactory clustering outcome.



(a) Real label

(b) Clustering result of DLCC (min strategy)

Figure 4: Graph of simulated data with ray shapes [14]. x and y axes are the first two variables of ray data.

Figure 4 shows the first two dimensional visualization of ray shapes data simulated by Passino [14]. The orientation of each cluster is obviously different, which means that a

global covariance matrix cannot provide a reasonable similarity matrix. In Figure 4, the clustering result given by DLCC closely resembles the real label, because the covariance matrices for constructing the similarity matrix here are provided by EEV GMM.

Table 1: Information regarding some chosen data sets, as well as settings of DLCC (min strategy) in them, where SM represents the approach for building similarity matrix, EU denotes the Euclidean distance.

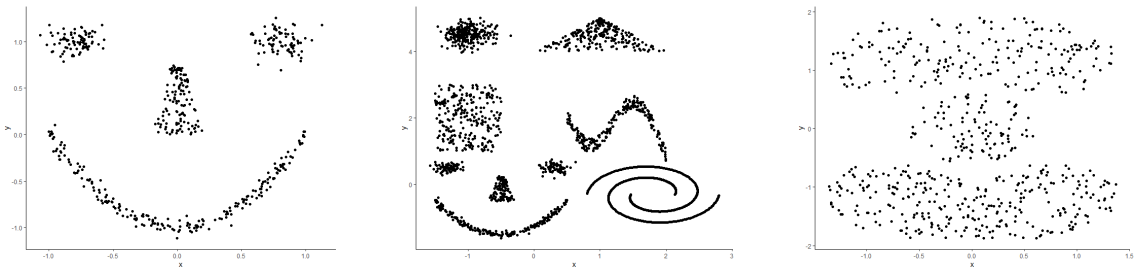
Data set	n	d	K	SM	δ	s	Maxdepth	Classification method
Iris	150	4	3	AEEV	0.70	50	TRUE	Default
Seed	210	7	3	AEEV	0.80	70	TRUE	Default
Wine	178	13	3	EU	0.70	50	TRUE	Default
Ray	1000	5	4	EEV	0.70	250	FALSE	Default

Table 1 provides some basic information of 4 data sets, as well as estimated number of clusters K and parameters in DLCC. The clustering performance of DLCC in these data sets is compared to that of GMM and PAM, using the external metric Adjusted Rand Index (ARI) [15] to evaluate the clustering quality. For convenience, the number of clusters in GMM and PAM are both set to K in Table 1. As shown in Table 2, DLCC generally gives better performances with suitable similarity matrix and parameters.

Table 2: Clustering performances comparisons in chosen data sets.

DATA SET	ARI		
	DLCC	GMM	PAM
Iris	0.904	0.904	0.642
Seed	0.787	0.737	0.747
Wine	0.982	0.930	0.741
Ray	0.823	0.780	0.484

3.2 Max strategy



(a) Smiley (b) Blend (c) Cassini

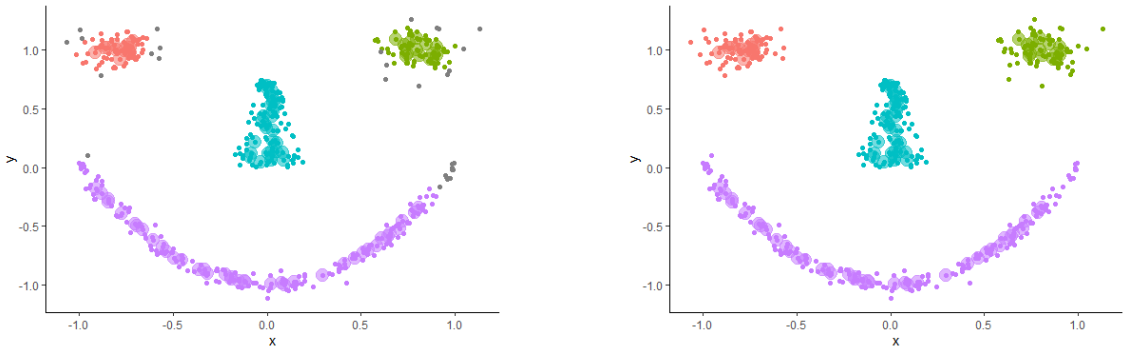
Figure 5: Graphs of three simulated data sets.

Figure 5 shows three synthetic data sets based on the mlbench package in R [16]. The third graph titled Cassini has three clusters, where two banana-shaped clusters gripping an oval-shaped cluster. From top to bottom, the number of points for three clusters are 200, 100 and 300 respectively, which results in varying densities among the three clusters.

Figure 6 reveals that the max strategy tends to define many filtered centers. With the chains of filtered centers, eyes, nose and mouth are correctly clustered.

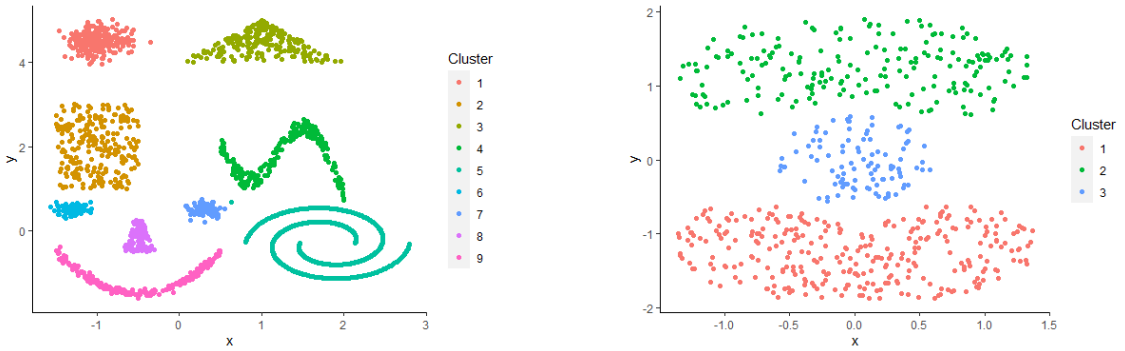
Table 3: Information regarding synthetic data sets, as well as the settings and performances of DLCC (max strategy) in them, where Mclass in Classification method means the mixture-model based classification.

Data set	n	d	K	δ	s	Maxdepth	Classification method	ARI
Smiley	500	2	4	0.43	25	FALSE	Default	1
Blend	2000	2	9	0.27	25	FALSE	Mclass	0.858
Cassini	600	2	3	0.40	25	TRUE	Default	1



(a) Filtered centers and temporary clusters (b) Final clusters

Figure 6: Visualizations about for DLCC (max strategy) on the simulated smiley data set, where the color represents the clustering result. Grey points in temporary clusters are unlabelled observations



(a) Clustering result in Blend (b) Clustering result in Cassini

Figure 7: Clustering results of DLCC (max strategy) in Blend and Cassini

Figures 6 and 7, and Table 3 show DLCC achieves perfect clustering in both smiley and Cassini, demonstrating its ability to tackle clustering problems with varying cluster densities. Although DLCC's performance in Blend is acceptable, it is unable to differentiate the spiral into two clusters. In fact all filtered centers in the spiral with $f = 1$, which indicates that the geometry of any subset in the spiral used to locate local centers is non-convex. Denote this type of cluster as a local non-convex cluster, and DLCC seems to be not performing well with local non-convex clusters. Besides, a miss clustered point between the eye and the wave may be corrected by using simple distance-based classification method like K-nearest neighbors instead of depth or model based classification methods.

From the examples above using both min and max strategies, the following rule-of-thumb on "maxdepth" seems reasonable. If the shapes of all latent clusters are convex, and no clusters overlap, then "maxdepth" should be set to TRUE. For clusters with small overlapping regions, if the sizes of the covariance matrices of them are similar, a true "maxdepth" is also reasonable.

In conclusion, DLCC algorithm can obtain impressive clustering results under different scenarios of clustering problems, and its framework has the potential to be extended to other depth functions.

4 Parameter selection (max strategy)

The selection of parameters under the max strategy seems to be ambiguous. In this section, a parameter selection algorithm under the max strategy, as well as a novel internal clustering criterion which can work for non-convex clustering problems are introduced. To avoid confusion, DLCC is defaulted to the max strategy in this section.

4.1 Hierarchical structure of DLCC

Two parameters in DLCC are size s and similarity threshold δ. The set of filtered centers, $\boldsymbol{C} = \{c_1, \ldots, c_P\}$, for a particular data set is fixed if s is fixed. Another parameter δ is for grouping filtered centers in $\boldsymbol{C}$. Recall Definition 2.2, let two filtered centers c_i and c_v be the most similar filtered centers between groups g_1 and g_2 with $\text{sim}(c_i, c_v) = \tilde{\delta} \leq \delta$, then if δ is updated as $0.99\tilde{\delta}$, the groups g_1 and g_2 will combine into one. Similarly, with smaller value of δ, more groups will be merged, which illustrates a hierarchical structure.

With an initial value of δ, an initial partition of filtered centers can be generated. The cut-off value for any two groups is the highest similarity value between filtered centers in the two groups. Although the number of groups reaches maximum when $\delta = 1$, it is not necessary to begin with $\delta = 1$ for selecting the most reasonable value of δ. Here, we suggest an upper bound for δ, which is defined as (11)

$$U(\delta) = \min\{\max\{\text{sim}(c_p, \boldsymbol{C} \setminus c_p), p = 1, \ldots, P\}\}. \tag{11}$$

For any $\delta \geq U(\delta)$, there will be at least one group consisting of a single filtered center. The max strategy assumes that each cluster is built by connecting multiple small sets, hence this circumstance is generally not expected and the initial value of δ is set to $0.99U(\delta)$. Afterward, a similarity matrix of initial groups can be obtained, where the (i, j)th entry of this matrix is the cut-off value between groups g_i and g_j. This similarity matrix of groups yields a list of δ and a corresponding list of the number of groups G. A toy example is presented in (12). Except 1 and 0, any two columns/rows with the largest element are merged together from one matrix to the next. From the toy example, it is easy to see the list of G is $\{6, 4, 3, 2\}$ and corresponding δ values are in $\{0.6 \leq \delta < U(\delta), 0.5 \leq \delta < 0.6, 0.4 \leq \delta < 0.5, 0 \leq \delta < 0.4\}$. For convenience, the list of δ is defined as $\boldsymbol{\delta} = \{0.6, 0.5, 0.4, 0\}$. Notably, with fixed s and filtered centers, the

potential values of G are also fixed, and the minimal number of groups can be greater than one if there are no neighbors that overlap across groups.

$$\begin{pmatrix} 1 & 0.6 & 0 & 0 & 0 & 0 \\ 0.6 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.5 & 0.3 & 0.2 \\ 0 & 0 & 0.5 & 1 & 0.4 & 0.1 \\ 0 & 0 & 0.3 & 0.4 & 1 & 0.6 \\ 0 & 0 & 0.2 & 0.1 & 0.6 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.5 & 0.3 \\ 0 & 0.5 & 1 & 0.4 \\ 0 & 0.3 & 0.4 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.4 \\ 0 & 0.4 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (12)$$

From the lists of G and δ, if the number of clusters is known, the suitable δ with respect to a chosen s can be found immediately.

4.2 Guessing process

If there is no information for the number of clusters, a guessing process for selecting the most suitable δ is provided as follows:

Step 1 If $\boldsymbol{\delta} \setminus \{0\} = \emptyset$, $\hat{\delta} = 0.99U(\delta)$; If $\boldsymbol{\delta} \setminus \{0\}$ only contains one element δ_1, $\hat{\delta} = \delta_1$; If the minimum number in the list of G is larger than 1, $\hat{\delta} = \min\{\boldsymbol{\delta} \setminus \{0\}\}$. Else, run left steps.

Step 2 Define the top $\min(2d, P/2)$ similar filtered centers with respect to any $c_p \in \boldsymbol{C}$ as its neighbouring centers. Compute the average similarity between each c_p and its neighbouring centers. Note the set of the average similarity as $\mathbf{m} = \{m_1, \ldots, , m_P\}$.

Step 3 Select filtered centers whose average similarity is smaller than the q quantile of $\mathbf{m}$, where q is a parameter with default value 10%. Denote those filtered centers as edge centers $\boldsymbol{C}_{\text{edge}}$, and remove any edge center c_i from $\boldsymbol{C}_{\text{edge}}$ if $\max\{\text{sim}(c_i, \boldsymbol{C}_{\text{edge}} \setminus \{c_i\})\} = 0$.

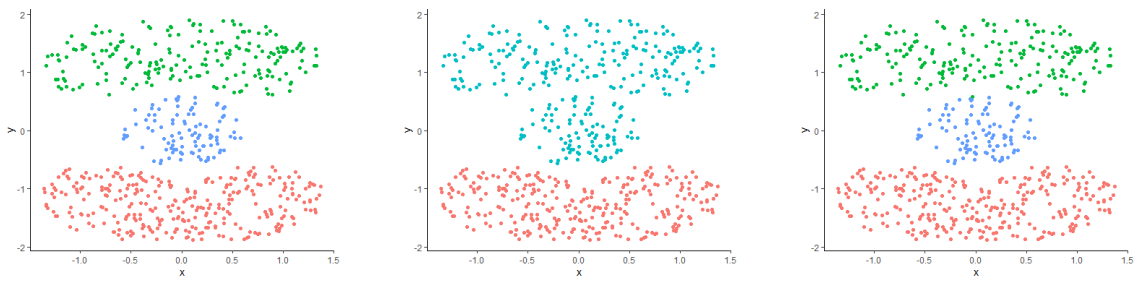
Step 4 For any c_i in $\boldsymbol{C}_{\text{edge}}$, find $l_i = \max\{\text{sim}(c_i, \boldsymbol{C} \setminus \{c_i\})\}$ and $\tilde{l}_i = \max\{\text{sim}(c_i, \boldsymbol{C}_{\text{edge}} \setminus \{c_i\})\}$. Calculate $\text{gap}_i = l_i - \tilde{l}_i$.

Step 5 Let $\mathcal{G}$ be the set of gaps. For $\text{gap}_j = \max\{\mathcal{G}\}$, if $\tilde{l}_j \in \boldsymbol{\delta}$, $\hat{\delta} = \tilde{l}_j$. Else, remove gap_j from $\mathcal{G}$ and repeat Step 5.

This guessing process assumes that any two groups are divided between filtered centers that are less similar to their neighbouring centers. The similarity between two adjacent edge centers can be considered between-group similarity if they are divided into two groups. Furthermore, for any δ value in $\boldsymbol{\delta}$, a filtered center must belong to the same group with the closest filtered center of it. Hence, the similarity l_i is the with-in group similarity. By finding the largest gap between with-in group and between-group similarities, the process guesses the value of δ; However if the guessing value is not in $\boldsymbol{\delta}$, which indicates that the critical cut does not occur between that edge center and its adjacent edge center, and the guess is incorrect, the process will continue to guess the next possible value of δ.

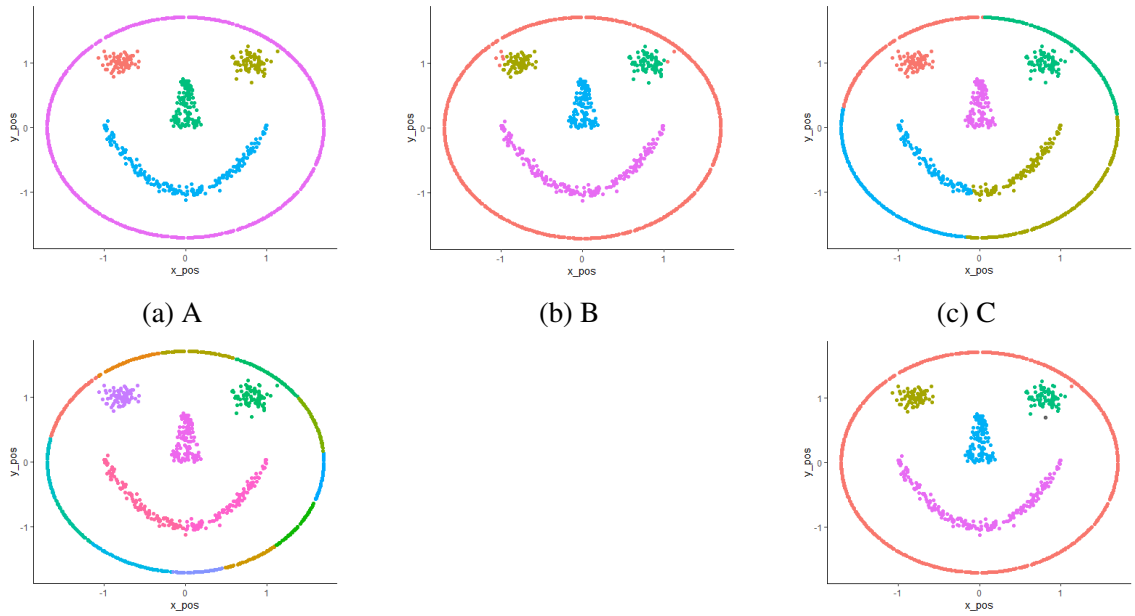
4.3 Density-based Clustering score

Given a list of size, algorithms discussed in Sections 4.1 and 4.2 can provide the corresponding δ for each s. An internal metric for evaluating clustering performance is required for determining the optimal pair of parameters, because the ground truth is typically unavailable in real-world clustering applications. Nevertheless, current well-known internal metrics such as silhouette width [17] and Calinski-Harabasz score [18] cannot perform well in both non-convex clustering problems and clustering problems with extremely closed clusters. To illustrate their limitations, two synthetic data sets with different clustering results are shown in Figures 8 and 9, where the first data set is the Cassini data and the second data set called SmileyF adds a "face" for the Smiley data. The average silhouette widths and the Calinski-Harabasz scores for those clustering results in the two data sets are presented in Table 4.



(a) A (b) B (c) C

Figure 8: A-C are 3 clustering results in the synthetic data set Cassini.



(a) A (b) B (c) C

(d) D (e) E

Figure 9: A-E are 5 clustering results in the synthetic data set SmileyF. One black point in E represents an unlabeled outlier.

Table 4: The average silhouette widths (asw) and the Calinski-Harabasz scores (CH) for clustering results in Figures 8 and 9.

DATA SET \ Clustering Result	A	B	C	D	E
Cassini: ASW	0.341	0.491	0.339		
Cassini: CH	642.13	842.52	641.99		
SmileyF: ASW	-0.146	-0.152	0.435	0.514	-0.147
SmileyF: CH	45.25	43.18	1470.62	2687.72	44.74

The clustering results in A are the actual labels for both data sets, whereas both asw and CH yield low scores for the clustering results in A. Both criteria favour the clustering result B for the Cassini data set, which contains only two clusters. Regarding the SmileyF data set, clustering results C and D award high scores according to two criteria; yet, based on human assessments, C and D should be the two poorest results out of 5. Therefore, those two internal metrics are not suitable for choosing the best pair of parameters for DLCC, and an innovative metric for evaluating clustering performances is proposed, which is called Density-based Clustering (DC) score.

It is a common sense to think about within cluster and between cluster properties for a clustering criterion; while for non-convex clustering problems, within cluster/between cluster distances/variances that calculated by one observation with respect to all observations in

the same/different cluster are not appropriate. To evaluate the performance of non-convex clusters, we explore calculating within and between cluster distances for each observation in a small, fixed region, which motivates us to involve the concept of a density-based clustering algorithm. Here, DBCA and the global similarity matrix introduced in Section 2.1 are employed. From now on, in lieu of within cluster/between cluster distances, within cluster/between cluster similarities are utilized.

Before proceeding, some definitions of DBCA are necessary.

Definition 4.1. ***Core neighbors****: Give a similarity threshold η, a point $\boldsymbol{x_j}$ is a core neighbor of $\boldsymbol{x_i}$ if $D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \geq \eta$ [8].*

Definition 4.2. ***Depth-connection****: A point $\boldsymbol{x_v}$ is depth-connected to $\boldsymbol{x_i}$ if there is a chain of points $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_y$ where each two adjacent points are core neighbors for each other, and $\tilde{\boldsymbol{x}}_1 = \boldsymbol{x_v}$ and $\tilde{\boldsymbol{x}}_p = \boldsymbol{x_i}$ [8].*

Definition 4.3. ***Cluster in DBCA****: A chain $\tilde{\mathbf{C}} = \{\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_y\}$ is called a maximal depth-connected chain if there is no $\boldsymbol{x_j} \notin \tilde{C}$ satisfying $\sup D_M(\tilde{\mathbf{C}} \mid \boldsymbol{x_j}) \geq \eta$. Set $\tilde{n}$ as the minimum number of points required to form a cluster in DBCA. All observations in a maximal depth-connected chain satisfying $\#\{\tilde{\mathbf{C}}\} \geq \tilde{n}$ construct a cluster.*

Definition 4.4. ***Outliers:*** *All observations remain unlabelled in DBCA are regarded as outliers.*

The greatest possible value of the similarity threshold η in DBCA, which can make all observations in cluster k be depth-connected except for outliers, is used to set the radius of a fixed region to calculate within cluster similarity for a cluster k, note the radius as η_k. Similarly, for computing between cluster similarity, the greatest possible value of η that can achieve the depth-connection of all observations in the data set is selected, note as η_Y. Denote the set of outliers for each cluster as $\mathbf{o}_k$ and the number of points in each cluster as

n_k. Then the within cluster similarity of cluster k is defined in (13).

$$
\begin{aligned}
J_k &= \sum_{i=1}^{n_k} w_i \mu_i, \quad \text{where} \\
\mu_i &= \begin{cases} \frac{\sum\{D_M(\boldsymbol{x_j}|\boldsymbol{x_i}) | D_M(\boldsymbol{x_j}|\boldsymbol{x_i}) \geq \eta_k\}}{\#\{\boldsymbol{x_j} | D_M(\boldsymbol{x_j}|\boldsymbol{x_i}) \geq \eta_k\}}, \text{if } \boldsymbol{x_i} \notin \mathbf{o}_k \\ \max\{D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \mid \boldsymbol{x_j} \notin \mathbf{o}_k\}, \text{if } \boldsymbol{x_i} \in \mathbf{o}_k \end{cases} \\
w_i &= \begin{cases} \frac{\#\{\boldsymbol{x_j} | D_M(\boldsymbol{x_j}|\boldsymbol{x_i}) \geq \eta_k\}}{\sum_{\boldsymbol{x_v} \notin \mathbf{o}_k} \#\{\boldsymbol{x_j} | D_M(\boldsymbol{x_j}|\boldsymbol{x_v}) \geq \eta_k\} + \#\{\mathbf{o}_k\}}, \text{if } \boldsymbol{x_i} \notin \mathbf{o}_k \\ \frac{1}{\sum_{\boldsymbol{x_v} \notin \mathbf{o}_k} \#\{\boldsymbol{x_j} | D_M(\boldsymbol{x_j}|\boldsymbol{x_v}) \geq \eta_k\} + \#\{\mathbf{o}_k\}}, \text{if } \boldsymbol{x_i} \in \mathbf{o}_k \end{cases}
\end{aligned} \tag{13}
$$

where $\boldsymbol{x_i}, \boldsymbol{x_j}, \boldsymbol{x_v} \in \mathcal{X}_k$, and $\mathcal{X}_k$ consists of observations in cluster k. For the between cluster similarity for cluster k, if $\boldsymbol{x_i} \in \mathcal{X}_k$ has $\max\{D_M(\mathbf{X} \setminus \boldsymbol{\mathcal{X}_k} \mid \boldsymbol{x_i})\} < \eta_Y$, where $\mathbf{X}$ is the total data set, then $\boldsymbol{x_i}$ is said as an inner observation of cluster k. Denote $\mathbf{I}_k$ as the set of inner observations in cluster k, and the between cluster similarity for cluster k is defined in (14).

$$
\begin{aligned}
H_k &= \sum_{i=1}^{n_k} \tilde{w}_i \tilde{\mu}_i, \quad \text{where} \\
\tilde{\mu}_i &= \begin{cases} \frac{\sum\{D_M(\boldsymbol{x_h}|\boldsymbol{x_i}) | D_M(\boldsymbol{x_h}|\boldsymbol{x_i}) \geq \eta_Y\}}{\#\{\boldsymbol{x_h} | D_M(\boldsymbol{x_h}|\boldsymbol{x_i}) \geq \eta_Y\}}, \text{if } \boldsymbol{x_i} \notin \mathbf{I}_k \\ \eta_Y, \text{if } \boldsymbol{x_i} \in \mathbf{I}_k \end{cases} \\
\tilde{w}_i &= \begin{cases} \frac{\#\{\boldsymbol{x_h} | D_M(\boldsymbol{x_h}|\boldsymbol{x_i}) \geq \eta_Y\}}{\sum_{\boldsymbol{x_u} \notin \mathbf{I}_k} \#\{\boldsymbol{x_u} | D_M(\boldsymbol{x_h}|\boldsymbol{x_u}) \geq \eta_Y\} + \#\{\mathbf{I}_k\}}, \text{if } \boldsymbol{x_i} \notin \mathbf{I}_k \\ \frac{1}{\sum_{\boldsymbol{x_u} \notin \mathbf{I}_k} \#\{\boldsymbol{x_u} | D_M(\boldsymbol{x_h}|\boldsymbol{x_u}) \geq \eta_Y\} + \#\{\mathbf{I}_k\}}, \text{if } \boldsymbol{x_i} \in \mathbf{I}_k \end{cases}
\end{aligned} \tag{14}
$$

where $\boldsymbol{x_i} \in \mathcal{X}_k$ and $\boldsymbol{x_h}, \boldsymbol{x_u} \notin \mathcal{X}_k$. After that, the total DC score for a clustering result is expressed as (15).

$$
\text{DC} = \sum_{k=1}^{K} \frac{n_k J_k}{H_k}, \tag{15}
$$

where n_k guarantees that the clustering performance for the cluster with more observations is more crucial. Weight function w_i in J_k gives observations with more core neighbors more weight and gives little weight for outliers. While for H_k, weight function $\tilde{w}_i$ gives observations which are closed to other clusters more weights, but little weight for inner

observations. Some properties of DC score are listed below:

Property 1: For a cluster k, if $\eta_{k'} < \eta_k$, then $\mu_{i'} \leq \mu_i$ for any $\boldsymbol{x_i} \in \mathcal{X}_k$.

Property 2: For a cluster k, if $\eta_{k'} < \eta_k$, then $\tilde{\mu}_{i'} \leq \tilde{\mu}_i$ for any $\boldsymbol{x_i} \in \mathcal{X}_k$.

Property 3: For a cluster k, the score given by any outlier is less than η_k.

Property 4: For a cluster k, $H_k \geq \eta_Y$, for more inner observations in k, H_k will be more closed to η_Y.

The proofs of Properties 1 and 2 are shown in Appendix 6.

Table 5 presents the DC scores for above examples of clustering results of Cassini and SmileyF, which seems to be much more reasonable than the average silhouette widths and Calinski-Harabasz scores in Table 4.

Table 5: The DC scores for clustering results in Figures 8 and 9, where $\tilde{n} = 3$.

	A	B	C	D	E
Cassini	629.73	628.57	629.05		
SmileyF	1822.27	1814.40	1666.02	1752.26	1821.83

All derived parameter pairs are tested after the internal metric for measuring clustering performances has been established. If $\hat{\delta}$ is chosen in the guessing process and $\hat{\delta} \in \boldsymbol{\delta}$, both $\hat{\delta}$ and $0.99\hat{\delta}$ are tried to examine if the choice of the cut is reasonable or not. Lastly, the parameter pair that produces the clustering results with the greatest DC score is chosen.

5 Projection Depth with a fixed center

Zuo and Serfling proposed the projection depth in 2000, and it has some favourable properties, such as a relatively high breakdown point for the sample projection median [19]. The depth function of the projection depth for a point $\boldsymbol{z}$ with respect to a data set $\mathbf{X}$ is defined as (16).

$$PD(\boldsymbol{z}|\mathbf{X}) = (1 + \sup_{\|\boldsymbol{u}\|=1} \frac{|\boldsymbol{u}^T\boldsymbol{z} - \text{Med}(\boldsymbol{u}^T\mathbf{X})|}{\text{MAD}(\boldsymbol{u}^T\mathbf{X})})^{-1}, \tag{16}$$

where Med stands for the median and MAD is the median absolute deviation defined as $\text{MAD}(U)=\text{Med}(|U - \text{Med}(U)|)$ for a univariate random variable U, and $u^T\mathbf{X} = \{u^T\boldsymbol{x}_1, u^T\boldsymbol{x}_2, \ldots, u^T\boldsymbol{x}_n\}$. The second term in (16) represents the Stahel-Donoho outlyingness with respect to $\mathbf{X}$, and hence (16) can be written as $(1 + O)^{-1}$, where $O = \sup_{\|\boldsymbol{u}\|=1} \frac{|\boldsymbol{u}^T\boldsymbol{z}-\text{Med}(\boldsymbol{u}^T\mathbf{X})|}{\text{MAD}(\boldsymbol{u}^T\mathbf{X})}$.

As inspired by the idea of using the Mahalanobis depth definition to measure similarities among observations, a fixed center version of the projection depth is proposed. A projection depth function with a fixed center $\boldsymbol{x}_j$ is shown in (17).

$$\widetilde{PD}(\boldsymbol{z}|\mathbf{X}, \boldsymbol{x}_j) = (1 + \sup_{\|\boldsymbol{u}\|} \frac{|\boldsymbol{u}^T\boldsymbol{z} - (\boldsymbol{u}^T\boldsymbol{x}_j)|}{\text{Med}(|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{u}^T\boldsymbol{x}_j|)})^{-1}, \tag{17}$$

where $\text{Med}(\boldsymbol{u}^T\mathbf{X})$ in the original function is replaced by $\boldsymbol{u}^T\boldsymbol{x}_j$ and $\widetilde{PD}(\boldsymbol{x}_j|\mathbf{X}, \boldsymbol{x}_j) = 1$. The depth values of (17) can be explained as the similarity of other observations to a single point.

Below are a few explanations of how to compute the projection depth with a fixed center for bivariate data. The computing strategy is based on Liu and Zuo's exact algorithm for calculating projection depth, in which they assume the data are in general position [20, 21]. Firstly we consider

$$\tilde{O} = \sup_{\|\boldsymbol{u}\|} \frac{|\boldsymbol{u}^T\boldsymbol{z} - (\boldsymbol{u}^T\boldsymbol{x}_j)|}{\text{Med}(|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{u}^T\boldsymbol{x}_j|)}) = \sup |Q|. \tag{18}$$

Q can be examined directly without considering the absolute value symbol, because Q is odd with respect to $\boldsymbol{u}$. For any $\boldsymbol{u}$, there is a permutation in $\boldsymbol{u}^T\mathbf{X}$, note the permutation as

$\{b_1, b_2, \ldots, b_n\}$, then there is

$$\boldsymbol{u}^T\boldsymbol{x}_{b_1} \leq \boldsymbol{u}^T\boldsymbol{x}_{b_1}, \cdots, \leq \boldsymbol{u}^T\boldsymbol{x}_{b_n}.$$

Note the current position of $\boldsymbol{u}^T\boldsymbol{x}_j$ is b_m, then a set of $\boldsymbol{u}$ can be defined as $\mathbb{U} = \{\boldsymbol{u}|\mathbb{A}^T\boldsymbol{u} \leq \mathbf{0}\}$, where $\mathbb{A} = \{\boldsymbol{x}_{b_1} - \boldsymbol{x}_{b_m}, \cdots, \boldsymbol{x}_{b_{m-1}} - \boldsymbol{x}_{b_m}, \boldsymbol{x}_{b_m} - \boldsymbol{x}_{b_{m+1}}, \cdots, \boldsymbol{x}_{b_m} - \boldsymbol{x}_{b_n}\}$. Simply speaking, the position of $\boldsymbol{x}_j$ in the permutation keeps the same when $\boldsymbol{u} \in \mathbb{U}$. Furthermore, the whole space of $\boldsymbol{u}$ can be divided into finite sets $\mathbb{U}_1, \mathbb{U}_2, \ldots, \mathbb{U}_U$. Based on the definition, when $\boldsymbol{u}$ moves from $\mathbb{U}_1$ to another region $\mathbb{U}_2$, b_m will change. Now reverting to a single set $\mathbb{U}$, for any $\boldsymbol{u} \in \mathbb{U}$, there is

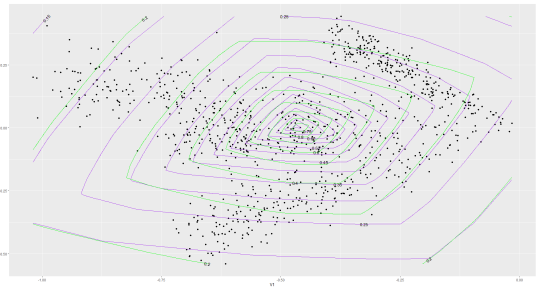
$$\begin{aligned} &\left|\boldsymbol{u}^T\boldsymbol{x}_l - \boldsymbol{u}^T\boldsymbol{x}_j\right| \\ &\qquad = \begin{cases} -(\boldsymbol{u}^T\boldsymbol{x}_l - \boldsymbol{u}^T\boldsymbol{x}_j), & \text{if } l \in \{b_1, b_2, \ldots, b_{m-1}\}, \\ \boldsymbol{u}^T\boldsymbol{x}_l - \boldsymbol{u}^T\boldsymbol{x}_j, & \text{if } l \in \{b_m, b_{m+1}, \ldots, b_n\}. \end{cases} \end{aligned}$$

Similarly, for any $\boldsymbol{u}$ there is a permutation for $|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{u}^T\boldsymbol{x}_j|$. Moreover, a region $\mathbb{D} \subset \mathbb{U}$ which ensures the median of $|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{u}^T\boldsymbol{x}_j|$ is independent with $\boldsymbol{u}$ can be defined. The median of $|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{u}^T\boldsymbol{x}_j|$ in $\mathbb{D}$ is noted as $\boldsymbol{u}^T V$. Then $Q = \frac{\boldsymbol{u}^T(\boldsymbol{z}-\boldsymbol{x}_j)}{\boldsymbol{u}^T V}$ over $\mathbb{D}$, which illustrates that Q is a piecewise linear fractional function over a finite number of pieces like $\mathbb{D}$, and for each piece the maximum value of Q only happens at the edge [21].

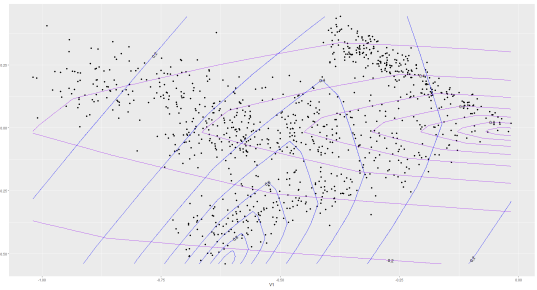
In the two dimensional case, unit vectors form a circle, and any unit vector can be represented by $\boldsymbol{u} = (\cos\alpha, \sin\alpha)$, where $\alpha \in [0, 2\pi]$. In fact, due to the symmetry, we only need to consider $\alpha \in [0, \pi]$. The circle is split into many pieces to find local maximum values of Q, and at the edge of each piece, $\boldsymbol{x}_{\text{med}}$, which satisfies $|\boldsymbol{u}^T\boldsymbol{x}_{\text{med}} - \boldsymbol{u}^T\boldsymbol{x}_j| = \text{Med}(|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{u}^T\boldsymbol{x}_j|)$, will change. We start with $\alpha = 0$ and passes through all pieces counter-clockwise. For each time the position of $\boldsymbol{x}_j$ changes in the permutation, b_m is either plus one or minus one depending on which observation switches position with it. Those unit vectors located at the edges can be used to calculate local maximum values, and as a result, we can find the global maximum value of Q to calculate the depth value. Lai and Zuo (2011) provided a very detailed explanation for calculating the projection depth in two dimensions [22].

Figure 10 reveals contour plots of (16) and (17) in the Ray data mentioned in Section 3, where the depths are calculated by the first two variables of the Ray data. From Figure 10a,

the contour plot of the original projection depth (PD) and the projection depth with the fixed center (PDfix) exhibit similar shapes, while they are slightly different in orientation. Moreover, PDfix's contour plot is also convex and polylateral. Figure 10b illustrates how the contour plots of PDfix can have completely various orientations depending on the center points chosen, indicating that the orientation of the contour will be automatically modified to account for the positions of other points.



(a) Original projection depth (Purple) VS Projection depth with the fixed center (The chosen center is the deepest point given by (16), Green)



(b) Projection depth with the fixed center (small y value, Blue) VS Projection depth with the fixed center (large x value, Purple)

Figure 10: Projection depth's contour plots in Ray data set with first two variables.

The majority of the properties of PD, like Affine invariance, Null at infinity, Monotone on rays, and so forth, are inherited by PDfix. The idea of fixed center version of data depth may have the potential to expand the applications of data depth.

6 Appendix

Lemma 6.1. *For a cluster k, if $\eta_{k'} < \eta_k$, then $\mu_{i'} \leq \mu_i$ for any $\boldsymbol{x_i} \in \mathcal{X}_k$. Further, if there is no outlier, the average weighted within cluster similarity for observations $\boldsymbol{x_i} \in \mathcal{X}_k$ will remain the same or become smaller for smaller η value, i.e., $\tilde{J}_{k'} \leq J_k$ (Note that $\tilde{J}_{k'} \neq J_{k'}$, because different η value means different clustering results).*

Proof. Let $\eta_{k'} < \eta_k$, for any observation $\boldsymbol{x_i} \in \mathcal{X}_k$ and $\notin \mathbf{o}_k$, there are

$$\mu_i = \frac{Q_i}{n_{Q_i}}, \tag{19}$$

$$\mu_{i'} = \frac{Q_i + W_i}{n_{Q_i} + n_{W_i}}, \tag{20}$$

where $Q_i = \sum\{D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \mid D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \geq \eta_k\}$, $n_{Q_i} = \#\{\boldsymbol{x_j} \mid D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \geq \eta_k\}$, $W_i = \sum\{D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \mid \eta_{k'} \leq D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) < \eta_k\}$ and $n_{W_i} = \#\{\boldsymbol{x_j} \mid D_M(\boldsymbol{x_j} \mid \eta_{k'} \leq D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) < \eta_k\}$. If $n_{W_i} = 0$, $\mu_i = \mu_{i'}$. For the case $n_{W_i} > 0$, $W_i, Q_i, n_{W_i}, n_{Q_i}$ are all positive, there are

$$\eta_{k'} \leq W_i/n_{W_i} < \eta_k \leq Q_i/n_{Q_i}, \tag{21}$$

$$W_i n_{Q_i} < Q_i n_{W_i}.$$

$$\mu_i - \mu_{i'} = \frac{Q_i n_{W_i} - W_i n_{Q_i}}{n_{Q_i}(n_{Q_i} + n_{W_i})} > 0. \tag{22}$$

Therefore, $\mu_i > \mu_{i'}$ when $\eta_{k'} < \eta_k$.

Without considering outliers, for the within cluster similarity, there are

$$J_k = \frac{\sum Q_i}{\sum n_{Q_i}}, \tag{23}$$

$$\tilde{J}_{k'} = \frac{\sum(Q_i + W_i)}{\sum(n_{Q_i} + n_{W_i})}. \tag{24}$$

To examine if $J_k > \tilde{J}_{k'}$, we can begin with a simple case with only two observations in a cluster k, which is shown in (25).

$$\frac{Q_1 + Q_2}{n_{Q_1} + n_{Q_2}} - \frac{Q_1 + W_1 + Q_2 + W_2}{n_{Q_1} + n_{W_1} + n_{Q_2} + n_{W_2}} = \frac{n_{W_1}Q_1 - n_{Q_1}W_1 + n_{W_2}Q_2 - n_{Q_2}W_2 + n_{W_2}Q_1 - n_{Q_1}W_2 + n_{W_1}Q_2 - n_{Q_2}W_1}{(n_{Q_1} + n_{Q_2})(n_{Q_1} + n_{W_1} + n_{Q_2} + n_{W_2})}, \tag{25}$$

where n_{W_1} and n_{W_2} are non-zero. Following (21), $n_{W_1}Q_1 - n_{Q_1}W_1 > 0$ and $n_{W_2}Q_2 - n_{Q_2}W_2 > 0$. Moreover, all $Q_i/n_{Q_i} \geq \eta_k$ and all $W_j/n_{W_j} < \eta_k$ with $n_{W_j} \neq 0$; Thereby, even when $i \neq j$, $n_{Q_i}W_j < n_{W_j}Q_i$ holds. Then there are $n_{W_2}Q_1 - n_{Q_1}W_2 > 0$ and $n_{W_1}Q_2 - n_{Q_2}W_1 > 0$. Hence,

$$\frac{Q_1 + Q_2}{n_{Q_1} + n_{Q_2}} - \frac{Q_1 + W_1 + Q_2 + W_2}{n_{Q_1} + n_{W_1} + n_{Q_2} + n_{W_2}} > 0. \tag{26}$$

Similarly, if $Q_1 + Q_2$, $n_{Q_1} + n_{Q_2}$, $W_1 + W_2$ and $n_{W_1} + n_{W_2}$ are written as $Q_{1\sim2}$, $n_{Q_{1\sim2}}$, $W_{1\sim2}$ and $n_{W_{1\sim2}}$, it is easy to show $J_k - \tilde{J}_{k'} > 0$ with more points. Therefore, $J_k > \tilde{J}_{k'}$. Besides, $J_k = \tilde{J}_{k'}$ if and only if $\sum n_{W_i} = 0$. □

Lemma 6.2. *For a cluster k, if $\eta_{k'} < \eta_k$ then $\tilde{\mu}_{i'} \leq \tilde{\mu}_i$ for any $\boldsymbol{x_i} \in \mathcal{X}_k$. Further, if core neighbors of all $\boldsymbol{x_i} \in \mathcal{X}_{k'}$ are in $\mathcal{X}_{k'}$, then the average between cluster similarity for observations $\boldsymbol{x_i} \in \mathcal{X}_k$ will remain the same or become smaller for smaller η value, i.e., $\tilde{H}_{k'} \leq H_k$ (Note that $\tilde{H}_{k'} \neq H_{k'}$, because different η value means different clustering results).*

Proof. Let $\eta_k > \eta_{k'} \geq \eta_Y$ and assume for every $\boldsymbol{x_j}$ satisfying $D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \geq \eta_{k'}$, there is $\boldsymbol{x_j} \in \mathcal{X}_{k'}$. Then for any $\boldsymbol{x_i} \in \mathcal{X}_k$, there is,

$$\begin{aligned}\{\boldsymbol{x_j} \mid D_M(\boldsymbol{x_j} \mid \boldsymbol{x_i}) \geq \eta_{k'}\} \;=\; & \{\boldsymbol{x_v} \mid D_M(\boldsymbol{x_v} \mid \boldsymbol{x_i}) \geq \eta_k\} + \\ & \{\boldsymbol{x_h} \mid \eta_{k'} \leq D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) < \eta_k\},\end{aligned}$$

where $\boldsymbol{x_j} \in \mathcal{X}_{k'}$, $\boldsymbol{x_v} \in \mathcal{X}_k$ and $\boldsymbol{x_h} \notin \mathcal{X}_k$, which means with a higher η, there may exist some observations being assigned to other clusters.

$$\tilde{\mu}_{i'} = \frac{V_i}{n_{V_i}}, \tag{27}$$

$$\tilde{\mu}_i = \frac{V_i + L_i}{n_{V_i} + n_{L_i}}, \tag{28}$$

$$\tilde{H}_{k'} = \frac{\sum V_i + n_{I_1}\eta_Y}{\sum n_{V_i} + n_{I_1}}, \tag{29}$$

$$H_k = \frac{\sum V_i + \sum L_i + n_{I_2}\eta_Y}{\sum n_{V_i} + \sum n_{L_i} + n_{I_2}}, \tag{30}$$

where $V_i = \sum\{D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) \mid \eta_{k'} > D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) \geq \eta_Y\}$, $n_{v_i} = \#\{\boldsymbol{x_h} \mid \eta_{k'} > D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) \geq \eta_Y\}$, $n_{I_1} = \#\{\mathbf{I_{k'}}\}$, $L_i = \sum\{D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) \mid \eta_{k'} \leq D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) < \eta_k\}$,

$n_{L_i} = \#\{\boldsymbol{x_h} \mid \eta_{k'} \leq D_M(\boldsymbol{x_h} \mid \boldsymbol{x_i}) < \eta_k\}$ and $n_{I_2} = \#\{\mathbf{I_{k'}}\}$. Similar with the proof of Property 1, it is easy to show $\tilde{\mu}_{i'} - \tilde{\mu}_i < 0$ when $n_{L_i} > 0$ and $\tilde{\mu}_{i'} = \tilde{\mu}_i$ when $n_{L_i} > 0$. For the case $\sum n_{L_i} > 0$, consider

$$\frac{\sum V_i + n_{I_1}\eta_Y}{\sum n_{V_i} + n_{I_1}} - \frac{\sum V_i + \sum L_i + n_{I_2}\eta_Y}{\sum n_{V_i} + \sum n_{L_i} + n_{I_2}} = \frac{\sum V_i \sum n_{L_i} - \sum L_i \sum n_{V_i} + n_{I_1}(\sum n_{L_i}\eta_Y - \sum L_I) + (n_{I_1} - n_{I_2})(\sum n_{V_i}\eta_Y - \sum V_i)}{(\sum n_{V_i} + n_{I_1})(\sum n_{V_i} + \sum n_{L_i} + n_{I_2})}.$$

Because $\eta_Y \leq \sum V_i / \sum n_{V_i} < \eta_{k'} \leq \sum L_i / \sum n_{L_i}$, there are $\sum V_i \sum n_{L_i} - \sum L_i \sum n_{V_i} < 0$ and $n_{I_1}(\sum n_{L_i}\eta_Y - \sum L_I) < 0$. Since $\eta_{k'} < \eta_k$ (With larger η, more similar points can be assigned to other clusters, and hence less number of inner observations.), $n_{I_1} \geq n_{I_2}$, and we know $\sum \eta_Y \leq \sum L_I / n_{L_i}$, then there is $(n_{I_1} - n_{I_2})(\sum n_{V_i}\eta_Y - \sum V_i) \leq 0$. Therefore, $\tilde{H}_{k'} < H_k$ when $\sum n_{L_i} > 0$. If $\sum n_{L_i} = 0$, $H_k = \tilde{H}_{k'}$. □

References

[1] Karl Mosler. Depth statistics. In *Robustness and complex data structures*, pages 17–34. Springer, 2013.

[2] Ricardo Fraiman, Regina Y Liu, and Jean Meloche. Multivariate density estimation by probing depth. *Lecture Notes-Monograph Series*, pages 415–430, 1997.

[3] Regina Y Liu and Kesar Singh. Rank tests for multivariate scale difference based on data depth. *DIMACS series in discrete mathematics and theoretical computer science*, 72:17, 2006.

[4] David L Donoho and Miriam Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, pages 1803–1827, 1992.

[5] Rebecka Jörnsten. Clustering and classification based on the l1 data depth. *Journal of Multivariate Analysis*, 90(1):67–89, 2004.

[6] Karl Mosler and Richard Hoberg. Data analysis and classification with the zonoid depth. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 72:49, 2006.

[7] Oleksii Pokotylo, Pavlo Mozharovskyi, and Rainer Dyckerhoff. Depth and depth-based classification with r-package ddalpha. *arXiv preprint arXiv:1608.04109*, 2016.

[8] Myeong-Hun Jeong, Yaping Cai, Clair J Sullivan, and Shaowen Wang. Data depth based clustering analysis. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2016.

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[10] Mia Hubert and Michiel Debruyne. Minimum covariance determinant. *Wiley interdisciplinary reviews: Computational statistics*, 2(1):36–43, 2010.

[11] Richard G Brereton. The mahalanobis distance and its relationship to principal component scores. *Journal of Chemometrics*, 29(3):143–145, 2015.

[12] Paul D McNicholas. *Mixture model-based classification*. Chapman and Hall/CRC, 2016.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[14] Francesco Sanna Passino, Nicholas A Heard, and Patrick Rubin-Delanchy. Spectral clustering on spherical coordinates under the degree-corrected stochastic blockmodel. *Technometrics*, pages 1–12, 2022.

[15] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[16] Friedrich Leisch and Evgenia Dimitriadou. *mlbench: Machine Learning Benchmark Problems*, 2021. R package version 2.1-3.

[17] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[18] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[19] Yijun Zuo. Projection-based depth functions and associated medians. *The Annals of Statistics*, 31(5):1460–1490, 2003.

[20] Xiaohui Liu, Yijun Zuo, and Zhizhong Wang. Exactly computing bivariate projection depth contours and median. *Computational Statistics & Data Analysis*, 60:1–11, 2013.

[21] Xiaohui Liu and Yijun Zuo. Computing projection depth and its associated estimators. *Statistics and Computing*, 24(1):51–63, 2014.

[22] Yijun Zuo and Shaoyong Lai. Exact computation of bivariate projection depth and the stahel–donoho estimator. *Computational Statistics & Data Analysis*, 55(3):1173–1179, 2011.

[23] Yijun Zuo and Robert Serfling. General notions of statistical depth function. *Annals of statistics*, pages 461–482, 2000.