Columbia University

IEOR E8100 Advanced Topics In OR - Machine Learning

Fall 2016

Research Project

Non-Parametric Density-Based Clustering Heuristic

Vincent Faber
Baptiste Richard

Contents

1 Introduction

The Density-Based Clustering Validation (DBCV), as developed by Davoud Moulavi Et al [1], provides a good way of measuring the quality of a clustering, based on within-cluster density and between-cluster density connectedness. Through a newly defined kernel density function, such index efficiently assesses the quality of a given clustering.

Though the DBCV allows to accurately rate a given clustering solution, it does not provide an algorithm allowing to find such a clustering. In this short research project, we present a simple clustering heuristic, inspired from the DBCV methodology, that relies on the density sparseness inside a given cluster as well as the separation between distinct clusters, in the mutual reachability space.

Interestingly, this heuristic is basically non-parametric, as explained later, and does not require making any of the usual clustering assumption about the data (e.g. number of clusters). Empirically, when the data is not too complex, the clustering obtained at the very iteration of the algorithm at which the DBCV is maximized often matches the ground truth clustering used to generate the data. We therefore use this fact to design a stopping condition for our heuristic, which indicates that the optimal clustering has been reached. Such results call for further theoretical research regarding the convexity of the DBCV values as the algorithm iterates. In some cases however, such stopping rule causes the algorithm to terminate too early or too late, and to return a sub-optimal clustering, and further work is needed to design a better stopping order.

2 Definitions and Concepts

The backbone of the algorithm relies on some of the metrics and dissimilarity measures defined by Davoud Moulavi Et al in their paper [1]. Before explicitly going through the heuristic, we therefore first define some of these concepts.

Consider the data set $D = \{x_1, x_2, ..., x_n\}$ where $D \subseteq R^b$. Further assume a clustering over D, namely $C = \{C_1, C_2, ..., C_L\}$. Let $d(x_i, x_j)$ be the Euclidean distance between x_i and x_j.

2.1 Core Distances

Assume $x_i \in D$ belongs to cluster C_p. The core distance of x_i is defined as,

$$a_{core}(x_i) = \left(\frac{\sum_{x_j \in C_p} \left(\frac{1}{d(x_i, x_j)} \right)^b}{|C_p|} \right)^{-\frac{1}{b}}$$

2.2 Mutual Reachability Distance

The mutual reachability distance matrix between points in D can be computed as follow:
$M = \{m_{ij}\}$ for all pairs $i, j = 1, 2, 3, ..., n$ where,

$$m_{ij} = MR(x_i, x_j) = \max\Big(a_{core}(x_i), a_{core}(x_j), d(x_i, x_j)\Big)$$

2.3 MR Graph and Minimum Spanning Tree

From the MR matrix M, one can compute the complete MR graph, G.
In the MR space, each point x_i will be connected to all other points such that the length of the edge between x_i and x_j simply is $m_{ij} = MR(x_i, x_j)$. The minimum spanning tree of G, denoted $MST^{(0)}(G)$, can be computed via Boruvka's or Prim's algorithm. Recall that $MST^{(0)}(G)$ can simply be represented as a symmetric matrix, $A^{(0)}_{MST} = \{a^{(0)}_{ij}\}$ where:

- $a^{(0)}_{ij} = a^{(0)}_{ji} = 0$ if there does not exist an edge between x_i and x_j in $MST^{(0)}(G)$
- $a^{(0)}_{ij} = a^{(0)}_{ji} = MR(x_i, x_j)$ if there exists an edge between x_i and x_j in $MST^{(0)}(G)$.

2.4 Density-Based Clustering Validation

Finally, still following the paper by Davoud Moulavi Et al [1], we define the density-based clustering validation as follow:

- Given a clustering $C = \{C_1, C_2, ..., C_L\}$
- The density sparseness of a cluster C_i denoted $DSC(C_i)$, is the heaviest/longest internal edge of the minimum spanning tree of the complete MR graph of the points in C_i only. We denote it $MST(C_i)$. Note that the lower $DSC(C_i)$ is, the denser cluster C_i is.
- The density separation of a pair of clusters (C_i, C_j), denoted $DSPC(C_i, C_j)$ for $i \neq j$, is the smallest MR distance possible between internal nodes of $MST(C_i)$ and $MST(C_j)$
- The validity index of cluster C_i is defined as:

$$V(C_i) = \frac{\min_{1 \leq j \leq L}\{DSPC(C_i, C_j)\} - DSC(C_i)}{\max\{\min_{1 \leq j \leq L}\{DSPC(C_i, C_j)\}, DSC(C_i)\}}, i \neq j$$

- Finally, the density-based clustering validation of the clustering C denoted $DBCV(C)$, is computed as follow:

$$DBCV(C) = DBCV(\{C_1, ..., C_L\}) = \sum_{i=1}^{L} \frac{|C_i|}{n} V(C_i)$$

3 Non-Parametric Density-Based Clustering Heuristic

3.1 Description

The goal of this heuristic is to proceed to sequential removals of the heaviest internal edges of the minimum spanning trees of the complete MR graph of G, $MST^{(0)}(G)$. Such operation will create a more and more disconnected graph, whose fully connected sub-graphs will correspond to the newly updated clusters. By sequentially removing these edges, we create clusters that have low density sparseness, and high density separation with other clusters. We wish to repeat the above operation several times so as to gradually improve the quality of the resulting clustering that we monitor by computing the DBCV of the updated clustering at each step.

Empirical tests (no theoretical proof as of yet) indicate that the clustering obtained at the very iteration at which the DBCV value is maximized tends to corresponds to the ground truth clustering. The algorithm therefore uses this fact as stopping condition. We realize that this is not always the case, which calls for further research and refinement of the rule causing the algorithm to terminate.

We make the following notation remarks:

1. After removing an edge from the initial minimum spanning trees of the complete MR graph of G, $MST^{(0)}(G)$, the graph becomes disconnected. Even though minimum spanning trees are not disconnected by definition, we denote this disconnected graph $MST^{(1)}(G)$, in order to simplify notations.

2. More generally, $MST^{(m+1)}(G)$ is the graph obtained via the removal of the heaviest internal edge from $MST^{(m)}(G)$.

3. Finally, we point out that the number of clusters at the $m-$th step is $m+1$, which is also the number of fully connected sub-graphs of $MST^{(m)}(G)$.

3.2 Algorithm

Consider a data set $D = \{x_1, x_2, ..., x_n\}$ where $D \subseteq R^b$.

1. Set $m = 0$ and set the initial clustering to D, in other words $C^{(0)} = \{D\}$. Set $k << n$ e.g. $k = \lfloor n/100 \rfloor$

2. At m=0, there is a single cluster and we therefore compute the core distances using the k closest neighbors. Let $KNN(x, i)$ be the distance between point x and its i^{th} nearest neighbor. Hence, for all $i = 1, ..., n$ the core distance $a_{core}(x_i)$ is defined as:

$$a_{core}(x_i) = \left(\frac{\sum_{j=1}^{k} \left(\frac{1}{KNN(x_i, j)} \right)^b}{k} \right)^{-\frac{1}{b}}$$

3. Compute the mutual reachability distance matrix M over all points in D
(See definition in 2.2)

4. Compute the complete MR graph G, the associated minimum spanning tree $MST^{(0)}(G)$ and its associated matrix $A^{(0)}_{MST}$ (See definition in 2.3)

5. Iterate until stop condition is reached:

   (a) Find the pair of indices (i^*, j^*) where $i, j \in D$ and $i \neq j$, such that the edge between the corresponding x_i and x_j is the largest[1]:

$$(i^*, j^*) = \underset{(i,j)}{\text{argmax}} \quad a^{(m)}_{\{ij\}}$$

   (b) Define $A^{(m+1)}_{MST}$ by setting $a^{(m)}_{\{i^*j^*\}} = a^{(m)}_{\{j^*i^*\}} = 0$ in $A^{(m)}_{MST}$. In other words,

$$A^{(m+1)}_{MST} = \{a^{(m+1)}_{ij}\}$$

   where,

   - $a^{(m+1)}_{ij} = a^{(m)}_{ij}$ for all i, j such that $(i, j) \neq (i^*, j^*)$
   - $a^{(m)}_{i^*j^*} = a^{(m)}_{j^*i^*} = 0$

   (c) Retrieve the graph $MST^{(m+1)}(G)$ corresponding to $A^{(m+1)}_{MST}$ and extract the resulting clusters from it as follow:

   - Given $MST^{(m+1)}(G)$, we denote $\{g_1, g_2, ..., g_{m+2}\}$ as the $m + 2$ fully connected sub-graphs of $MST^{(m+1)}(G)$.
   - Define the clustering $C^{(m+1)} = \{C_1, C_2, ..., C_{m+2}\}$ where,

$$C_i = \{x_l | x_l \in g_i\}$$

[1] In the implementation, we only consider removing internal edges and edges such that the resulting sub-graph has at least 3 nodes

(d) Compute the DBCV of the resulting clustering $C^{(m+1)} = \{C_1, C_2, ..., C_{m+2}\}$, denoted $DBCV^{(m+1)}$ (See defintion in 2.4)

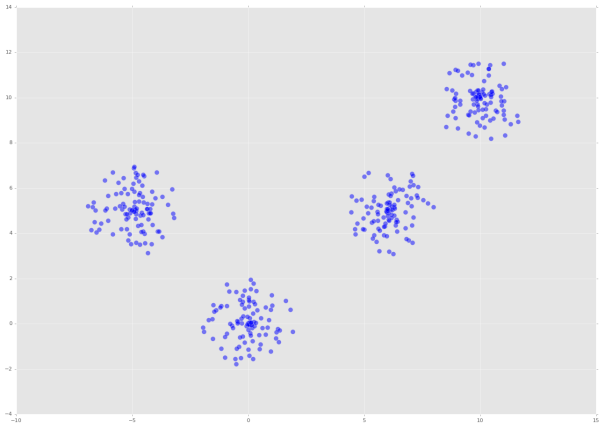
(e) Stop condition:

- If $DBCV^{(m+1)} \geq DBCV^{(m)} \rightarrow$ set $m \leftarrow m+1$ and repeat step 5.a through 5.e
- If $DBCV^{(m+1)} < DBCV^{(m)} \rightarrow STOP$ and backtrack. The final clustering $C^{(m)}$, was reached at the previous step

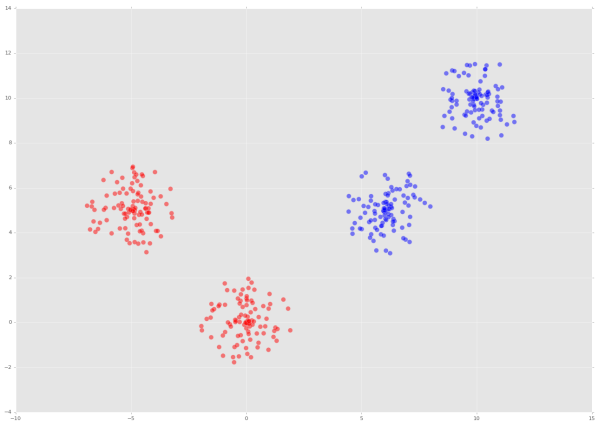
4 Application, Advantages and Future Steps

4.1 Examples in R^2

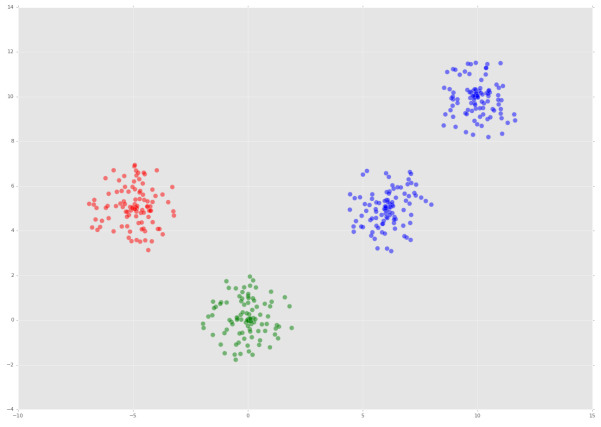
The simulation below visually describes the clustering obtained at each iteration of the algorithm on a data set with 4 balls in R^2:



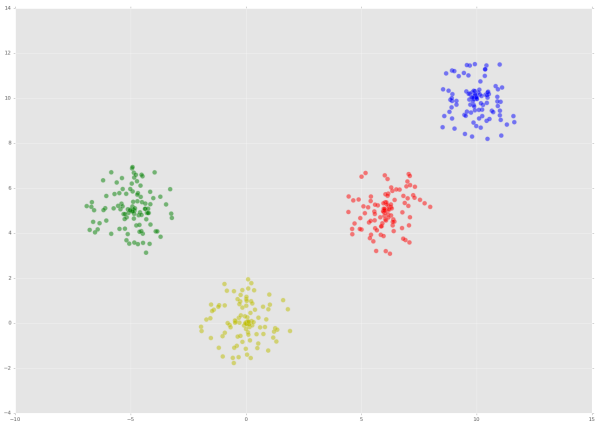
(a) Initial Dataset



(b) 1^{st} Iteration: DBCV = 0.523



(c) 2^{nd} Iteration: DBCV = 0.602



(d) 3^{rd} Iteration (Final): DBCV = 0.611

On the 4^{th} iteration below, the DBCV drastically drops from 0.611 to 0.431, triggering the stop order, and outputting the clustering resulting from the 3^{rd} iteration.

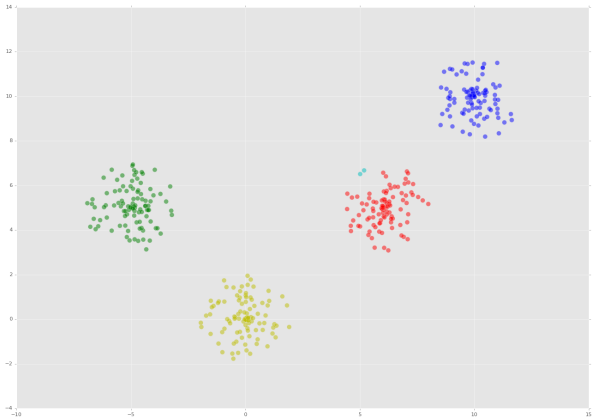
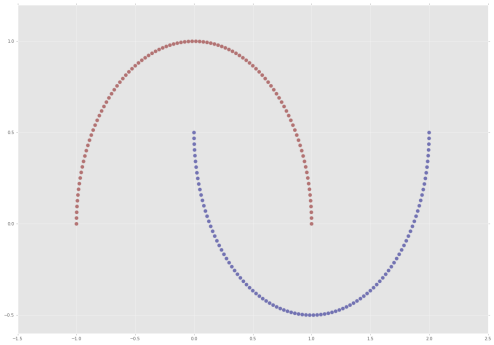


Figure 2: 4^{th} Iteration: $DBCV^{(4)} = 0.431 < DBCV^{(3)} \rightarrow$ Trigger Stop + Backtrack

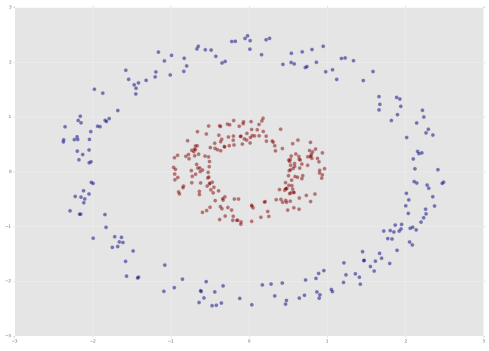
It is interesting to notice that the 4^{th} iteration created an extra cluster (cyan), small and very close to an already existing one (red). This step makes the density separation of this pair of cluster $DSPC(C_{cyan}, C_{red})$, extremely low which makes $V(C_{red})$ negative (-0.103) and decreases the overall DBCV value to 0.431. The algorithm therefore emits a stop order, indicating that the clustering from the previous iteration was optimal.

This type of data set can be easily clustered using methods such as the K-means++ or DBSCAN. However, unlike these algorithm, the above heuristic does not rely on assumptions regarding the data (e.g. number of clusters or $\epsilon-$neighborhood).

The heuristic also performs well with non-globular clusters, as showed below. Here, the algorithm converges in 1 step only, since further edge removals in the MR space minimum spanning tree only decreases the DBCV value:

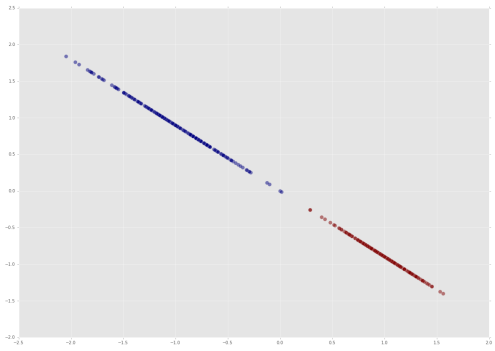


(a) 1^{st} (Final) Iteration: DBCV = 0.153

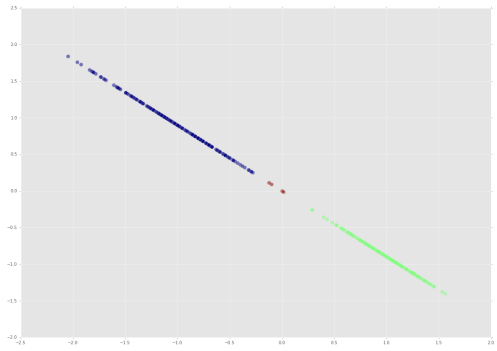


(b) 1^{st} (Final) Iteration: DBCV = 0.436

Similarly, on this line-like data set, the algorithm also converges in a single iteration too:



(a) 1^{st} (Final) Iteration: DBCV = 0.153



(b) 2^{nd} Iteration: DBCV = -0.116 $\rightarrow$ Stop + Backtrack

4.2 Advantages

As previously mentioned, one of the main advantages of such algorithm is that, unlike other clustering methods, it does not rely on meta-parameters. Indeed, the parameter k used in the initial computations of the core distances reflects the number of neighbors to take into account in the K-NN metrics. However empirically, it turns out that the results are pretty robust to a wide range of k values and the final output clustering remains unchanged, as long as $k << n$. In that sense, this heuristic can be considered as non-parametric. Another advantage of this method is that once the initial MR distance matrix has been computed, the main step is to simply search for the heaviest edge[2] in the current $MST^{(m)}(G)$. Such tasks boils down to searching for the largest value in the corresponding symmetric matrix $A^{(m)}_{MST}$, which can be done under $O(n^2)$. Note that $MST^{(0)}(G)$ is only computed once, and that the algorithm simply removes edges from it, extracts the resulting clusters, and computes the DBCV values of the updated clusterings.

The DBCV's calculation is longer, with order $O(kn^2 \log n)$, and requires computing $a_{core}(.)$'s, $MR(.,.)$'s, $MST(.)$'s, $DSC(.)$'s, $DSPC(.,.)$'s and $VC(.)$'s for each clusters and pairs of clusters.

[2]In the implementation, we only consider removing internal edges and edges such that the resulting sub-graph has at least 3 nodes

4.3 Limitations and Future Steps

The above empirical tests indicates that when the data is not too complex, the clustering obtained at the very iteration of the algorithm at which the DBCV is maximized tends to match the ground truth clustering used to generate the data. We therefore use this fact as a stopping condition. Such results call for further theoretical research regarding the convexity of the DBCV values as the algorithm iterates.

However, there are cases where the algorithm stops before or after the ground truth clustering is actually reached, which indicates that further improvements of the stopping rule is necessary. Interestingly, if we let the algorithm iterate without any stopping order, the ground truth clustering is often reached at some point. This means that the DBCV might be the right metrics but that our stopping condition is not yet optimal.

Finally we point out that from the $C^{(m)}$ clustering to the $C^{(m+1)}$ clustering, only a single edge has been removed from $MST^{(m)}(G)$, and all but 2 clusters in $C^{(m+1)}$ are exactly similar to clusters in $C^{(m)}$. In other words at each step, only 2 new density sparseness values and only 2 sets of values of density separation need to be recomputed: they are the only factors responsible in the evolution of the DBCV value from step m to step $m+1$.

References

[1] Davoud Moulavi, Pablo A. Jaskowiak, Ricardo J. G. B. Campello, Arthur Zimekz, Jorg Sander. "Density-Based Clustering Validation" *SIAM* - 2014