DBSCAN CLUSTERING
LECTURE: UNSUPERVISED LEARNING AND EVOLUTIONARY COMPUTATION USING R

Jakob Bossek

MALEO Group, Department of Computer Science, Paderborn University, Germany

1^{st} Dec, 2024

Learning Goals

- Visual inspection of clustering results of k-means on non-spherical data
- Another clustering algorithm: DBSCAN
- Intrinsic and extrinsic cluster evaluation

Recap

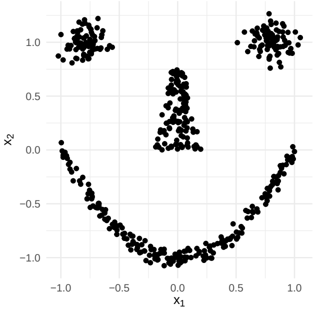
Hierarchical clustering

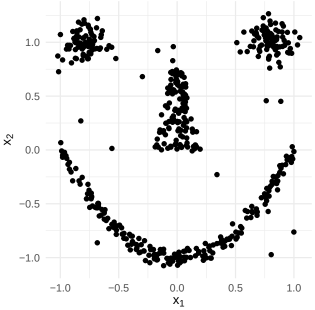
- Agglomerative approach: merge "closest" clusters until there is one cluster left
  $\rightsquigarrow$ Simple, yet appealing approach.
- Different linkage-functions to define distance between sets
- Cut dendrogram a-posteriori to obtain a clustering
- See Murtagh and Contreras 2012 for a survey

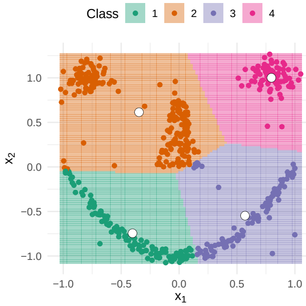
k-means clustering

- Partition-based a-priori approach (k is a parameter)
  $\rightsquigarrow$ Kind of captures our intuition of good clusters.
- Heuristic method requires multiple restarts
  $\rightsquigarrow$ Still, even after 1 000 restarts we cannot guarantee convergence to global optimum.
- Elbow method is a simple approach to determine the "best" k

Failure for k-means







Drawbacks ...

... of so-far introduced clustering approaches:[1]

- Partition-based k-means is
  - Designed for convex-shaped clusters
    A shape S is called *convex* if for every two $x, y \in S$, all points on the straight line between x and y are in S.
  - $\leadsto$ Cannot detect nested clusters
  - Sensitive to noise
- HC-algorithms suffer from:
  - sensitivity to noise and outliers
  - Breaks large clusters
  - The order of the data has an impact on the final results

[1] Note, that we do not aim to bash these algorithms! They just have different cluster models and are very much used in practice.

Adapted problem definition

In the following we allow for a modified definition of a k-partition with noise.
I.e. we allow for a $(k+1)^{\text{st}}$ set N:

Definition (Extended k-partition)

Given a data set $\mathcal{X}$ an *extended k-partition* is a decomposition of $\mathcal{X}$ into $k+1$ sub-sets, $C_1, \ldots, C_k$, C_{k+1} such that

1. $C_1, \ldots, C_k$ are non-empty,
2. $C_i \cap C_j = \emptyset$ for $1 \leq i \neq j \leq k+1$ and
3. $\left(\bigcup_{i=1}^{k} C_i\right) \cup C_{k+1} = \mathcal{X}$.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN

Density-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise

Core idea

Density-based approach:

- Points of a cluster are grouped close
  I.e., a point x belongs to a cluster if there are enough points close to x (*dense area*)
- Explicit handling of noise / outlier points
  Points in non-dense areas likely do not belong to any cluster.

DBSCAN: some facts

DBSCAN is the most cited clustering algorithm to date

- Article "*A density-based algorithm for discovering clusters in large spatial databases with noise*" published at ACM SIGKDD conference[2] 1996 (Ester et al. 1996)
- According to Google scholar[3] the citation count is 34 595
- Awarded ACM SIGKDD "test of time" award in 2014
- Follow-up article "*DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN*" by Schubert et al. (Schubert et al. 2017) in ACM Transactions on Database Systems (TODS) journal

---

2 One of the major data mining conferences.

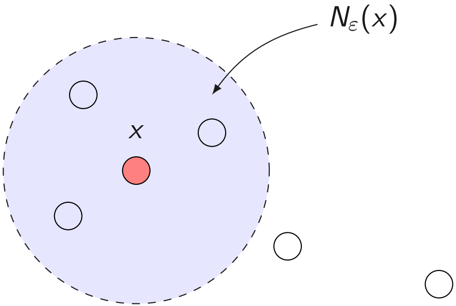
3 Queried 25 November, 2024 at 4pm; count count was at 22 881 3 years ago

DBSCAN

Definition (ε-neighborhood)

The ε-*neighborhood* of a point $x \in \mathcal{X}$ for some $\varepsilon > 0$ is defined as

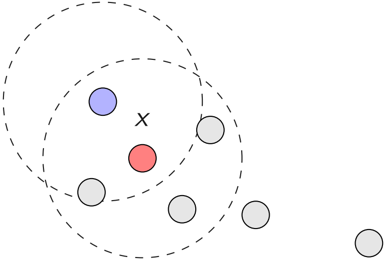
$$N_\varepsilon(x) := \{y \in \mathcal{X} \mid d(x, y) \leq \varepsilon\}.$$



DBSCAN

Definition (Core point)

Given a parameter $\texttt{minPts} > 0$ for *minimal number of points* and $\varepsilon > 0$ we define that a point x is a *core point* of a cluster if $|N_\varepsilon(x)| \geq \texttt{minPts}$.



Problem: *border points* on the edge of clusters usually have less neighbors than core points.

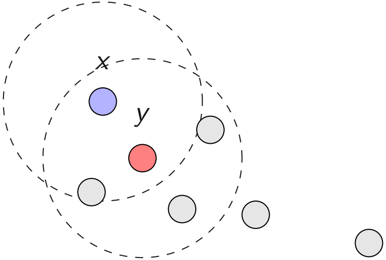
DBSCAN

Definition (directly density-reachable)

A point $x \in \mathcal{X}$ is *directly density-reachable* from $y \in \mathcal{X}$ with regard to ε and `minPts` if

$$(1) \quad x \in N_\varepsilon(y) \quad \text{and} \quad (2) \quad |N_\varepsilon(y)| \geq \texttt{minPts}.$$

Here, e.g., for $\texttt{minPts} = 3$, x is directly density-reachable from y:

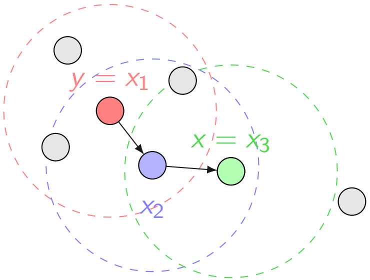


DBSCAN

Definition (density-reachable)

A point $x \in \mathcal{X}$ is *density-reachable* from $y \in \mathcal{X}$ with regard to `minPts` and $\varepsilon > 0$ if there is a chain/sequence of points $x_1, \ldots, x_l$ such that $x_1 = y$, $x_l = x$ such that x_{i+1} is directly density-reachable from x_i for $1 \leq i < l$.

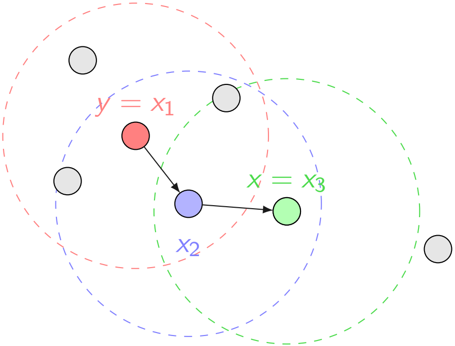
E.g., x is density-reachable from y via $y = x_1, x_2, x_3 = x$ for `minPts`$= 4$



DBSCAN

Problem: This density-reachable relation is not symmetric.[4]

Here, y is density-reachable from x, but x is not density reachable from y!



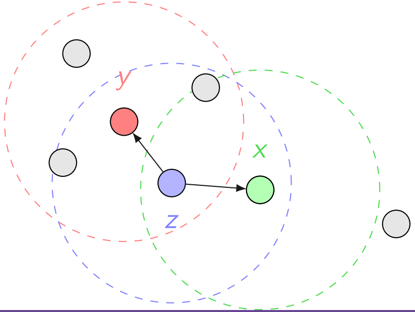
[4] If x is density-reachable from y, the vice verse is not necessarily true.

DBSCAN

Definition (density-connected)

A point $x \in \mathcal{X}$ is *density-connected* with a point $y \in \mathcal{X}$ with regard to $\texttt{minPts}$ and $\varepsilon > 0$ if there is a point $z \in \mathcal{X}$ such that x and y are both density-reachable from z.

Here, both x and y are density-reachable from z (for $\texttt{minPts} = 4$). Hence, x and y are density-connected.



DBSCAN

Based on these definitions a DBSCAN cluster is defined as follows:

Definition (DBSCAN cluster)

A cluster C is a subset of the the data set $\mathcal{X}$ such that the following two conditions hold:

1. $\forall x, y$: if $x \in C$ and y is density-reachable from x w.r.t. `minEps` and ε, then $y \in C$. (**maximality**)
2. $\forall x, y \in C$: x is density-connected to y w.r.t. `minEps` and ε. (**connectivity**)

DBSCAN

Tie breaker rule

If two clusters C_1 and C_2 are close together, there might exist a point x, which is a border point for both clusters. It cannot be a core point as in such a case the two clusters would have been merged! DBSCAN assigns x to the cluster that has been 'discovered' first.

DBSCAN

The actual algorithm

1. Identify the set $S \subset \mathcal{X}$ of core points.
2. Pick a core point $x \in S$ uniformly at random.
3. Calculate the set of points $R \subset \mathcal{X}$ which are density-reachable from x (w. r. t. ε and $\texttt{minPts}$)
   $\rightsquigarrow$ DBSCAN found a cluster! Remove x and R from $\mathcal{X}$.
4. Repeat steps (2) and (3) until $S = \emptyset$.
5. Return clusters and the set of outliers which contain all points not assigned to any cluster.

DBSCAN

Algorithm DBSCAN - Detailed Psuedo-Code

Require: Database DB, distance function distFun, ε, `minPts`

$C \leftarrow 0$ ▷ cluster counter
for point $x \in$ DB **do**
    continue if label(x) is not *undefined*
    NS $\leftarrow$ RangeQuery(()DB, distFun, x, ε)
    if $|NS| <$ `minPts` **then**
        label(x) $=$ noise
        continue
    $C \leftarrow C + 1$; label(x) $\leftarrow C$ ▷ Label initial point
    $S \leftarrow N \setminus \{x\}$ ▷ Relevant neighbors
    for point $x' \in S$ **do**
        if label(x') is "noise" **then**
            label(x') $\leftarrow C$
        continue if label(x) is not *undefined*
        label(x') $\leftarrow C$
        NS $\leftarrow$ RangeQuery(()DB, distFun, x', ε)
        if $NS \geq$ `minPts` **then** ▷ Density check: is x' a core point?
            $S \leftarrow S \cup N$

DBSCAN

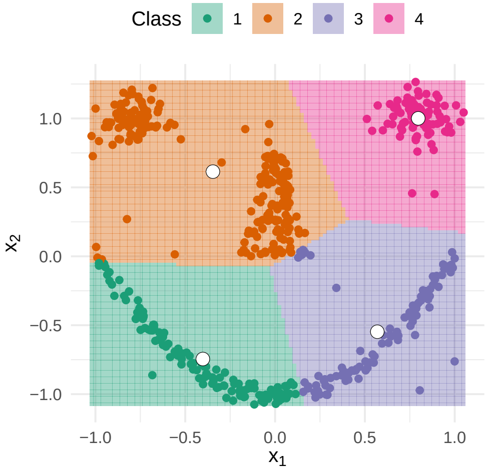
Algorithm REGIONQUERY (linear scan)[5]

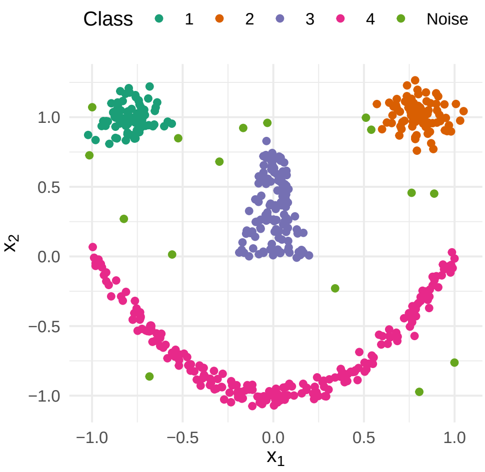
Require: Database DB, distance function distFun, point x, ε

```
Neighbors N ← ∅
for point x' ∈ DB do
    if distfun(x, x') ≤ ε then
        N ← N ∪ {x'}
return N
```

[5] Speed up using a database index.

DBSCAN: example

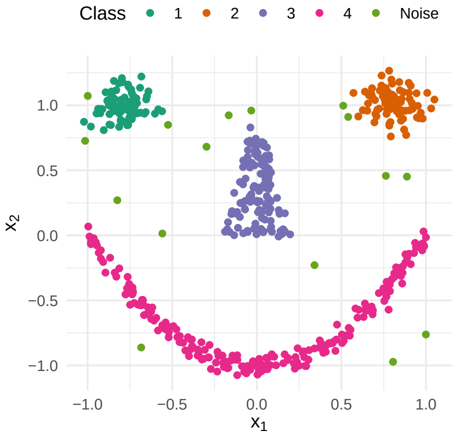


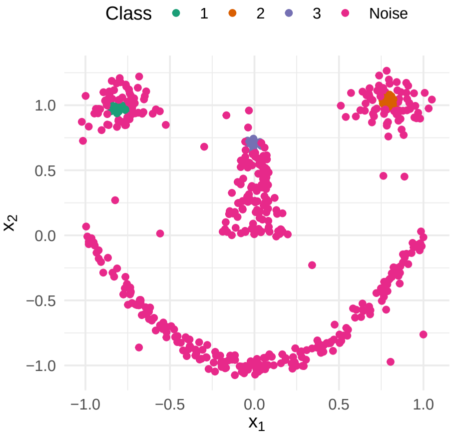


DBSCAN in animated action

How to choose the parameters?

Obvious: result strongly depends on the choice of ε and `minPts`![6]





How to choose the parameters?

Ester et al. (Ester et al. 1996) present a simple, yet effective and appealing heuristic to set both ε and `minPts`.

Observation

Let d be the distance of a point x to its k-th nearest neigbor. Then

- $|N_d(x)|$ contains most likely exactly $k + 1$ points
- Reducing k will usually have no drastic effect on d

How to choose the parameters?

For given k let

$$k\text{-dist} : \mathcal{X} \to \mathbb{R}^+$$

be the function that maps a point $x \in \mathcal{X}$ to its distance from its k-th nearest neighbor.

- Sort points in $\mathcal{X}$ in descending order of k-dist values
  $\rightsquigarrow$ *sorted k-dist graph*
- For an arbitrary point $x \in \mathcal{X}$, if we set $\varepsilon = k\text{-dist}(x)$ and $\texttt{minPts} = k$
  $\rightsquigarrow$ all points with equal or smaller k-dist values will be core points!

How to choose the parameters?

Idea

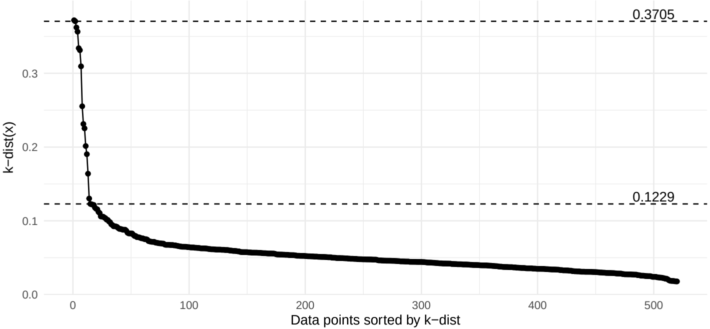
- Find *threshold point* x as the maximal k-dist value in "thinnest" cluster = first point in first "valley" ("elbow" point) in the sorted k-dist graph
  - Points left of x in sorted k-dist graph will be noise points (low density)
  - All points right of x (lower k-dist values) will be assigned to some cluster
- Determine threshold point x and set[7]

$$\varepsilon = k\text{-dist}(x) \quad \text{and} \quad \texttt{minPts} = k$$

[7] Experiments show that $k = 4$ is sufficient.

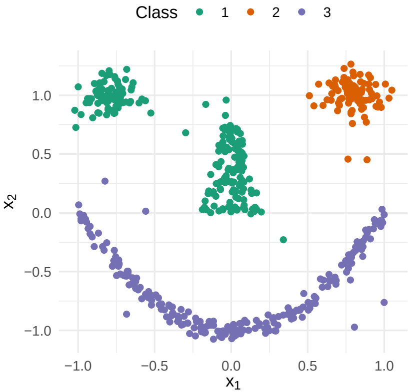
How to choose the parameters?

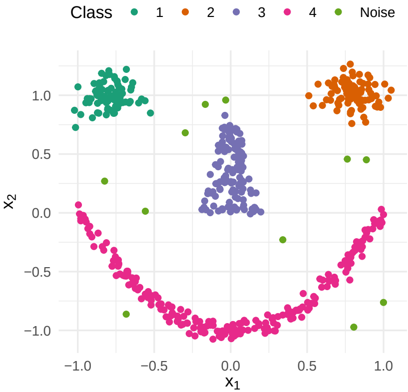
Sorted k-dist graph ($k = 4$) on smiley



"Elbow" at $\varepsilon = 0.1229$ seems to be a good choice according to the heuristic.

How to choose the parameters?





DBSCAN with `minPts` $= 4$ and bad (a) $\varepsilon = 0.3705$ (left) and (b) good $\varepsilon = 0.1229$ (right).

DBSCAN
Algorithm Complexity

- DBSCAN visits each point of the database, possibly multiple times (e.g., as candidates to different clusters)
- In practise though, the runtime complexity is mostly governed by the number of `regionQuery` invocations
  - DBSCAN executes exactly one such query for each point!
  - $\rightsquigarrow$ adopting an indexing structure that executes a neighborhood query in $\mathcal{O}(\log N)$ an overall average runtime complexity of $\mathcal{O}(N \log N)$ is obtained[8]
- The $\Theta(N^2)$ distance matrix can be kept in memory, whereas a non-matrix based implementation needs only $\mathcal{O}(N)$ memory

[8] If ε is chosen in a meaningful way, i.e. such that on average only $\mathcal{O}(\log N)$ points are returned

DBSCAN Clustering
Properties

Advantages ☺

- No need to specify no. of clusters a-priori
- Can find arbitrarily-shaped clusters
- Robust to outliers
- DBSCAN is designed for use with databases that can accelerate region queries, e.g. using an R^* tree
- Parameters `minPts` and ε can be set by a domain expert, if the data is well understood

Disavantages ☹

- Not entirely deterministic (e.g., border points assigned due to tie-breaker rule)
- If data is understood badly, "right" choice for the distance function is hard
- Based on Euclidean distance in most cases (curse of dimensionality: all points are far away for large p)
- Problems if clusters have different "densities (would require to select different combinations of ε and `minPts` per cluster)

Measuring Cluster Quality

External vs. internal

External (extrinsic)

Ground truth (ideal clustering) is available:

- Either if we are in benchmarking and use labelled data
- Built upon human expertise
- Often called *supervised method*

Internal (intrinsic)

No ground truth is available:

- Access goodness of a clustering by considering how well the clusters are separated
  I.e., the quality is evaluated on the clustered data itself!

External evaluation

Rand index (Rand 1971)

Let $C = \{C_1, \ldots, C_k\}$ be a clustering and $G = \{G_1, \ldots, G_l\}$ a ground-truth partition. Let

- TP (true positive) be the number of pairs of elements in $\mathcal{X}$ which are in the same subset in C and G
- TN (true negative) be the number of pairs of elements in $\mathcal{X}$ which are in different subsets in C and G
- FN (false negative) be the number of pairs of elements in $\mathcal{X}$ which are in the same subset in C but in different subsets in G
- FP (false positive) be the number of pairs of elements in $\mathcal{X}$ which are in different subsets in C but in the same subset in G

Calculate similarity to ground truth

$$RI = \frac{TP + TN}{TP + FP + FN + TN} = \frac{TP + TN}{\binom{N}{2}} \in [0, 1]$$

- Measure of the percentage of correct cluster assignments
  Takes value 1 if all pairs of points are either true positive or negative.

Internal evaluation

Dunn index (Dunn 1974)

For a clustering $C_1, \ldots, C_k$ the *Dunn index* is defined as

$$D(C_1, \ldots, C_k) := \frac{\min_{1 \le i < j \le k} d(C_i, C_j)}{\max_{1 \le l \le k} d'(C_l)}$$

where $d(C_i, C_j)$ is the *distance between the i-th and j-th cluster* and $d'(C_l)$ is the *intra-cluster distance* of cluster C_l.

- Both d and d' can be measured differently!
- High values preferable.

Internal evaluation

Davies-Bouldin index (Davies and Bouldin 1979)

For a clustering $C_1, \ldots, C_k$ the *Davies-Bouldin index* is defined as

$$D(C_1, \ldots, C_k) := \frac{1}{k} \sum_{i=1}^{k} \max_{1 \leq i \neq j \leq k} \left(\frac{\sigma_i + \sigma_j}{d(\mu_i, \mu_j)} \right)$$

where μ_i is the centroid / center of mass of the i-th cluster and $\sigma_i = \frac{1}{|C_i|} \sum_{x \in C_i} d(x, \mu_i)$ denotes the average distance of the points in the respective cluster to its centroid.

- Low values are preferred!
  Since this is in favor of high intra-cluster similarity and high inter-cluster dissimilarity

Internal evaluation

Silhouette (Rousseeuw 1987)

Let $C_1, \ldots, C_k$ be a clustering. For $x \in C_l$ let

$$a(x) = \frac{1}{|C_l| - 1} \sum_{\substack{y \in C_l \\ y \neq x}} d(x, y)$$

be the *mean distance between* x *and all other points in* x*'s cluster*. Let further for $x \in C_l$

$$b(x) = \min_{\substack{1 \leq i \leq k \\ i \neq l}} \frac{1}{|C_i|} \sum_{y \in C_i} d(x, y)$$

be the *smallest mean distance of* x *to all points in any other cluster*, i.e., the neighboring cluster. Then the *silhouette (value/width)* of $x \in C_l$ is defined as

$$s(x) = \begin{cases} \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}, & \text{if } |C_l| > 1 \\ 0, & \text{otherwise} \end{cases} \qquad = \begin{cases} 1 - \frac{a(x)}{b(x)}, & \text{if } a(x) < b(x) \\ 0 & \text{if } a(x) = b(x) \in [-1, 1] \\ \frac{b(x)}{a(x)} - 1, & \text{if } a(x) > b(x) \end{cases}$$

Exercises (at last ☺)

1. Calculate (using Manhattan distance for ease of calculation) the silhouette values for points B and F by hand. Try to come up with an interpretation of the values!

2. Show that $s(x) \in [-1, 1]$ always holds.

Sample solutions

Sample solutions

Sample solutions

Internal evaluation

Silhouette

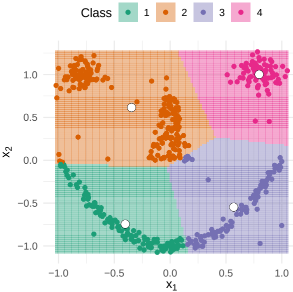
For $x \in C_l$ the *silhouette* is defined as

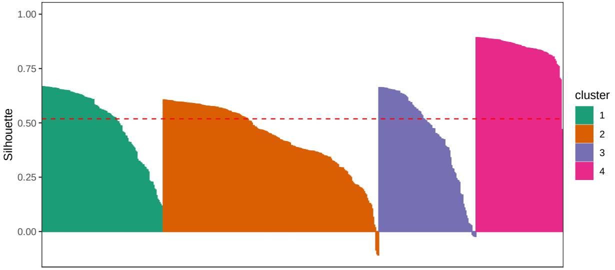
$$s(x) = \begin{cases} 1 - \frac{a(x)}{b(x)}, & \text{if } a(x) < b(x) \\ 0 & \text{if } a(x) = b(x) \in [-1, 1] \\ \frac{b(x)}{a(x)} - 1, & \text{if } a(x) > b(x) \end{cases}$$

- Values close to 1 $\leadsto$ x is well-clustered
  Requires low $a(x)$ (x is very similar to the point in its cluster) and high $b(x)$ (x very dissimlar to other clusters)
- Values close to -1 $\leadsto$ x is <u>not</u> well-clustered
- Idea: plot all silhouette values
  Many negative or low positive values $\leadsto$ to few or to many clusters.

Internal evaluation

Silhouette of k-means ($k = 4$) on smiley





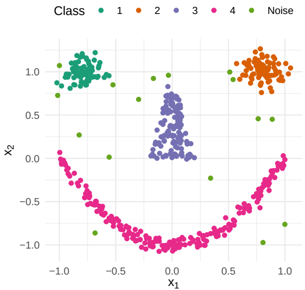
- Almost all silhouette values positive!

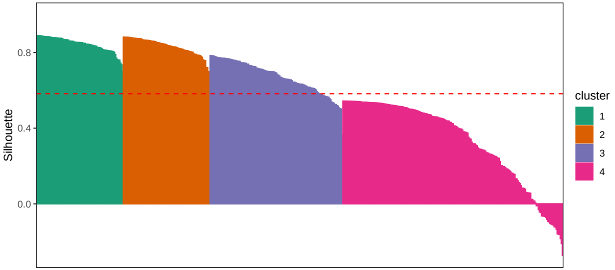
  Even though the clustering is obviously sub-optimal!
- Silhouette values for cluster 4 consistently very close to 1

  Makes sense since cluster 4 is detected nicely!
- Values for remaining clusters vary strongly.

Internal evaluation
Silhouette of DBSCAN on smiley

Note: noise filtered out (all negative values since these are treated as one single cluster!)



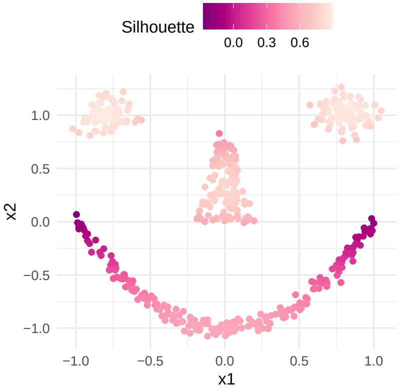


- Silhouette values for eyes and nose consistently high

  Makes sense since all are detected nicely!
- Silhouette values for mouth vary

  In line with the definition since the cluster is non-convex.

Internal evaluation
Silhouette of DBSCAN on smiley



Smiley data colored by silhouette values/widths of DBSCAN.

What we learned today

- k-means fails miserably on non-$\{$convex, spherical$\}$ cluster structure
- DBSCAN can effectively identify such clusters and identify outliers
- Quality measument is not easy!

References I

Murtagh, Fionn and Pedro Contreras (2012). "Algorithms for hierarchical clustering: an overview". In: *WIREs Data Mining and Knowledge Discovery* 2.1, pp. 86–97. DOI: https://doi.org/10.1002/widm.53.

Ester, Martin et al. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: AAAI Press, pp. 226–231.

Schubert, Erich et al. (July 2017). "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Trans. Database Syst.* 42.3. ISSN: 0362-5915. DOI: 10.1145/3068335.

Rand, William M. (1971). "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336, pp. 846–850. DOI: 10.1080/01621459.1971.10482356.

Dunn, J. C. (1974). "Well-Separated Clusters and Optimal Fuzzy Partitions". In: *Journal of Cybernetics* 4.1, pp. 95–104. DOI: 10.1080/01969727408546059.

Davies, David L. and Donald W. Bouldin (1979). "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2, pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909.

References II

Rousseeuw, Peter J. (1987). “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of Computational and Applied Mathematics* 20, pp. 53–65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7.