Data Mining Block “Clustering”

Myra Spiliopoulou [1]

FACULTY OF
COMPUTER SCIENCE

Clustering

is a family of mining algorithms,

designed to help you organize your data into groups of similar objects

Clustering

is a family of mining algorithms,
designed to help you organize your data into groups of similar objects

A clustering algorithm groups data points in such a way that the objects inside each cluster are more similar to each other than the objects outside the cluster.

Clustering

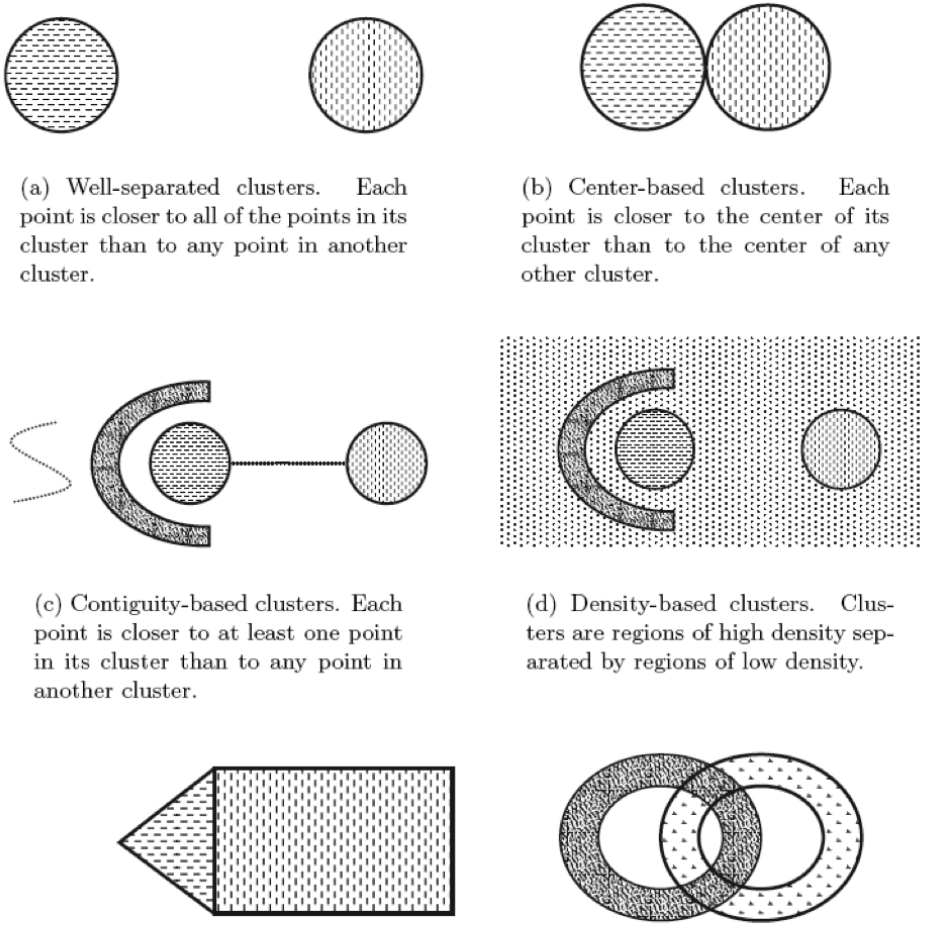
is a family of mining algorithms,
designed to help you organize your data into groups of similar objects

A clustering algorithm groups data points in such a way that the objects inside each cluster are more similar to each other than the objects outside the cluster.
For some clustering algorithms, this corresponds to

- minimizing the intra-cluster distance
- maximizing the inter-cluster distance

Different types of clusters

Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

Figure 8.2. Different types of clusters as illustrated by sets of two-dimensional points.

Goals of the Block

- Make yourself familiar with instruments that you can use to cluster data
- Learn how to choose the right instrument for your data
- Learn to recognize bad clusters

Clustering algorithms

- K-Means
- Hierarchical clustering
- Density-based clustering: DBSCAN

K-Means

K-means is good for a quick-and-dirty attempt to find groups in the data.

K-Means

INPUT: a set of data points D in feature space F; the number of clusters K

K-Means algorithm

1. Select K initial centroids
2. Repeat
   - Assign each point to the centroid closest to it
   - Recompute the positions of the K centroids

until the positions of the centroids do not change anymore.

K-Means

INPUT: a set of data points D in feature space F; the number of clusters K

K-Means algorithm

1. Select K initial centroids
2. Repeat
   - Assign each point to the centroid closest to it
   - Recompute the positions of the K centroids

until the positions of the centroids do not change anymore.

Optimization criterion for K-Means

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)^2$$

where c_i is the centroid of cluster C_i with $c_{ij} = \frac{\sum_{x \in C_i} x_j}{|C_i|}, j = 1 \ldots |F|$

K-Means

INPUT: a set of data points D in feature space F; the number of clusters K

K-Means algorithm

1. Select K initial centroids
2. Repeat
   - Assign each point to the centroid closest to it
   - Recompute the positions of the K centroids

until the positions of the centroids do not change anymore.

Optimization criterion for K-Means

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(x, c_i)^2$$

where c_i is the centroid of cluster C_i with $c_{ij} = \frac{\sum_{x \in C_i} x_j}{|C_i|}, j = 1 \ldots |F|$

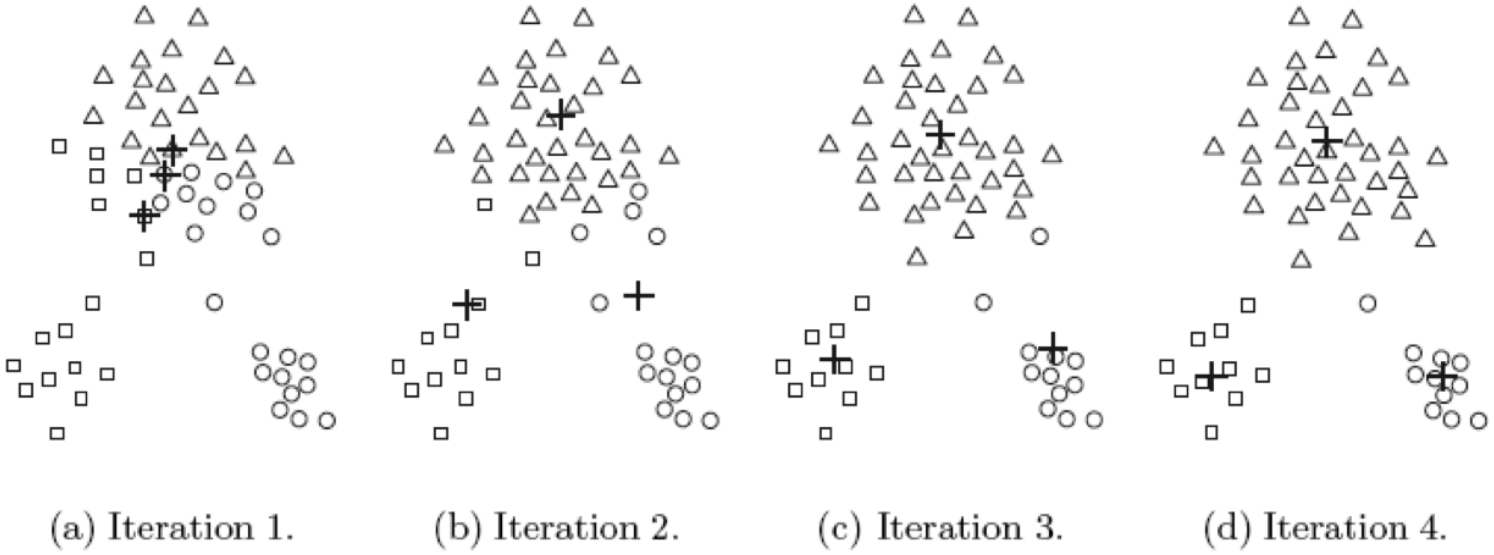
Complexity of K-Means

$O(K \cdot |F| \cdot |D| \cdot I)$ for feature space F, dataset D and number of iterations I

K-Means - Example

Building $K = 3$ clusters

Tan, Steinbach & Kumar, Ch.8 (2006)



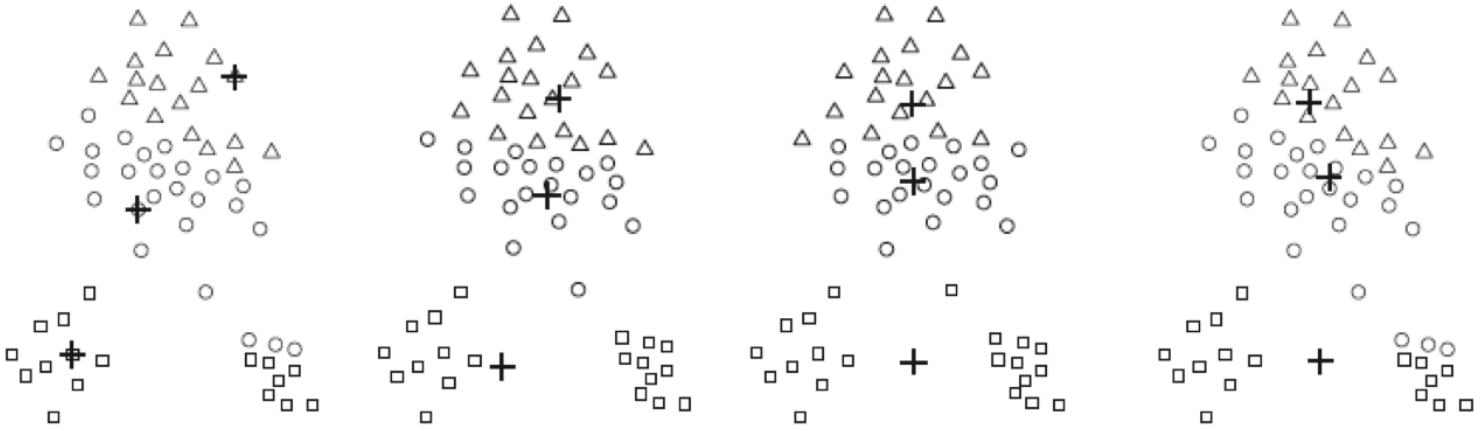
(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Figure 8.3. Using the K-means algorithm to find three clusters in sample data.

Example: same data, different initialization of centroids

Building $K = 3$ clusters

Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Figure 8.5. Poor starting centroids for K-means.

The K in K-Means

How to determine K ?

How to choose ”good” initial positions for the K centroids?

Disadvantages of K-Means

- K-Means favours spherical clusters
- K-Means favours clusters of the same size
- K-Means favours clusters of the same density
- K-Means is very sensitive to outlier objects

K-Means variant: Bisecting K-Means

Bisecting K-Means algorithm

Starting with a single root cluster that contains all objects in dataset D
REPEAT

1. Apply 2-Means on the cluster
2. Choose the most inhomogenenous of the leaf clusters
3. GOTO 1

UNTIL K leaf clusters have been built.

K-Means variant: Bisecting K-Means

Bisecting K-Means algorithm

Starting with a single root cluster that contains all objects in dataset D
REPEAT

1. Apply 2-Means on the cluster
2. Choose the most inhomogenenous of the leaf clusters
3. GOTO 1

UNTIL K leaf clusters have been built.

What advantage(s) brings Bisecting K-Means over K-Means?

Clustering algorithms

- √ K-Means
- Hierarchical clustering
- Density-based clustering: DBSCAN

Clustering algorithms

- √ K-Means
- Hierarchical clustering
- Density-based clustering: DBSCAN
- A small excursion on similarity functions and matrices

Similarity functions

Properties of a similarity function $s()$

For any two data points $x, y \in D$, it holds that:

- $s(x,y) \leq 1$ and $s(x,y) = 1 \leftrightarrow x = y$
- $s(x,y) = s(y,x)$

Similarity functions

Properties of a similarity function $s()$

For any two data points $x, y \in D$, it holds that:

- $s(x,y) \leq 1$ and $s(x,y) = 1 \leftrightarrow x = y$
- $s(x,y) = s(y,x)$

Properties of a distance function $d()$

For any two data points $x, y \in D$, it holds that:

- $d(x,y) \geq 0$ and $d(x,y) = 0 \leftrightarrow x = y$
- $d(x,y) = d(y,x)$
- For each $z \in D : d(x,z) \leq d(x,y) + d(y,z)$

Similarity functions

Properties of a similarity function $s()$

For any two data points $x, y \in D$, it holds that:

- $s(x,y) \leq 1$ and $s(x,y) = 1 \leftrightarrow x = y$
- $s(x,y) = s(y,x)$

Properties of a distance function $d()$

For any two data points $x, y \in D$, it holds that:

- $d(x,y) \geq 0$ and $d(x,y) = 0 \leftrightarrow x = y$
- $d(x,y) = d(y,x)$
- For each $z \in D : d(x,z) \leq d(x,y) + d(y,z)$

Very often, we use the complement of a distance function as a similarity function.

Similarity and Distance functions

Given are two data points $x, y \in D$ over an n-dimensional feature space F.

- $f1(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- $f2(x,y) = \sum_{i=1}^{n}|x_i - y_i|$
- $f3(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

Counting agreements and disagreements

Given are two data points $x, y \in D$ over an n-dimensional feature space F, so that the valuerange of each dimension $A \in F$ is the set $\{0, 1\}$.

- $agree_{11}(x, y) = \sum_{i=1}^{n} x_i y_i$

Counting agreements and disagreements

Given are two data points $x, y \in D$ over an n-dimensional feature space F, so that the valuerange of each dimension $A \in F$ is the set $\{0, 1\}$.

- $agree_{11}(x, y) = \sum_{i=1}^{n} x_i y_i$
- $agree_{00}(x, y) = \sum_{i=1}^{n} (1 - x_i)(1 - y_i)$

Counting agreements and disagreements

Given are two data points $x, y \in D$ over an n-dimensional feature space F, so that the valuerange of each dimension $A \in F$ is the set $\{0, 1\}$.

- $agree_{11}(x, y) = \sum_{i=1}^{n} x_i y_i$
- $agree_{00}(x, y) = \sum_{i=1}^{n} (1 - x_i)(1 - y_i)$
- $disagree_{10}(x, y) = \sum_{i=1}^{n} x_i (1 - y_i)$

Counting agreements and disagreements

Given are two data points $x, y \in D$ over an n-dimensional feature space F, so that the valuerange of each dimension $A \in F$ is the set $\{0, 1\}$.

- $agree_{11}(x, y) = \sum_{i=1}^{n} x_i y_i$
- $agree_{00}(x, y) = \sum_{i=1}^{n} (1 - x_i)(1 - y_i)$
- $disagree_{10}(x, y) = \sum_{i=1}^{n} x_i (1 - y_i)$
- $disagree_{01}(x, y) = \sum_{i=1}^{n} (1 - x_i) y_i$

Counting agreements and disagreements

Given are two data points $x, y \in D$ over an n-dimensional feature space F, so that the valuerange of each dimension $A \in F$ is the set $\{0, 1\}$.

- $agree_{11}(x,y) = \sum_{i=1}^{n} x_i y_i$
- $agree_{00}(x,y) = \sum_{i=1}^{n} (1-x_i)(1-y_i)$
- $disagree_{10}(x,y) = \sum_{i=1}^{n} x_i (1-y_i)$
- $disagree_{01}(x,y) = \sum_{i=1}^{n} (1-x_i) y_i$

$$RandIndex(x,y) = \frac{agree_{11}(x,y) + agree_{00}(x,y)}{agree_{11}(x,y) + agree_{00}(x,y) + disagree_{10}(x,y) + disagree_{01}(x,y)}$$

Counting agreements and disagreements

Given are two data points $x, y \in D$ over an n-dimensional feature space F, so that the valuerange of each dimension $A \in F$ is the set $\{0, 1\}$.

- $agree_{11}(x,y) = \sum_{i=1}^{n} x_i y_i$
- $agree_{00}(x,y) = \sum_{i=1}^{n} (1-x_i)(1-y_i)$
- $disagree_{10}(x,y) = \sum_{i=1}^{n} x_i (1-y_i)$
- $disagree_{01}(x,y) = \sum_{i=1}^{n} (1-x_i) y_i$

$$RandIndex(x,y) = \frac{agree_{11}(x,y) + agree_{00}(x,y)}{agree_{11}(x,y) + agree_{00}(x,y) + disagree_{10}(x,y) + disagree_{01}(x,y)}$$

$$JaccardCoefficient(x,y) = \frac{agree_{11}(x,y)}{agree_{11}(x,y) + disagree_{10}(x,y) + disagree_{01}(x,y)}$$

Similarity Matrix or Distance Matrix?

Example from Tan, Steinbach & Kumar, Ch.8 (2006)

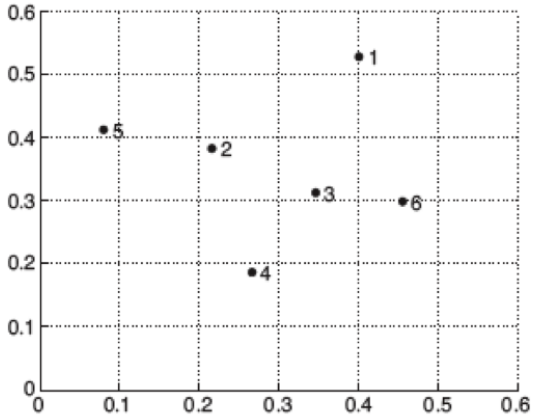


Figure 8.15. Set of 6 two-dimensional points.

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Table 8.3. xy coordinates of 6 points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.

Clustering algorithms

- √ K-Means
- √ Similarity functions
- Hierarchical clustering
- Density-based clustering: DBSCAN

Hierarchical clustering

This is a family of methods that processes your data to return a *tree of clusters*.
You run a horizontal cut on this tree to get a set of clusters.

Hierarchical clustering

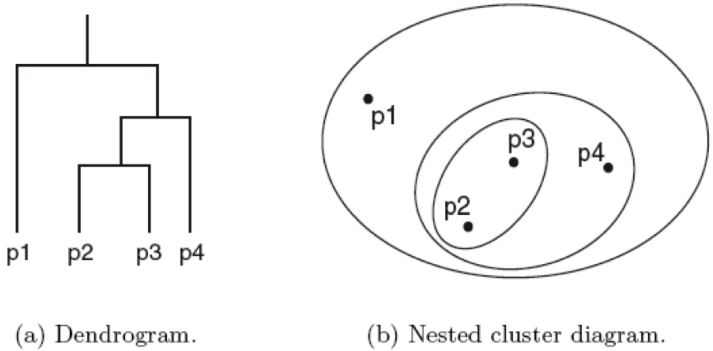
This is a family of methods that processes your data to return a *tree of clusters*.
You run a horizontal cut on this tree to get a set of clusters.

You use algorithms of this family because they are more robust, and you can easily switch one against the other, to fit the properties of your data better.

Hierarchical clustering

This is a family of methods that processes your data to return a *tree of clusters* - a dendrogramm.

Example of a dendrogramm — Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Dendrogram. (b) Nested cluster diagram.

Figure 8.13. A hierarchical clustering of four points shown as a dendrogram and as nested clusters.

You run a horizontal cut on this tree to get a set of clusters.

Bottom-up and top-down hierarchical clustering algorithms

Given is a set of data points D.

Bottom-up: Agglomerative clustering Tan, Steinbach & Kumar, Ch.8 (2006)

Starting with the bottom layer, where each data point is a cluster of its own:

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

Bottom-up and top-down hierarchical clustering algorithms

Given is a set of data points D.

Bottom-up: Agglomerative clustering Tan, Steinbach & Kumar, Ch.8 (2006)

Starting with the bottom layer, where each data point is a cluster of its own:

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

Top-down: Divisive clustering Tan, Steinbach & Kumar, Ch.8 (2006)

Starting with a single cluster that contains all data points:

repeat

Split one cluster into two

until k clusters remain

where k may be a user-defined number or k$= |D|$.

Agglomerative clustering algorithms

Clustering a small dataset — Tan, Steinbach & Kumar, Ch.8 (2006)

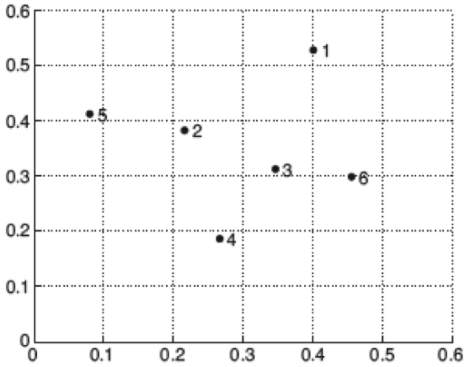


Figure 8.15. Set of 6 two-dimensional points.

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Table 8.3. xy coordinates of 6 points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.

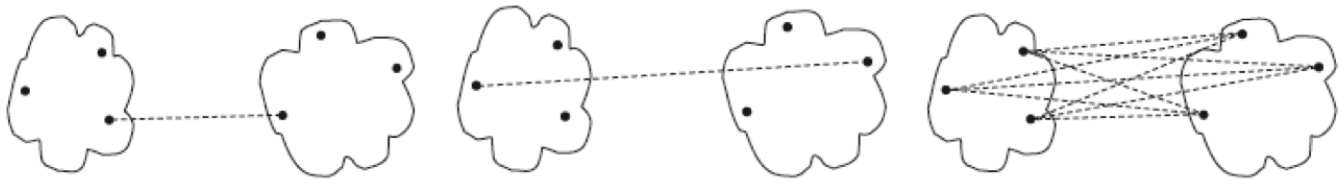
Which data points constitute the first cluster?

Agglomerative clustering algorithms

Agglomerative clustering algorithms differ in their definition of ”proximity between two clusters”.

- MIN - single link
- MAX - complete link
- Group average
- Distance between centroids
- Ward’s method - squared error

Computing proximity/distance — Tan, Steinbach & Kumar, Ch.8 (2006)



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

Figure 8.14. Graph-based definitions of cluster proximity

Agglomerative clustering algorithms:MIN

Distance between clusters X and Y

for each $x \in X$
    for each $y \in Y$
        compute $dist(x, y)$

sort the distance values into a list $L_{X,Y}$

Agglomerative clustering algorithms:MIN

Distance between clusters X and Y

for each $x \in X$
  for each $y \in Y$
    compute $dist(x, y)$

sort the distance values into a list $L_{X,Y}$

MIN – single link

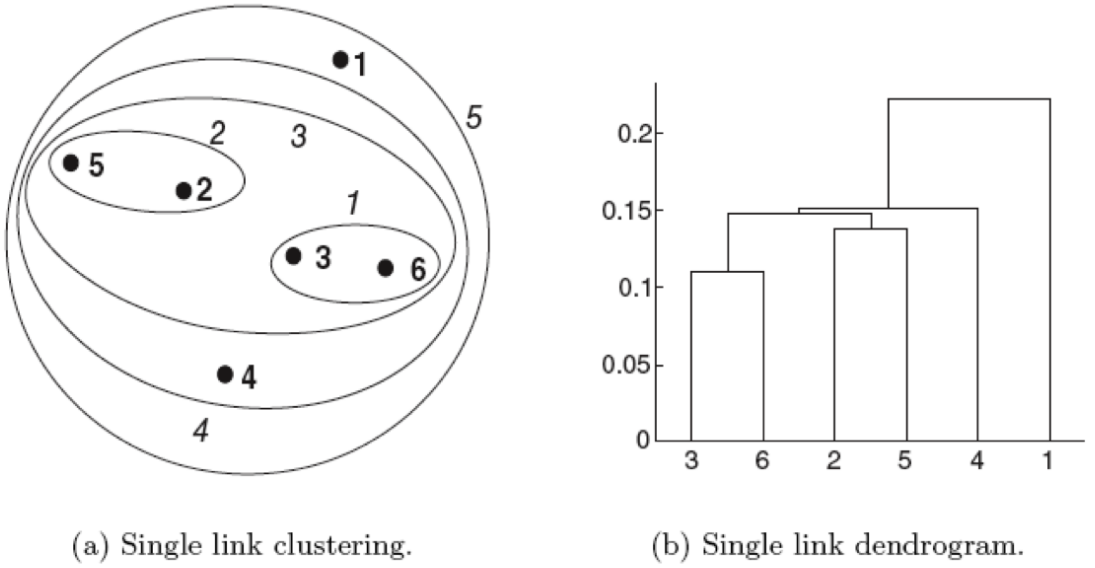
The distance between X, Y is the minimum value in $L_{X,Y}$.

$$d(X, Y) = \min L_{X,Y}$$

Agglomerative clustering algorithms: MIN

Applying MIN on the dataset

Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Single link clustering. (b) Single link dendrogram.

Figure 8.16. Single link clustering of the six points shown in Figure 8.15.

Agglomerative clustering algorithms: MAX

Distance between clusters X and Y

for each $x \in X$
  for each $y \in Y$
    compute $dist(x, y)$

sort the distance values into a list $L_{X,Y}$

MAX – complete link

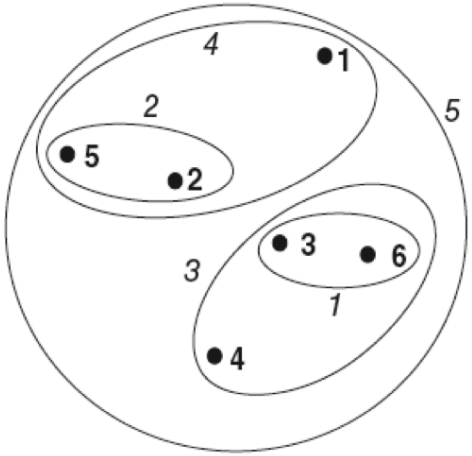
The distance between X, Y is the maximum value in $L_{X,Y}$:

$$d(X, Y) = \max L_{X,Y}$$

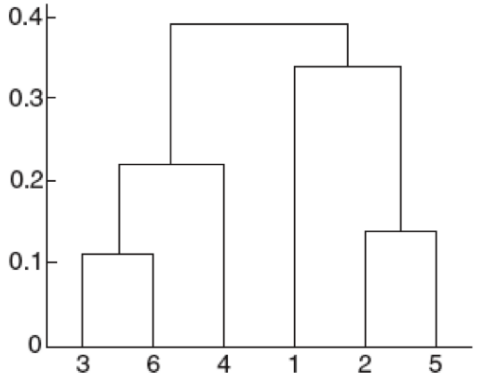
Agglomerative clustering algorithms: MAX

Applying MAX on the dataset

Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Complete link clustering.



(b) Complete link dendrogram.

Figure 8.17. Complete link clustering of the six points shown in Figure 8.15.

Agglomerative clustering algorithms: Group Average

Distance between clusters X and Y

for each $x \in X$
  for each $y \in Y$
    compute $dist(x, y)$

sort the distance values into a list $L_{X,Y}$

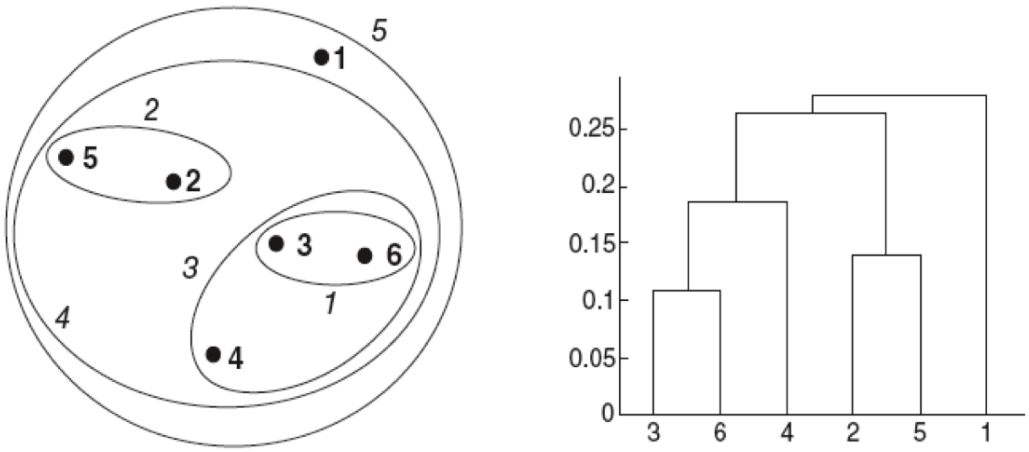
Group Average

The distance between X, Y is the average value in $L_{X,Y}$, i.e.

$$d(X, Y) = \frac{\sum_{x \in X}(\sum_{y \in Y} dist(x, y))}{|X| \cdot |Y|}$$

Agglomerative clustering algorithms: Group Average

Applying Group Average on the example dataset Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Group average clustering. (b) Group average dendrogram.

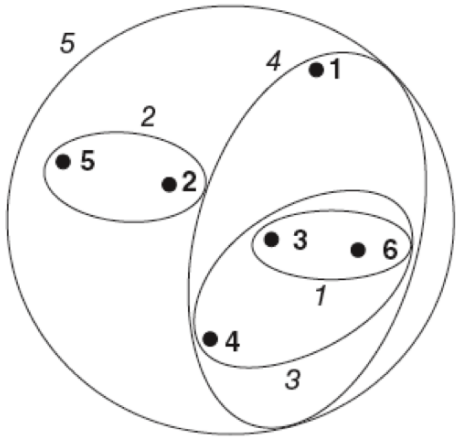
Figure 8.18. Group average clustering of the six points shown in Figure 8.15.

Agglomerative clustering algorithms: Ward's method

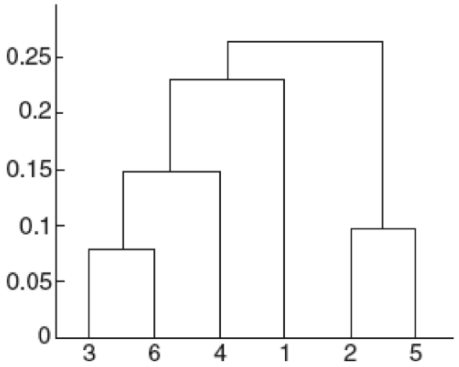
Ward's method

The distance between two clusters is the increase in the sum of squared errors that incurs when merging them.

Applying Ward's method on the example dataset Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Ward's clustering.



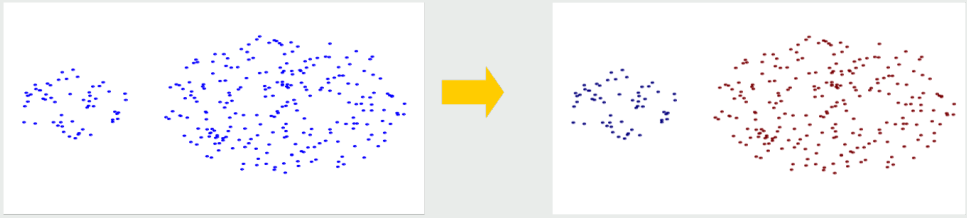
(b) Ward's dendrogram.

Figure 8.19. Ward's clustering of the six points shown in Figure 8.15.

Agglomerative clustering algorithms: Juxtaposing the methods

Advantages of MIN — Tan, Steinbach & Kumar, Ch.8 (2006)

MIN can discover non-spherical clusters



See also Fig. 8.2(c), contiguity-based clusters.

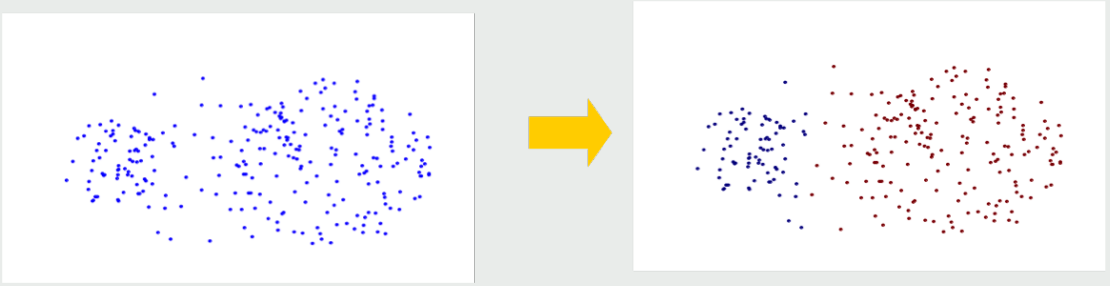
Disadvantages of MIN — Tan, Steinbach & Kumar, Ch.8 (2006)

MIN is sensitive to outliers and noise.

Agglomerative clustering algorithms: Juxtaposing the methods

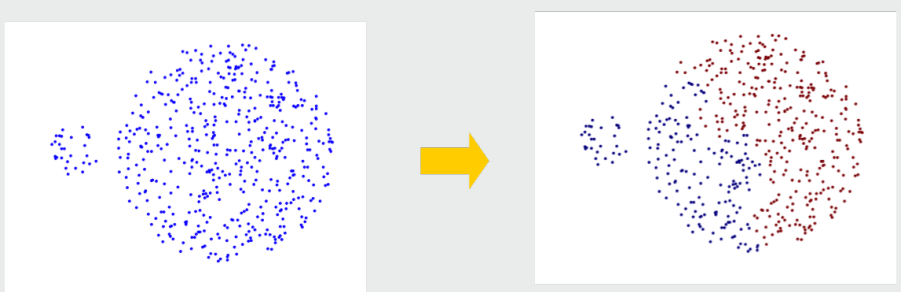
Advantages of MAX — Tan, Steinbach & Kumar, Ch.8 (2006)

MAX is not influenced by outliers and noise.



Disadvantages of MAX — Tan, Steinbach & Kumar, Ch.8 (2006)

MAX tends to split large clusters and favours spherical clusters.



Agglomerative clustering algorithms

Advantages:

+ Number of clusters needs not be specified.
+ Simple statistics and visualizations help deciding the cut position.

Disadvantages

− Complexity is $O(N^3)$ ($N = |D|$) (for some variants: $O(N^2 \log N)$).
− The order of merging affects the quality of subsequent iterations.

Dealing with outliers – quoting from the 2019 edition, p. 564-565:

> *Outliers pose the most serious problems for Ward's method and centroid-based hierarchical clustering approaches because they increase SSE and distort centroids. . . . As hierarchical clustering proceeds for these algorithms, outliers or small groups of outliers tend to form singleton or small clusters that do not merge with any other clusters until much later in the merging process. By discarding singleton or small clusters that are not merging with other clusters, outliers can be removed.*

Clustering algorithms

- √ K-Means
- √ Similarity functions
- √ Hierarchical clustering
- ► Density-based clustering: DBSCAN

Density-based clustering - DBSCAN

This is a family of methods that defines a cluster as a set of overlapping neighbourhoods.
DBSCAN [1] is the family progenitor.

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, 1996.

Density-based clustering - DBSCAN

This is a family of methods that defines a cluster as a set of overlapping neighbourhoods.
DBSCAN [1] is the family progenitor.

You use algorithms of this family when you know that your data have some dense areas surrounded by noise.

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, 1996.

Density-based clustering - DBSCAN

This is a family of methods that defines a cluster as a set of overlapping neighbourhoods.
DBSCAN [1] is the family progenitor.

You use algorithms of this family when you know that your data have some dense areas surrounded by noise.
These algorithms are best for geographical data, but there are further application areas.

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, 1996.

Density-based clustering with DBSCAN

The core concept of DBSCAN is that of a neighbourhood around a data point.

Densely populated neighbourhoods form the basis for a cluster.

A cluster is a maximal set of overlapping, densely populated neighbourhoods.

Density-based clustering with DBSCAN

The core concept of DBSCAN is that of a neighbourhood around a data point.

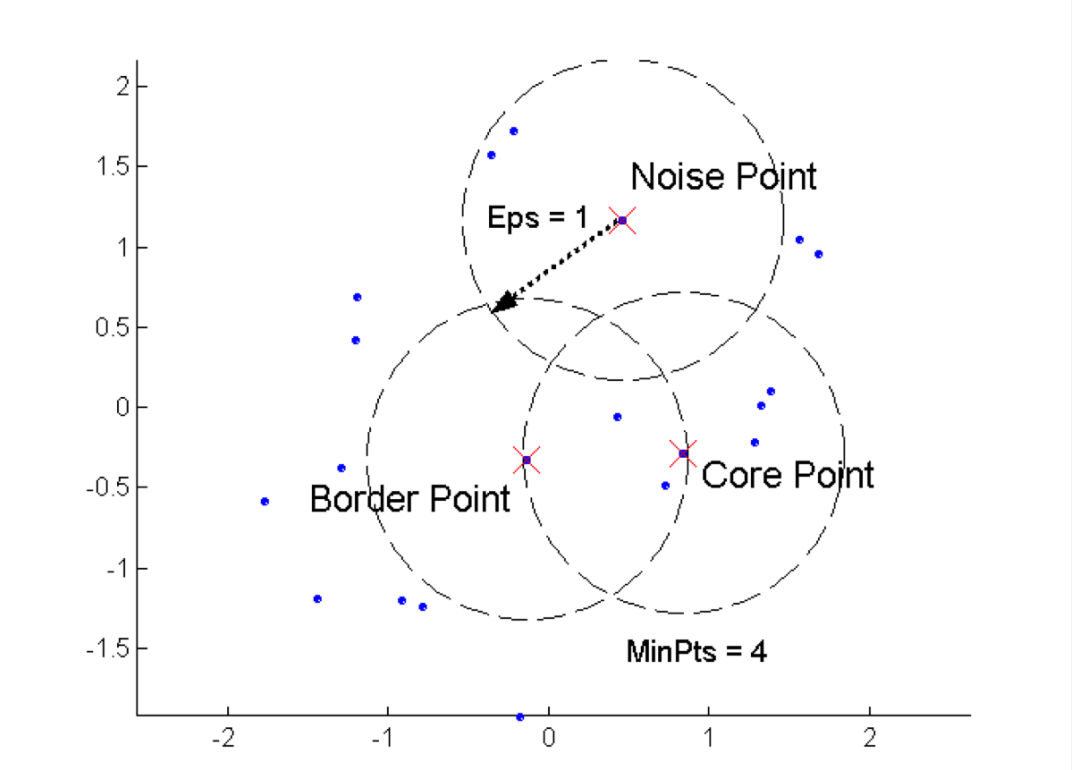
Densely populated neighbourhoods form the basis for a cluster.

A cluster is a maximal set of overlapping, densely populated neighbourhoods.

DBSCAN does not place all data points to clusters. Around each cluster there are sparsely populated areas, which are not part of any cluster.

DBSCAN

DBSCAN basic concepts — Tan, Steinbach & Kumar, Ch.8 (2006)



A point x is a core point, given eps and $minPts$, iff there are at least $minPts$ points within radius eps around x. A point y is a border point if there is a core point x, such that $dist(x, y) \leq eps$. All other points are noise points.

DBSCAN

DBSCAN parameters

- eps: radius of the hypersphere [a] around a data point, i.e. the *neighbourhood* of the data point
- $minPts$: neighbourhood size – can be defined as
  - number of neighbours of the data point in the center of the hypersphere
  - number of data points inside the hypersphere

[a] The hypersphere is defined in the feature space F.

The concept of *reachability* in DBSCAN

Given are two data points x, y in the dataset D over feature space F, and fixed parameter values for $eps, minPts$:

Directly density-reachable data point

y is *directly-density reachable* from x iff

- x is a core point AND
- y is in the neighbourhood of x

The concept of *reachability* in DBSCAN

Given are two data points x, y in the dataset D over feature space F, and fixed parameter values for $eps, minPts$:

Directly density-reachable data point

y is *directly-density reachable* from x iff

- x is a core point AND
- y is in the neighbourhood of x

Density-reachable data point

y is *density reachable* from x iff there are $x_1, \ldots, x_n \in D$ such that:

- $x = x_1$ and $y = x_n$ AND
- for each $i = 2, \ldots, n$ it holds that x_i is directly density-reachable from x_{i-1}

The concept of *reachability* in DBSCAN

Given are two data points x, y in the dataset D over feature space F, and fixed parameter values for $eps, minPts$:

Directly density-reachable data point

y is *directly-density reachable* from x iff

- x is a core point AND
- y is in the neighbourhood of x

Density-reachable data point

y is *density reachable* from x iff there are $x_1, \ldots, x_n \in D$ such that:

- $x = x_1$ and $y = x_n$ AND
- for each $i = 2, \ldots, n$ it holds that x_i is directly density-reachable from x_{i-1}

Density-connected data points

x, y are density-connected iff there is a data point $z \in D$ such that:

- x is density-reachable from z AND
- y is density-reachable from z

The concept of *cluster* in DBSCAN

Given is a dataset D over feature space F, and fixed parameter values for $eps, minPts$.

Cluster in DBSCAN

A cluster $C \subseteq D$ is a non-empty subset of D that has following properties:

- *Connectivity:* for each $x, y \in C$ it holds that x, y are density-connected.
- *Maximality:* for each $x, y \in D$ such that $x \in C$ and x, y are density-connected, it holds that $y \in C$.

The concept of *cluster* in DBSCAN

Given is a dataset D over feature space F, and fixed parameter values for $eps, minPts$.

Cluster in DBSCAN

A cluster $C \subseteq D$ is a non-empty subset of D that has following properties:

- *Connectivity:* for each $x, y \in C$ it holds that x, y are density-connected.
- *Maximality:* for each $x, y \in D$ such that $x \in C$ and x, y are density-connected, it holds that $y \in C$.

The noise areas

Let $C_1, \ldots, C_k$ be all the clusters found by DBSCAN for D.

$$noise(D) = D \setminus \cup_{i=1}^{k} C_i$$

DBSCAN

DBSCAN - simplified

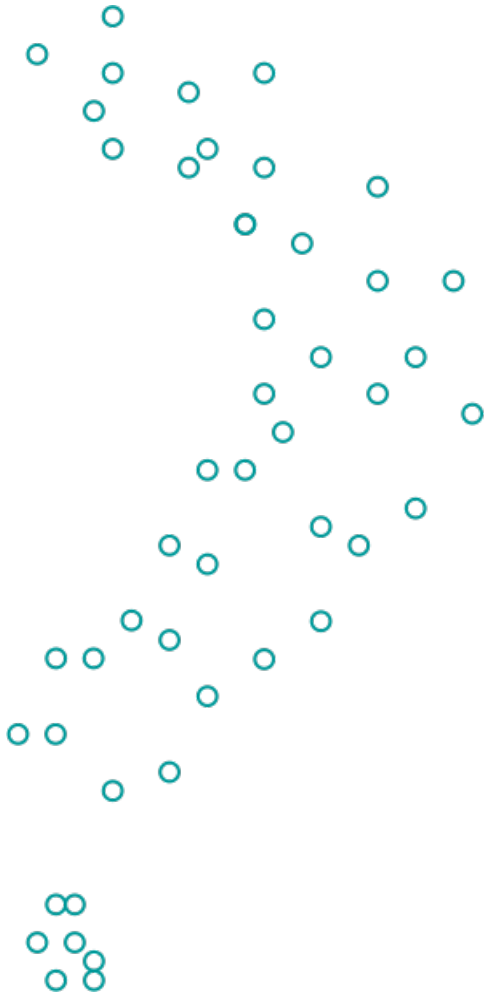
Tan, Steinbach & Kumar, Ch.8 (2006)

Algorithm 8.4 DBSCAN algorithm.

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

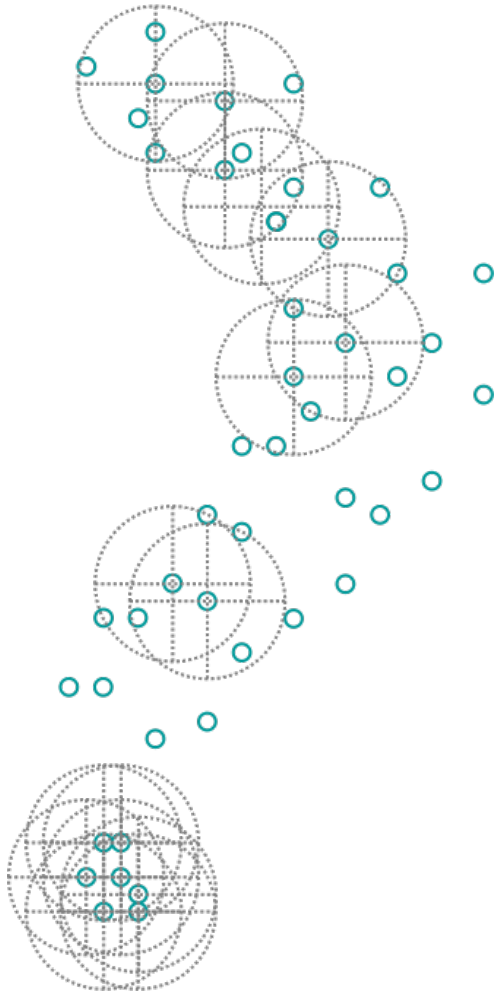
DBSCAN

Example (1 of 3): the dataset



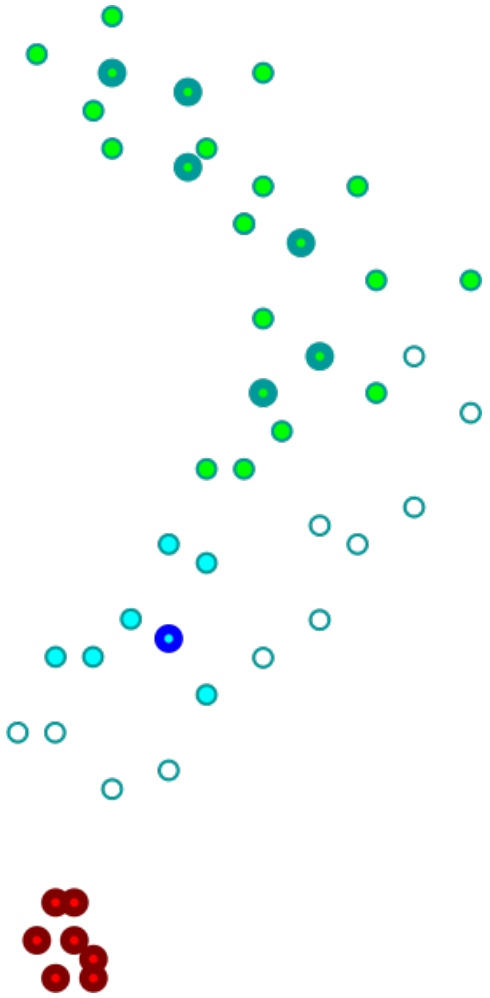
DBSCAN

Example (2 of 3) for $eps = 1$, $minPts = 5$: core points and neighbourhoods



DBSCAN

Example (3 of 3) for $eps = 1$, $minPts = 5$: clusters and noise areas [2]



[2] Beware of a mistake: the blue cluster has two core points, not one.

DBSCAN

Advantages of DBSCAN

- DBSCAN can find clusters of arbitrary geometrical form
- DBSCAN is insensitive to noise

DBSCAN

Advantages of DBSCAN

- DBSCAN can find clusters of arbitrary geometrical form
- DBSCAN is insensitive to noise

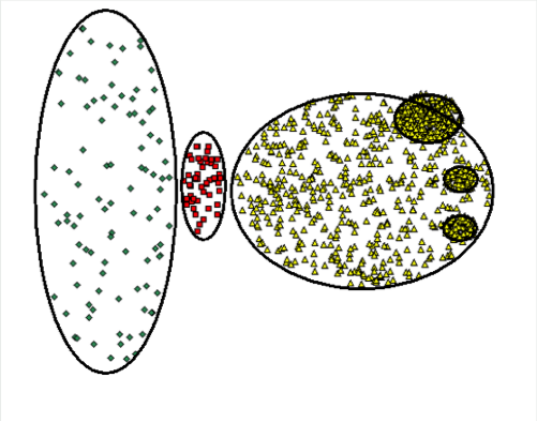
Disadvantages of DBSCAN

- DBSCAN runs into difficulties when there are dense areas with different density

DBSCAN: Disadvantage

Example dataset

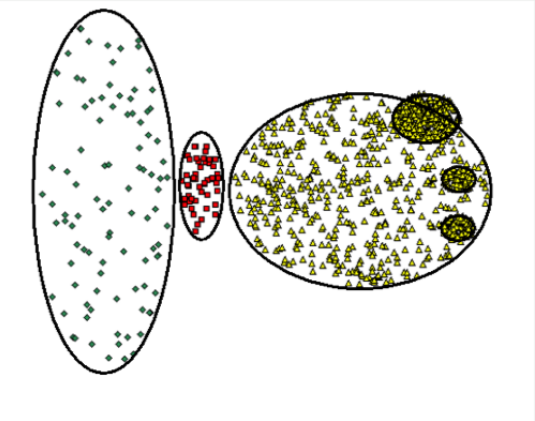
Tan, Steinbach & Kumar, Ch.8 (2006)



DBSCAN: Disadvantage

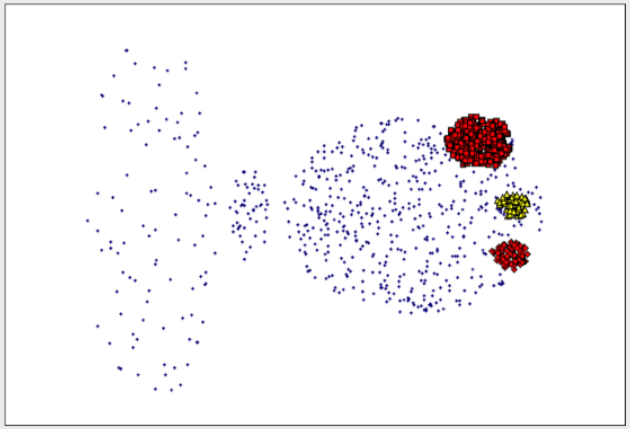
Example dataset

Tan, Steinbach & Kumar, Ch.8 (2006)

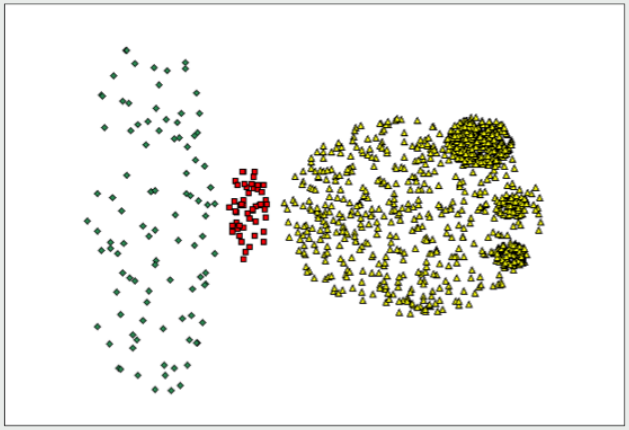


Two runs for $MinPts = 4$

Tan, Steinbach & Kumar, Ch.8 (2006)



$eps = 9.75$



$eps = 9.92$

Clustering algorithms

- √ K-Means
- √ Similarity functions
- √ Hierarchical clustering
- √ Density-based clustering: DBSCAN

Clustering algorithms

- √ K-Means
- √ Similarity functions
- √ Hierarchical clustering
- √ Density-based clustering: DBSCAN

How to find out whether the clustering algorithm has built good clusters?

What to evaluate when you do clustering?

What to evaluate when you do clustering?

- How good are the clusters?
- How good is the algorithm?

Evaluation in Clustering

- Evaluation of cluster quality 1 $\longrightarrow$ Internal Indices
- Evaluation of cluster quality 2 $\longrightarrow$ Models of randomness
- Evaluation of algorithm performance $\longrightarrow$ External Indices

Internal indices of cluster quality

- Sum of Squared Errors
- Cohesion and Separation
- Silhouette Coefficient

Internal indices of cluster quality

For a clustering ζ over a set D, computed with a distance function $d()$ (the distances are normalized):

- Sum of Squared Errors

Quality on the basis of distance of data points to "their" centroid

- $\forall X \in \zeta, x \in X : pointSSE(x) = d(x, center(X))^2$

Internal indices of cluster quality

For a clustering ζ over a set D, computed with a distance function $d()$ (the distances are normalized):

- Sum of Squared Errors

Quality on the basis of distance of data points to "their" centroid

- $\forall X \in \zeta, x \in X : pointSSE(x) = d(x, center(X))^2$
- $\forall X \in \zeta : cluSSE(X) = \frac{1}{|X|} \sum_{x \in X} pointSSE(x)$

Internal indices of cluster quality

For a clustering ζ over a set D, computed with a distance function $d()$ (the distances are normalized):

- Sum of Squared Errors

Quality on the basis of distance of data points to "their" centroid

- $\forall X \in \zeta, x \in X : pointSSE(x) = d(x, center(X))^2$
- $\forall X \in \zeta : cluSSE(X) = \frac{1}{|X|} \sum_{x \in X} pointSSE(x)$
- $SSE(\zeta) = \frac{1}{|\zeta|} \sum_{X \in \zeta} cluSSE(X) = \frac{1}{|\zeta|} \sum_{X \in \zeta} \left(\frac{1}{|X|} \sum_{x \in X} d(x, center(X))^2 \right)$

Internal indices of cluster quality

For a clustering ζ over a set D, computed with a distance function $d()$ (the distances are normalized):

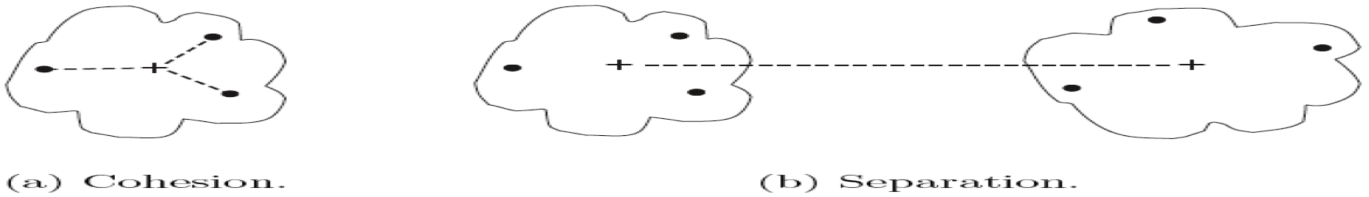
- Cohesion and Separation

Quality as in-cluster homogeneity and between-clusters gap

1. Centroid-based indices: $\forall X \in \zeta$
   - $cohesion(X) = 1 - \frac{1}{|X|} \sum_{x \in X} d(x, center(X))$
   - $separation(X) = \frac{1}{|\zeta|-1} \sum_{Y \in \zeta, Y \neq X} d(center(X), center(Y))$

   where values closer to 1 are better.

   Example from Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Cohesion. (b) Separation.

Figure 8.28. Prototype-based view of cluster cohesion and separation.

Internal indices of cluster quality

For a clustering ζ over a set D, computed with a distance function $d()$ (the distances are normalized):

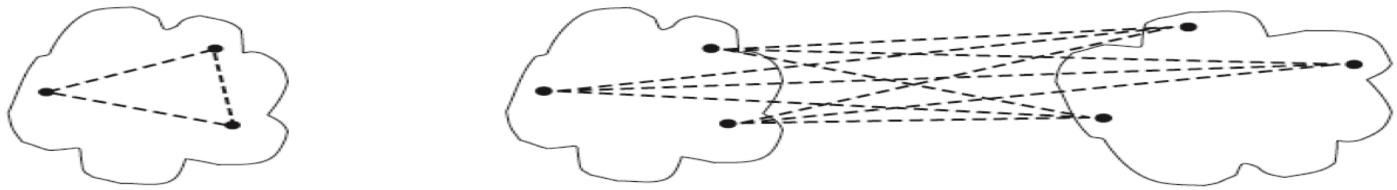
- Cohesion and Separation

Quality as in-cluster homogeneity and between-clusters gap

2. Graph-based indices: $\forall X \in \zeta$
   - $cohesion(X) = 1 - \frac{1}{|X|(|X|-1)} \sum_{x,y \in X; x \neq y} d(x,y)$
   - $separation(X) = \frac{1}{|\zeta|-1} \sum_{Y \in \zeta \setminus X} \frac{1}{|X| \cdot |Y|} \left(\sum_{x \in X, y \in Y} d(x,y) \right)$

   where values closer to 1 are better.

   Example from Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Cohesion. (b) Separation.

Figure 8.27. Graph-based view of cluster cohesion and separation.

Evaluation of cluster quality

Internal indices of cluster quality:

- √ Sum of Square Errors
- √ Cohesion and Separation: center-based and graph-based
- √ Silhouette Coefficient

and a visualization aid: Similarity matrix sorted on clusterID

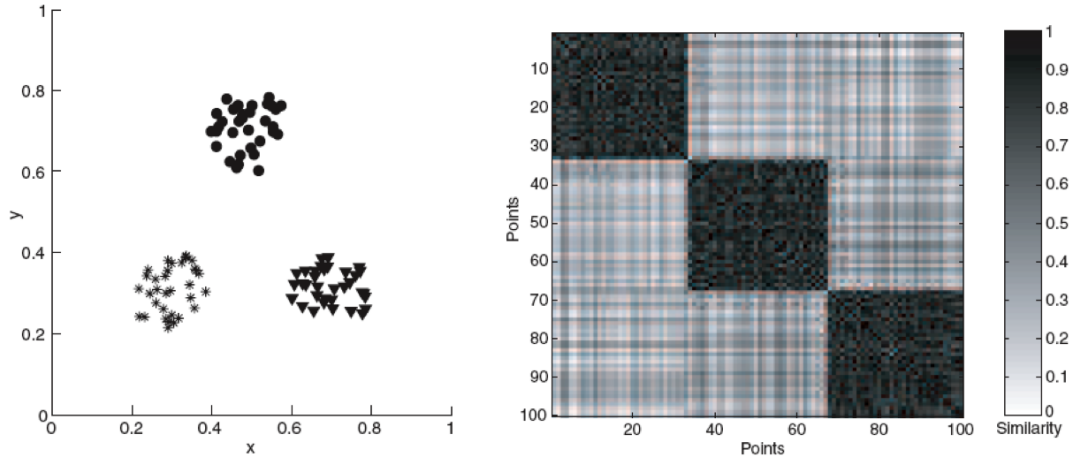
Evaluation of cluster quality

Internal indices of cluster quality:

- √ Sum of Square Errors
- √ Cohesion and Separation: center-based and graph-based
- √ Silhouette Coefficient

and a visualization aid: Similarity matrix sorted on clusterID

A clustered dataset — Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Well-separated clusters.

(b) Similarity matrix sorted by K-means cluster labels.

Figure 8.30. Similarity matrix for well-separated clusters.

Evaluation of cluster quality and the problem of random data

Another clustered dataset — Tan, Steinbach & Kumar, Ch.8 (2006)

(a) Original points.

(b) Three clusters found by DBSCAN.

(c) Three clusters found by K-means.

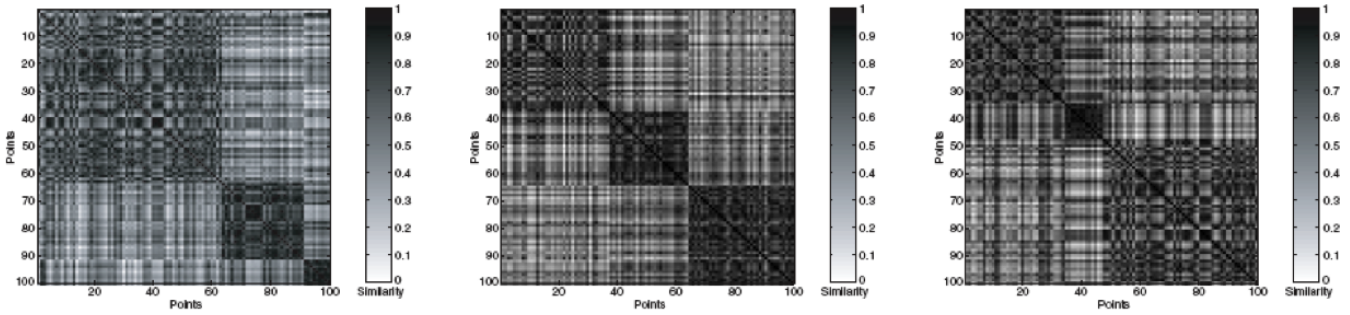
(d) Three clusters found by complete link.

Figure 8.26. Clustering of 100 uniformly distributed points.

Evaluation of cluster quality and the problem of random data

Example continued

Tan, Steinbach & Kumar, Ch.8 (2006)



(a) Similarity matrix sorted by DBSCAN cluster labels.

(b) Similarity matrix sorted by K-means cluster labels.

(c) Similarity matrix sorted by complete link cluster labels.

Figure 8.31. Similarity matrices for clusters from random data.

Evaluation of cluster quality and the problem of random data

The data we cluster may actually have NO clustering structure.

A clustering algorithm run on those data will in any case return some clusters.

? When is the value of the internal index good enough ?

? How to recognize whether the boxes in the similarity matrix are indeed clusters ?

Evaluation in Clustering

- √ Evaluation of cluster quality 1 $\longrightarrow$ Internal Indices
- Evaluation of cluster quality 2 $\longrightarrow$ Models of randomness

Evaluation of cluster quality with help of *Models of Randomness*

Models of Randomness - Approach 1 from Tan, Steinbach & Kumar, Ch.8 (2006)

Given is a dataset D with feature space F, and a set of clusters ξ learned with algorithm $\mathscr{A}$.

FOR $i = 1 \ldots N$

- generate a random dataset D_i, so that $|D_i| = |D|$ and $F_i = F$
- derive a model ζ_i for D_i using $\mathscr{A}$ with the same parameter settings as used for ζ.
- compute the quality of ζ_i with an internal index $q()$

DO compute the histogramm of the quality values of the N models

DO compare $q(\xi)$ with the values on the histogramm

DO DISCARD ξ if it is in the wrong area of the plot ???

Evaluation of cluster quality with help of *Models of Randomness*

Approach 1 - Example from Tan, Steinbach & Kumar, Ch.8 (2006)

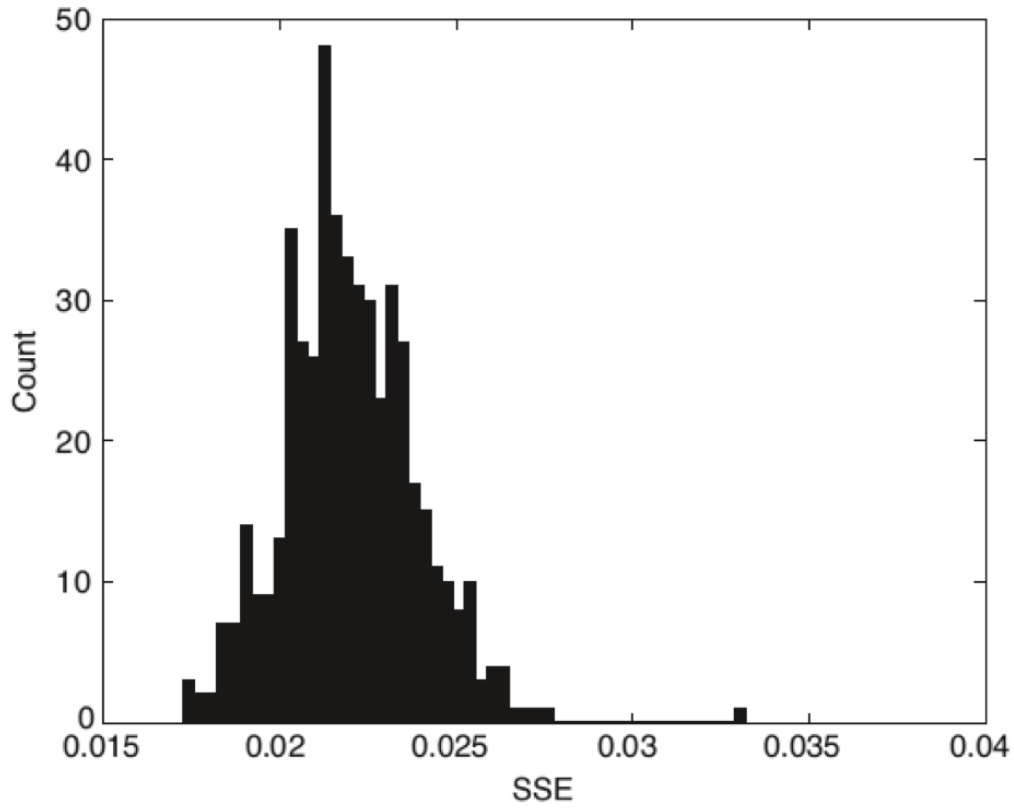


Figure 8.34. Histogram of SSE for 500 random data sets.

When is the SSE value of the model ξ good?

Evaluation of cluster quality with help of *Models of Randomness*

Models of Randomness - Approach 2 — Uli Niemann (Master thesis)[2]

Given is a dataset D with feature space F, and a set of clusters ξ learned with algorithm $\mathscr{A}$.

A model-of-randomness over D, F, i.e. $MofR(D, F)$ is a set of clusters ζ, such that $|\zeta| = |\xi|$ and the assignment of points to clusters in ζ is random.

FOR $i = 1 \ldots N$, generate a $MofR(D, F)$, ζ_i

DO compute the histogramm of the quality values of the N models

DO compare $q(\xi)$ with the values on the histogramm

DO DISCARD ξ if it is in the wrong area of the plot

[3] Niemann, U., Spiliopoulou, M., Völzke, H., and Kühn, J.-P. (2014). Subpopulation Discovery in Epidemiological Data with Subspace Clustering. *Foundations of Computing and Decision Sciences (FCDS)*, 39(4):271–300.

Evaluation in Clustering

- √ Evaluation of cluster quality 1 $\longrightarrow$ Internal Indices
- √ Evaluation of cluster quality 2 $\longrightarrow$ Models of randomness
- ▶ Evaluation of algorithm performance $\longrightarrow$ External Indices

External indices

Two groups of external indices:

Group 1: How well did the clustering algorithm guess the true classes of the data?

- Entropy
- Purity
- Precision & Recall — BLOCK: Classification
- F-measure — BLOCK: Classification

Group 2: To what extend do the clusters agree with the classes?

- √ Rand Index
- √ Jaccard Coefficient

where higher values are better.

External indices group 1 - class-oriented indices

Given is the dataset D, the true assignment of the data points in D to classes $\varphi = \{C_1, \ldots, C_L\}$ and a set of clusters $\xi = \{X_1, \ldots, X_K\}$ over D.

Entropy

$$entropy(\xi, \varphi) = \sum_{i=1}^{K} \frac{|X_i|}{|D|} clusEntropy(X_i, \varphi)$$

where $clusEntropy(X_i, \varphi) = -\sum_{j=1}^{L} p_{ij} \log_2(p_{ij})$ with $p_{ij} = \frac{|X_i \cap C_j|}{|X_i|}$.

External indices group 1 - class-oriented indices

Given is the dataset D, the true assignment of the data points in D to classes $\varphi = \{C_1, \ldots, C_L\}$ and a set of clusters $\xi = \{X_1, \ldots, X_K\}$ over D.

Entropy

$$entropy(\xi, \varphi) = \sum_{i=1}^{K} \frac{|X_i|}{|D|} clusEntropy(X_i, \varphi)$$

where $clusEntropy(X_i, \varphi) = -\sum_{j=1}^{L} p_{ij} \log_2(p_{ij})$ with $p_{ij} = \frac{|X_i \cap C_j|}{|X_i|}$.

Purity

$$purity(\xi, \varphi) = \sum_{i=1}^{K} \frac{|X_i|}{|D|} clusPurity(X_i, \varphi)$$

where $clusPurity(X_i, \varphi) = max_{j=1\ldots L}\{p_{ij}\}$,
i.e. each cluster X_i is assigned to the class, to which most of its members belong.

Closing

- √ Basic clustering algorithms: K-Means, DBSCAN, hierarchical clustering methods
- √ Similarity/distance functions for clustering
- √ Evaluation of internal cluster quality
- √ Evaluation of algorithm performance towards a ground truth

Closing

Clustering encompasses much more:

- More basic algorithms like K-Medoids
- Algorithms that perceive the data as a graph: Graph clustering algorithms
- Algorithms that build clusters in subsets of the feature space:
  - Subspace clustering
  - Projected clustering
- Algorithms that exploit knowledge of the expert:
  - Constraint-based clustering algorithms
  - Semi-supervised clustering algorithms
- Algorithms that assign the data probabilistically to groups
  - Probabilistic clustering algorithms
  - Fuzzy clustering algorithms, like C-Means
  - Topic models (generative models over the data space)

Thank you very much!

Questions?