Clustering Methods

Farnoosh Khodakarami

This material is prepared with Ali Madani and Farnoosh Khodakarami

DataSets

Features(Attribute/variable)

Data records (samples)

ID	Address	# Bed	#Bath	...	School Score	Year Build	Crime Rate
				...			
				...			

 # Features = Dimension of dataset

Unsupervised versus supervised learning

- **Supervised learning**
  - Trying to predict label or value of data points

- **Unsupervised learning**
  - Unlabeled input data

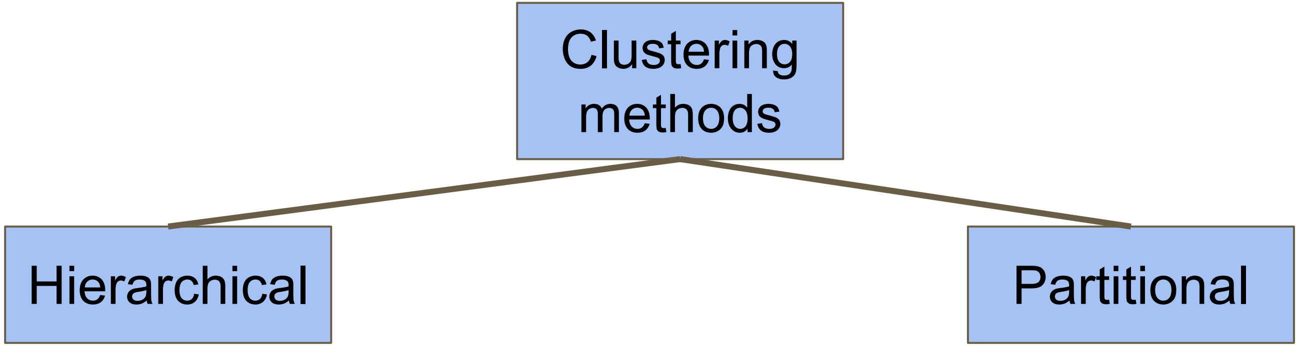
Why do we use clustering?

- Group data points based on their similarities using the given features
  - Identify similar data points (within the same group)
  - Identify dissimilar data points (within different groups)

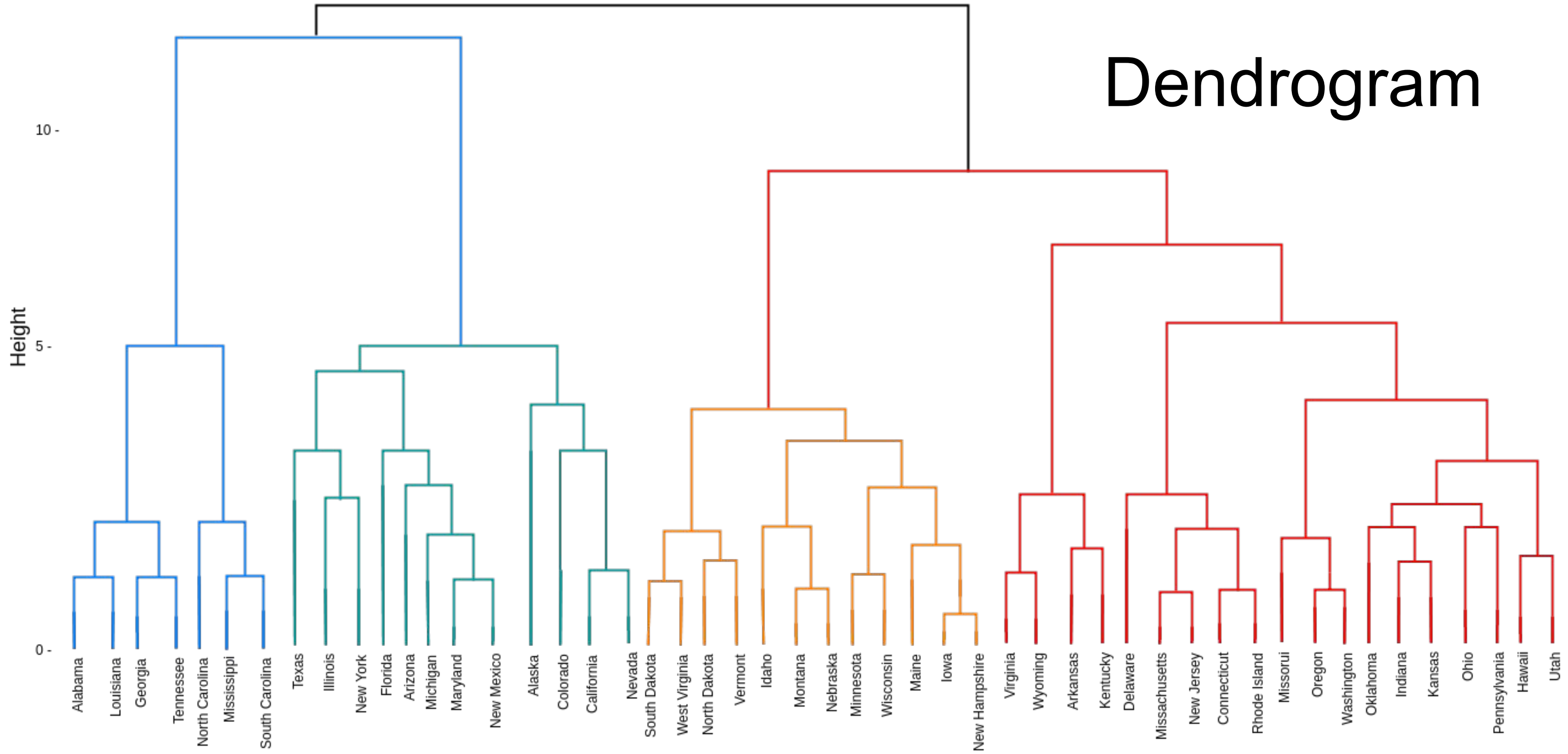
- Some methods also identifies outliers and points that cannot be assigned to any cluster

- Investigating differences between the groups
  - Survival of breast cancer patients

Categories of clustering methods that we focus on



Agglomerative hierarchical clustering



https://online.visual-paradigm.com/diagrams/templates/dendrogram/cluster-dendrogram/

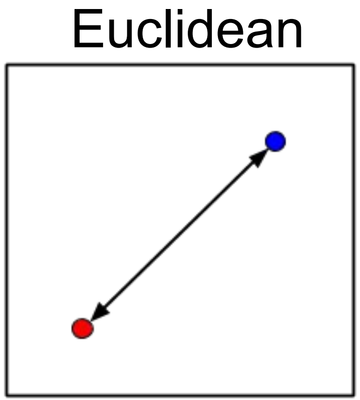
Steps of agglomerative hierarchical clustering

1. Computing dissimilarity or similarity between every pair of data points
2. Using linkage function to group objects into hierarchical cluster tree
   a. linkage function determines the distance between sets of data points as a function of the pairwise distances between data points in the groups.
3. Deciding where to cut the hierarchical tree into clusters.

Common distance measures used in clustering

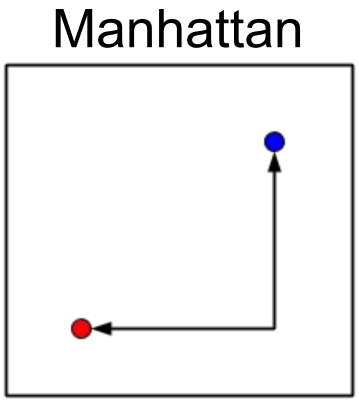
- Euclidean distance
- Manhattan distance
- Minkowski distance
- Chebychev distance
- Cosine similarity
- Hamming distance
- Binary distance
  - Jaccard index
  - Hamming distance

Euclidean distance



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan distance

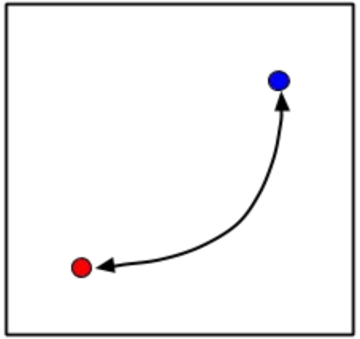


$$d = |x_1 - x_2| + |y_1 - y_2|$$

Minkowski distance

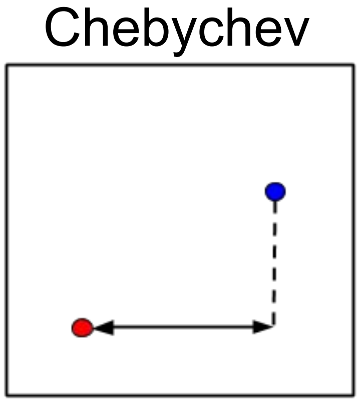
$$d = \left(|x_1 - x_2|^p + |y_1 - y_2|^p\right)^{1/p}$$

Minkowski



Chebychev distance

$$d = max(|x_1 - x_2|, |y_1 - y_2|)$$



Jaccard index

$$J(A, B) = \frac{A \cap B}{A \cup B}$$

Jaccard

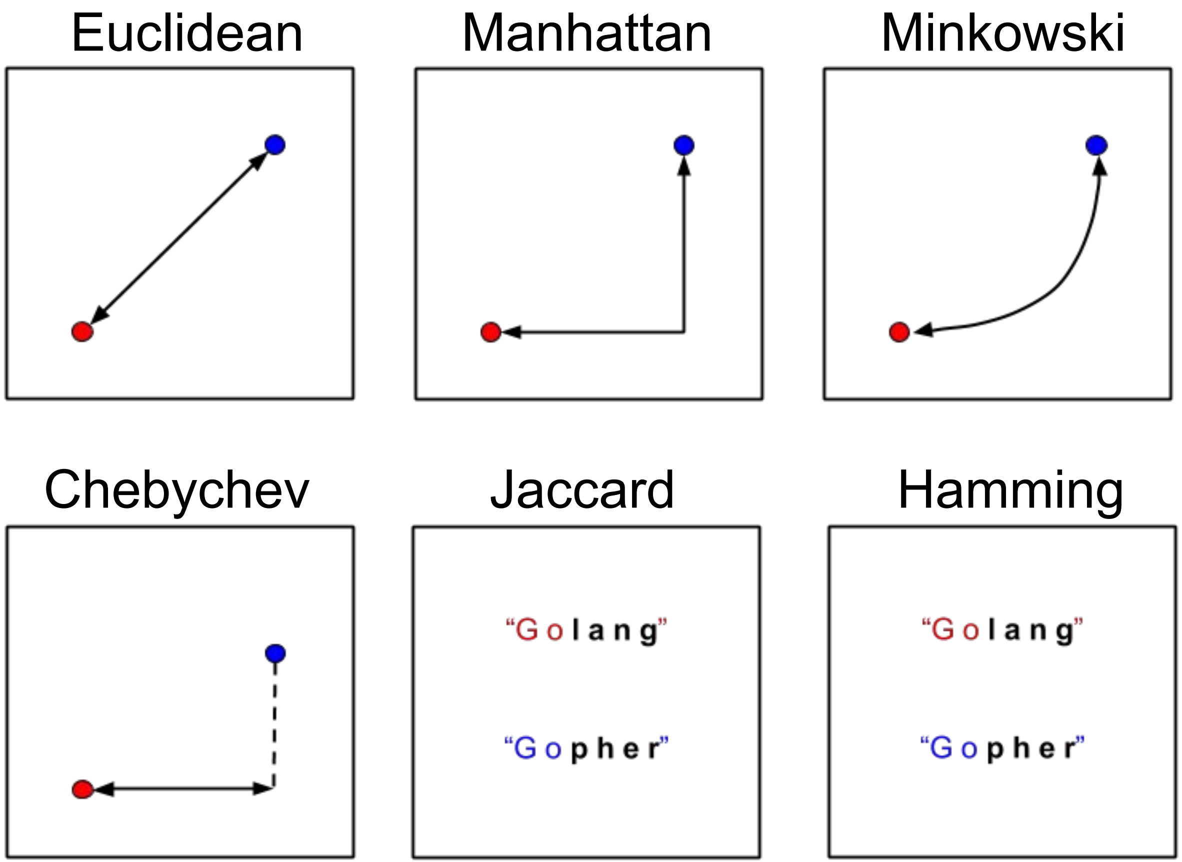
https://subscription.packtpub.com/

Hamming distance

d=***number of bits that they are different***

Hamming

Different distance measures



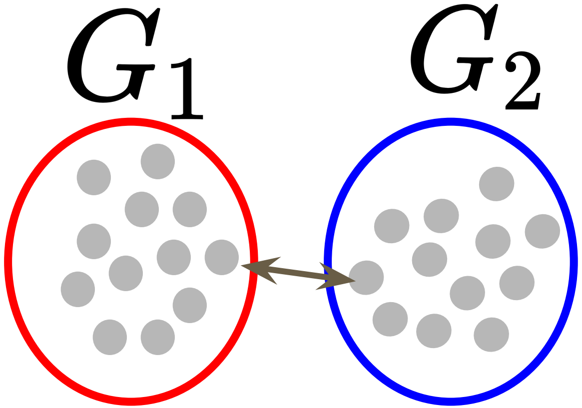
https://subscription.packtpub.com/

Some of widely-used linkage functions

- Single
- Complete
- Average
- Median
- Centroid

Some of widely-used linkage functions

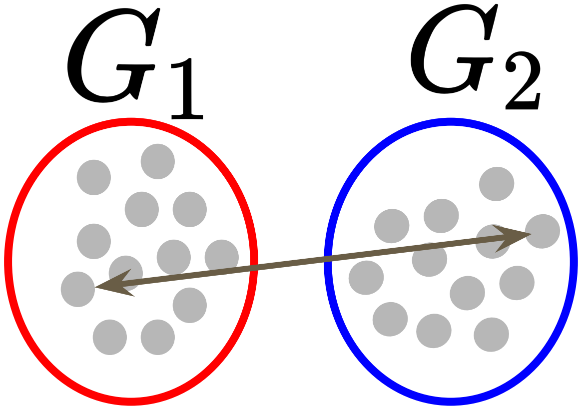
- **Single**
- Complete
- Average
- Median
- Centroid



$$D(G_1, G_2) = min(d(x, y)), x \in G_1, y \in G_2$$

Some of widely-used linkage functions

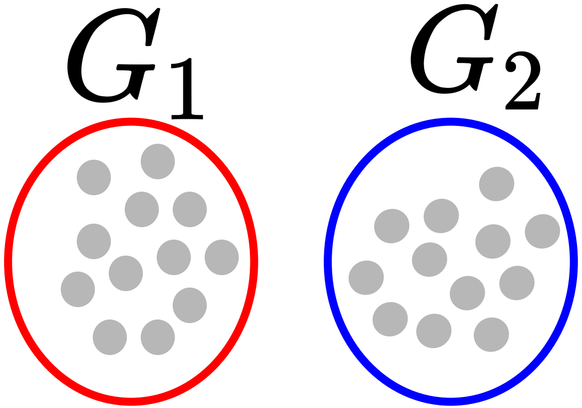
- Single
- **Complete**
- Average
- Median
- Centroid



$$D(G_1, G_2) = max(d(x,y)), x \in G_1, y \in G_2$$

Some of widely-used linkage functions

- Single
- Complete
- **Average**
- Median
- Centroid

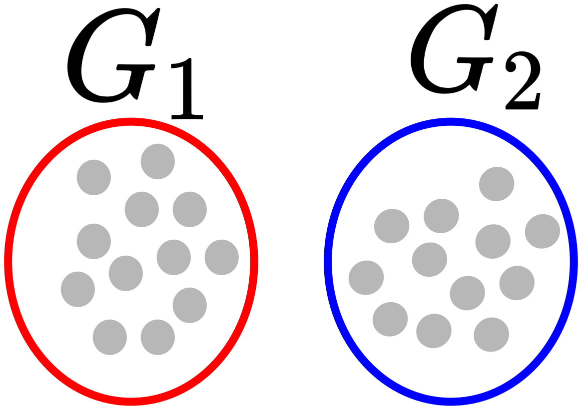


$$D(G_1, G_2) = \frac{1}{|N_{G_1}||N_{G_2}|} \Sigma_x \Sigma_y d(x, y)$$

$$x \in G_1, y \in G_2$$

Some of widely-used linkage functions

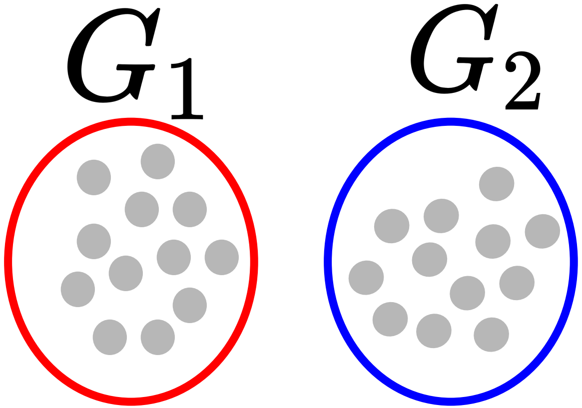
- Single
- Complete
- Average
- **Median**
- Centroid



Weighted center of mass distance (WPGMC)
Note. appropriate for Euclidean distances only

Some of widely-used linkage functions

- Single
- Complete
- Average
- Median
- **Centroid**

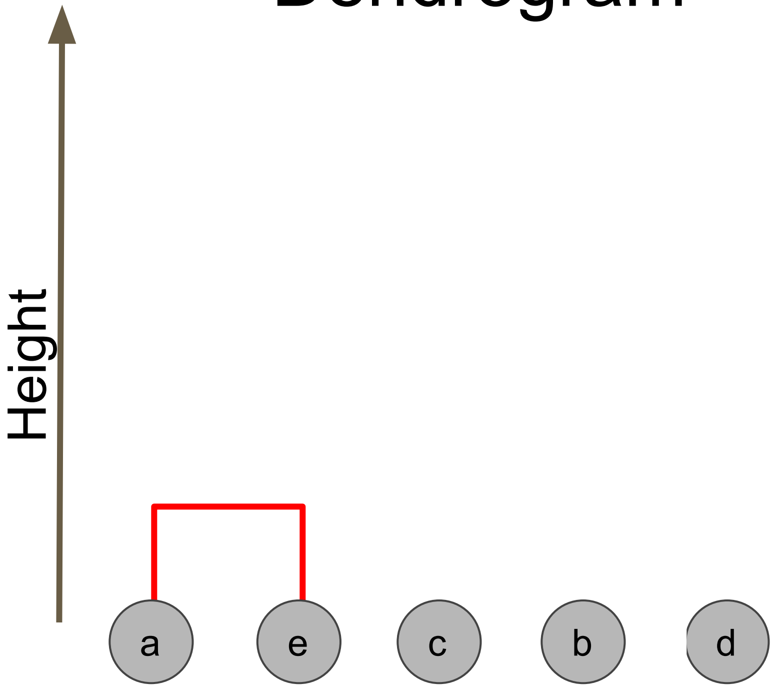


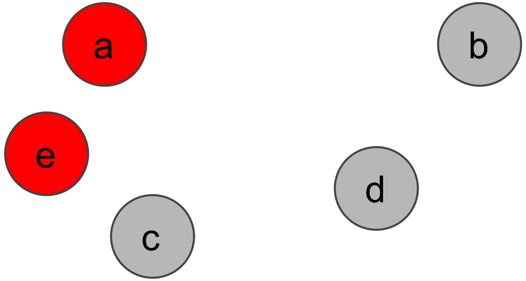
$$D(G_1, G_2) = ||c_{G_1} - c_{G_2}||$$

Note. appropriate for Euclidean distances only

Let’s see how agglomerative hierarchical clustering works

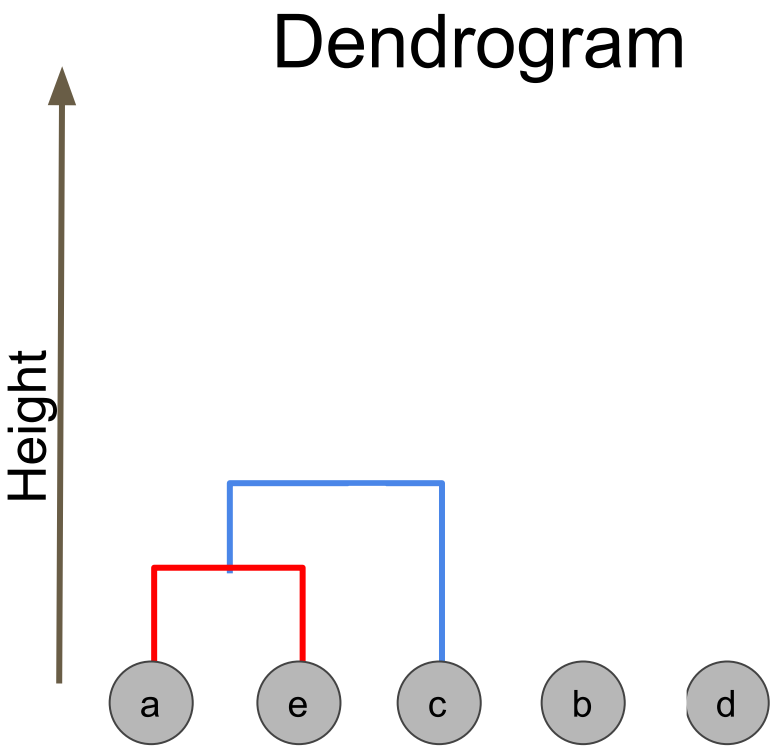
Dendrogram

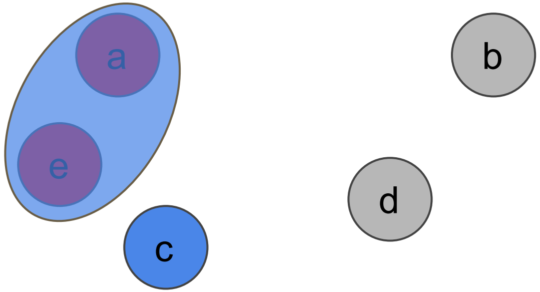




Data points should be rearranged for building the dendrogram

Let’s see how agglomerative hierarchical clustering works

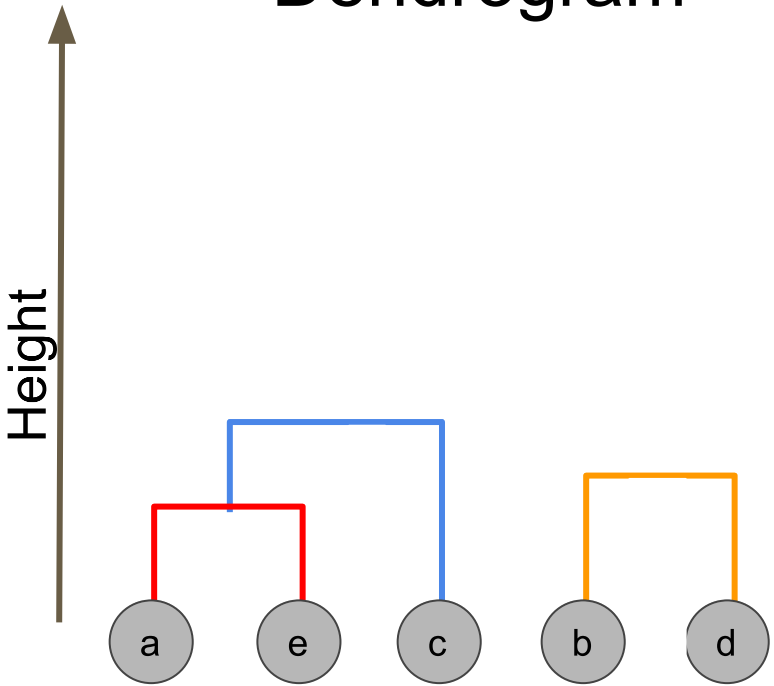


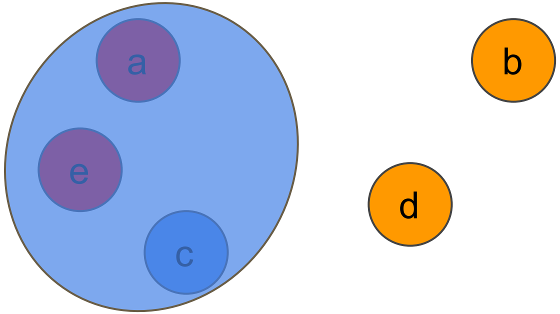


Data points should be rearranged for building the dendrogram

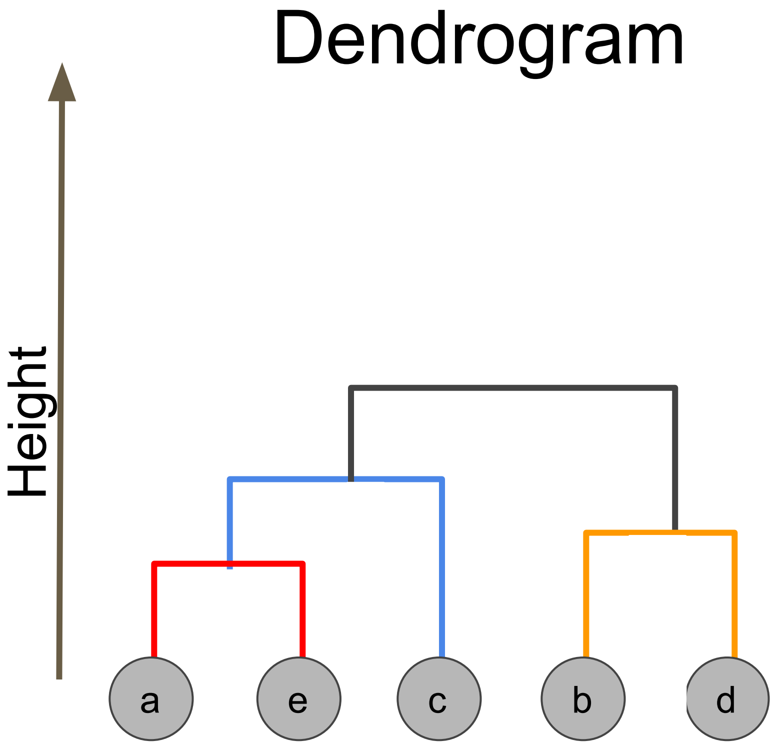
Let’s see how agglomerative hierarchical clustering works

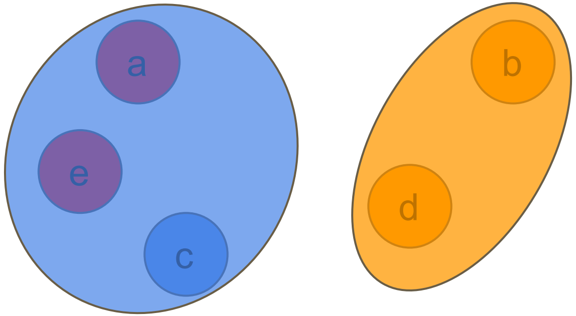
Dendrogram





Let's see how agglomerative hierarchical clustering works





Hierarchical versus partitional clustering

- Partitional clustering: division of the set of data points into clusters
  - each data object belongs to one subset

- Hierarchical clustering is a set of nested clusters
  - organized as a tree

K-means clustering

Steps of k-means clustering

1) Choosing k

K-means clustering

Steps of k-means clustering

1) Choosing k
2) Randomly selecting k data points (as initial centers)
    a) Final result depend on these points
        i) Because k-means converges to one of many possible local minima

K-means clustering

Steps of k-means clustering

1) Choosing k
2) Randomly selecting k data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)

K-means clustering

Steps of k-means clustering

1) Choosing k
2) Randomly selecting k data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)
4) Assign each data point to the nearest center

K-means clustering

Steps of k-means clustering

1) Choosing k
2) Randomly selecting k data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)
4) Assign each data point to the nearest center
5) Calculate new center of each cluster

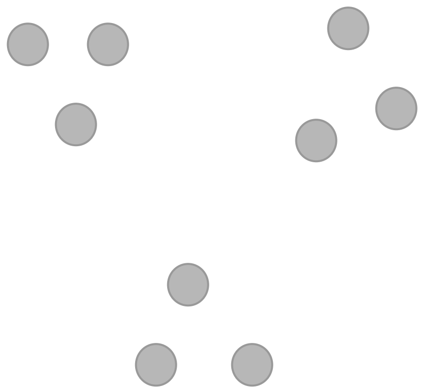
K-means clustering

Steps of k-means clustering

1) Choosing k
2) Randomly selecting k data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)
4) Assign each data point to the nearest center
5) Calculate new center of each cluster
6) Repeat steps (3) to (5) until convergence
    a) No point in changing cluster membership

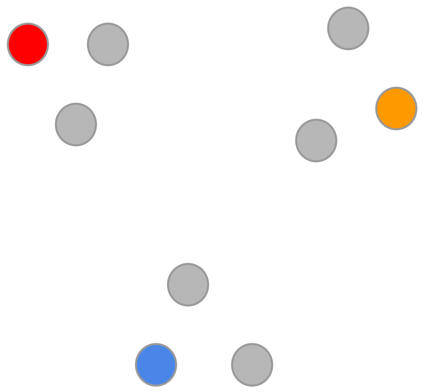
Implementing k-means manually (simple example)

1) Choosing k (good guess k=3)



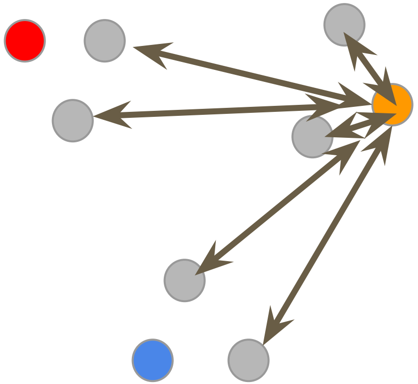
Implementing k-means manually (simple example)

1) Choosing k (good guess k=3)
2) Randomly selecting 3 data points (as initial centers)



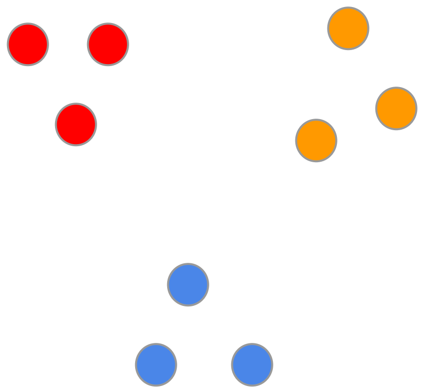
Implementing k-means manually (simple example)

1) Choosing k (good guess k=3)
2) Randomly selecting 3 data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)



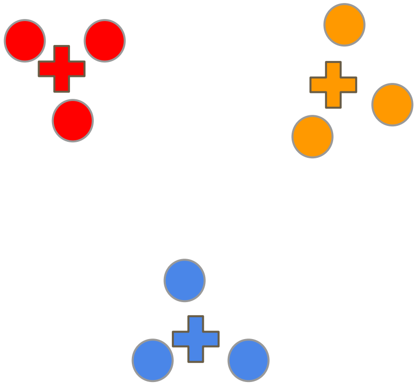
Implementing k-means manually (simple example)

1) Choosing k (good guess k=3)
2) Randomly selecting 3 data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)
4) Assign each data point to the nearest center



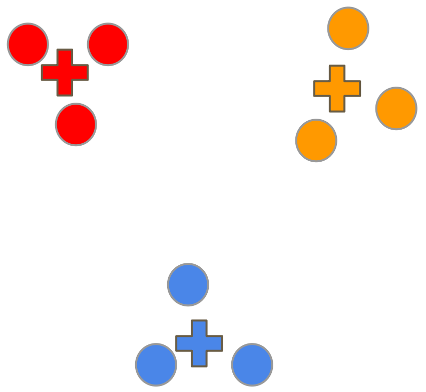
Implementing k-means manually (simple example)

1) Choosing k (good guess k=3)
2) Randomly selecting 3 data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)
4) Assign each data point to the nearest center
5) Calculate new center of each cluster



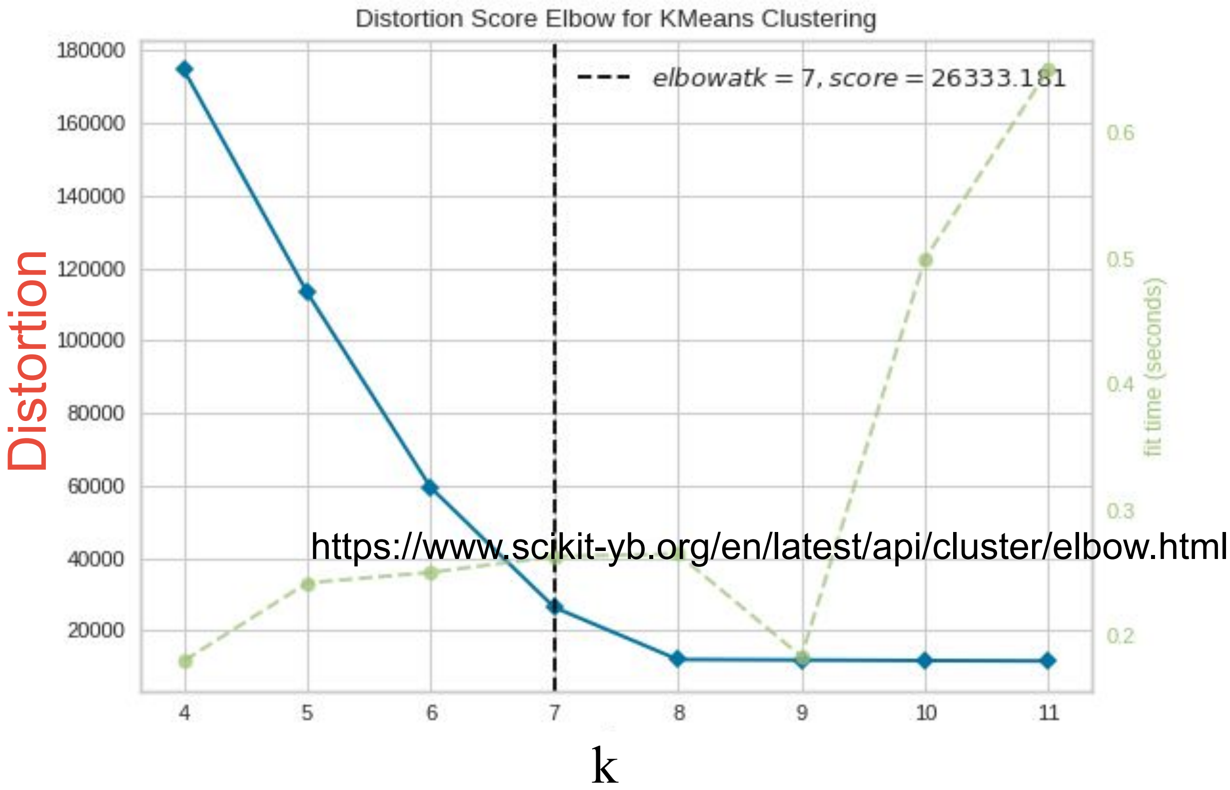
Implementing k-means manually (simple example)

1) Choosing k (good guess k=3)
2) Randomly selecting 3 data points (as initial centers)
3) Calculating distance of every data point to the chosen centers in step (2)
4) Assign each data point to the nearest center
5) Calculate new center of each cluster
6) No need for repetition
   a) Good choice of initial centers
      i) Fast convergence



Elbow method for selecting optimal K

Distortion is the sum of squared distances from each point to its assigned center



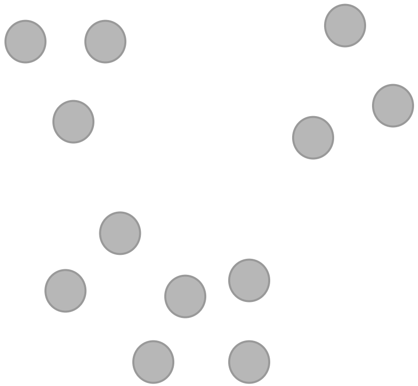
Hierarchical k-means (divisive hierarchical clustering)

- Start with all the data points
- Implement k-means (k=2) in an iterative manner
- Until reaching singletons (data points)

Hierarchical k-means (divisive hierarchical clustering)

Issue with this process:

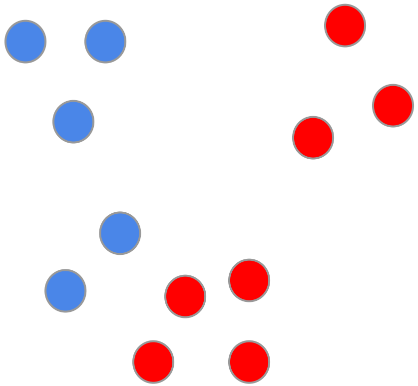
- Some real clusters could be dismissed because of iterative k-means



Hierarchical k-means (divisive hierarchical clustering)

Issue with this process:

- Some real clusters could be dismissed because of iterative k-means

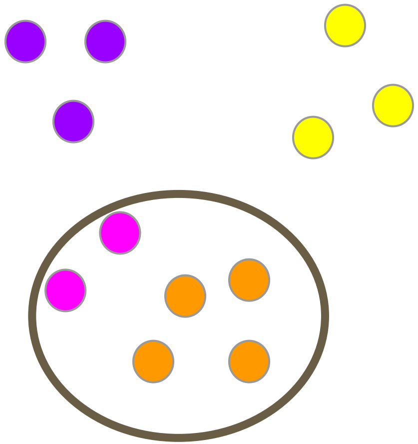


Hierarchical k-means (divisive hierarchical clustering)

Issue with this process:

- Some real clusters could be dismissed because of iterative k-means

We like these to be one cluster



Mini-Batch K-Means

Mini-Batch K-Means is a modified version of k-means that makes updates to the cluster centroids using mini-batches of samples rather than the entire dataset, which can make it faster for large datasets, and perhaps more robust to statistical noise.

Affinity propagation

- Number of clusters (k) does not need to be determined

Affinity propagation

- Number of clusters (k) does not need to be determined
- Using similarity matrix
    - Need pairwise similarity between data points
        - Similarity between data points i and j: negative euclidean distance between data points i and j

Affinity propagation

- Number of clusters (k) does not need to be determined
- Using similarity matrix
    - Need pairwise similarity between data points
        - Similarity between data points i and j: negative euclidean distance between data points i and j
- Determines clusters and representative data points for the clusters (exemplars)

Affinity propagation

- Number of clusters (k) does not need to be determined
- Using similarity matrix
    - Need pairwise similarity between data points
        - Similarity between data points i and j: negative euclidean distance between data points i and j
- Determines clusters and representative data points for the clusters (exemplars)
- Iterative massage passing
    - Data points compete to become exemplars
        - Exemplars are real data points not centers as in k-means

Affinity propagation

- Number of clusters (k) does not need to be determined
- Using similarity matrix
    - Need pairwise similarity between data points
        - Similarity between data points i and j: negative euclidean distance between data points i and j
- Determines clusters and representative data points for the clusters (exemplars)
- Iterative massage passing
    - Data points compete to become exemplars
        - Exemplars are real data points not centers as in k-means
- Initialization independent

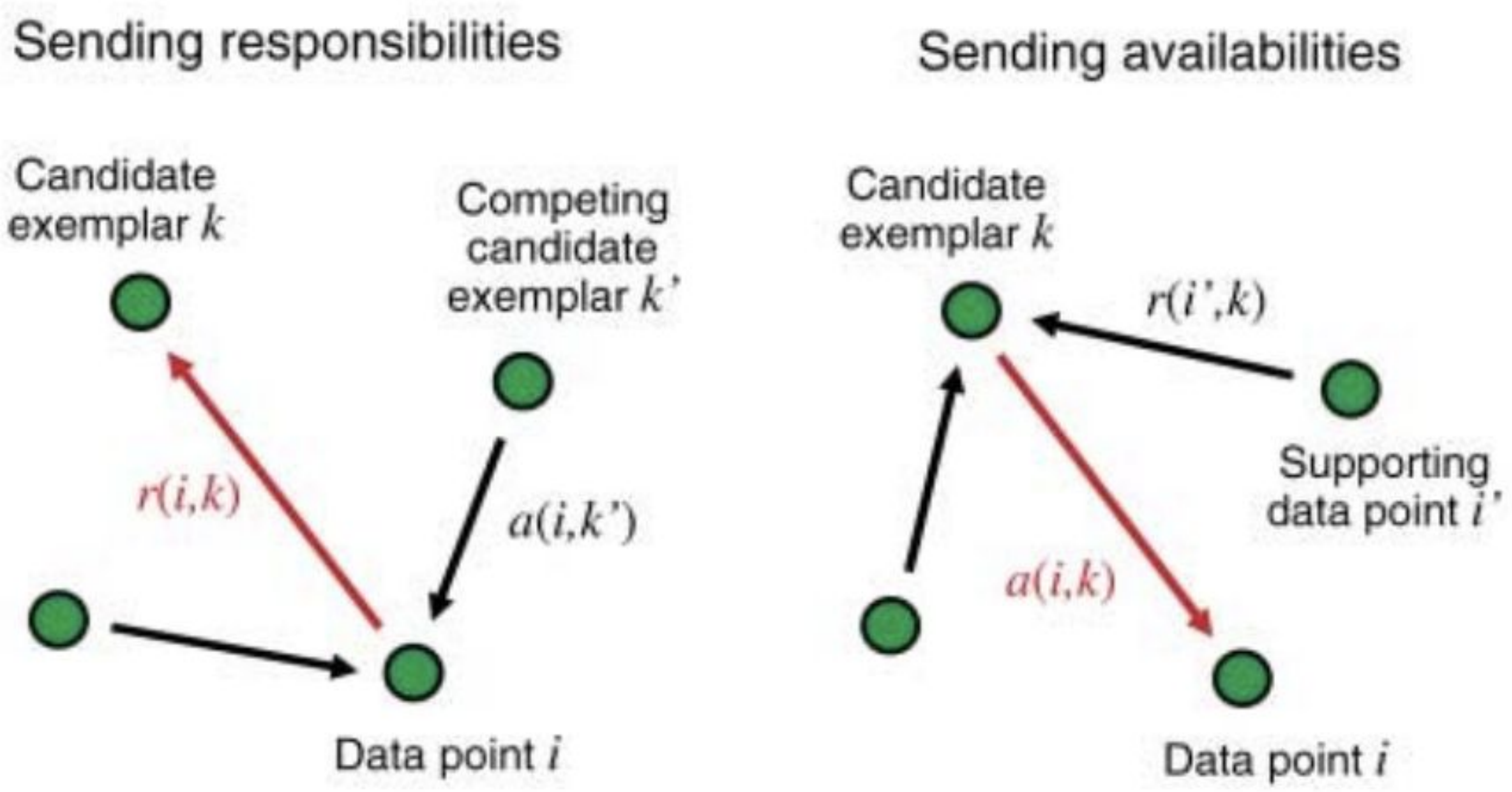
Message matrices

- **Responsibility matrix**: $R(i,k)$
  - Responsibility message from i to k
    - Accumulated evidence for how well-suited k is to serve as the exemplar for i

Message matrices

- **Responsibility matrix**: $R(i,k)$
  - Responsibility massage from i to k
    - Accumulated evidence for how well-suited k is to serve as the exemplar for i
- **Availability matrix**: $A(i,k)$
  - Availability massage from k to i
    - Accumulated evidence for how appropriate it is for i to choose k

Sending responsibility and availability



Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." *science* 315.5814 (2007): 972-976.

Update rules

1

$$R(i,k) \leftarrow S(i,k) - max(A(i,t) + S(i,t))$$
$$t \neq k \; i,k \in \{1,\ldots,l\}$$

$S(k,k)$:

- Called input preferences
- Individual tendencies of samples to become exemplars
- Important to determine number of clusters
    - Higher values results in more clusters

Update rules

1

$$R(i,k) \leftarrow S(i,k) - max(A(i,t) + S(i,t))$$
$$t \neq k \; i,k \in \{1,...,l\}$$

2

$$A(i,k) \leftarrow min(0, R(k,k) + \Sigma_j max(0, R(j,k)))$$
$$i \neq k, j \neq \{i,k\},\; i,k \in \{1,...,l\}$$

Update rules

1

$$R(i,k) \leftarrow S(i,k) - max(A(i,t) + S(i,t))$$
$$t \neq k \; i,k \in \{1,\dots,l\}$$

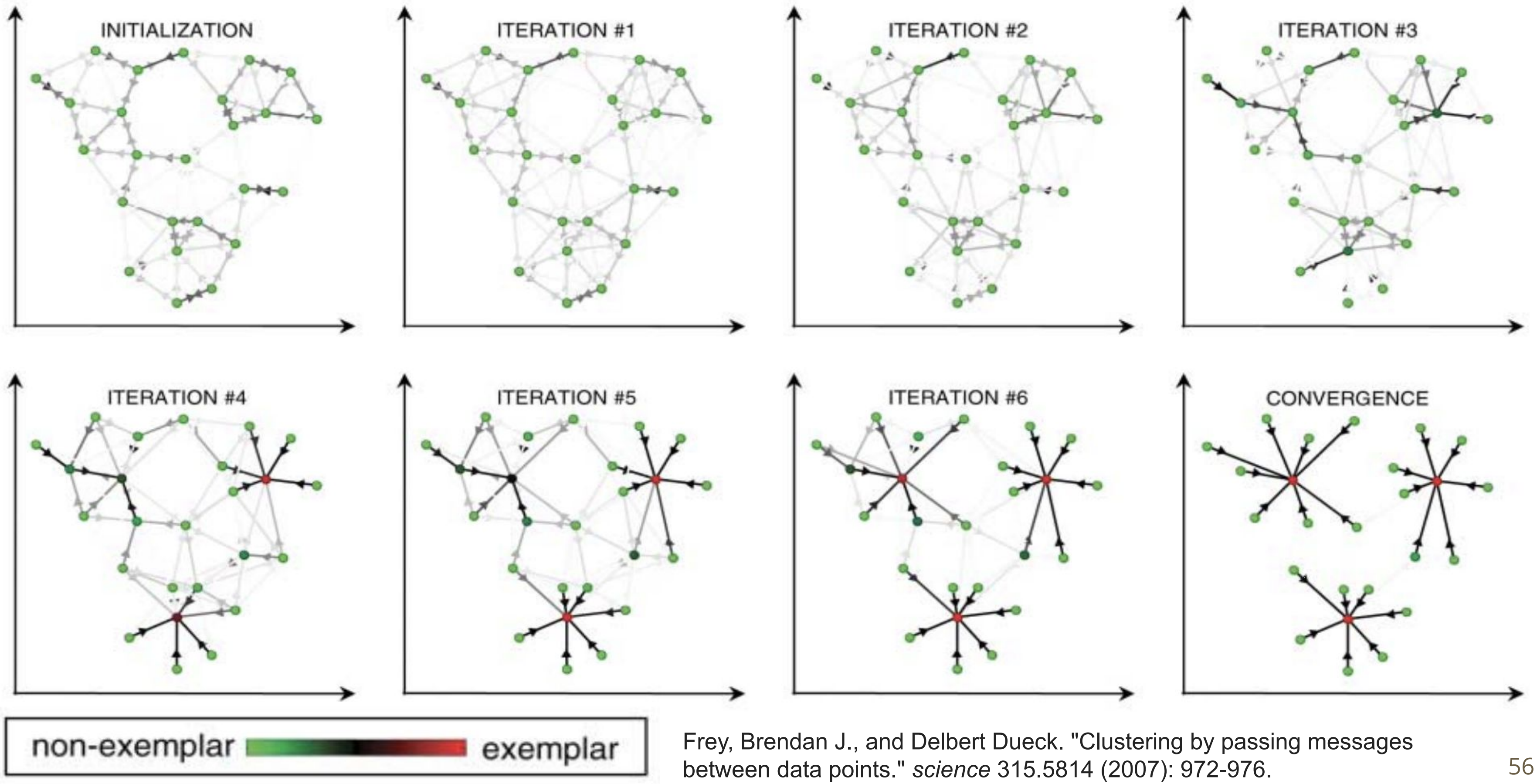
2

$$A(i,k) \leftarrow min(0, R(k,k) + \Sigma_j max(0, R(j,k)))$$
$$i \neq k, j \neq \{i,k\}, \; i,k \in \{1,\dots,l\}$$

3

$$A(k,k) \leftarrow \Sigma_j max(0, R(j,k))$$
$$j \neq k, \; k \in \{1,\dots,l\}$$

Iterations of affinity propagation for 2 dimensional data points



Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." *science* 315.5814 (2007): 972-976.

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

Let's look at clusters from another point of view:

- Clusters is a maximal set of density-connected points

Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd*. Vol. 96. No. 34. 1996.

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

Let's look at clusters from another point of view:

- Clusters is a maximal set of density-connected points

Two main parameters:

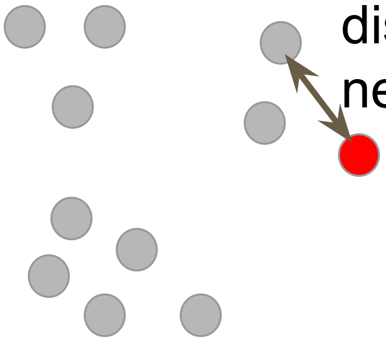
- **Epsilon (Eps)**: Maximum radius of neighbourhood
- **Minimum number of points (MinPts):** Minimum number of points in the Eps-neighbourhood of a data point

Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd*. Vol. 96. No. 34. 1996.

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

Two main parameters:

- **Epsilon (Eps)**: Maximum radius of neighbourhood
- **Minimum number of points (MinPts):** Minimum number of points in the Eps-neighbourhood of a data point



dist<Eps => this point is in the neighbourhood of data point shown in red

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

Two main parameters:

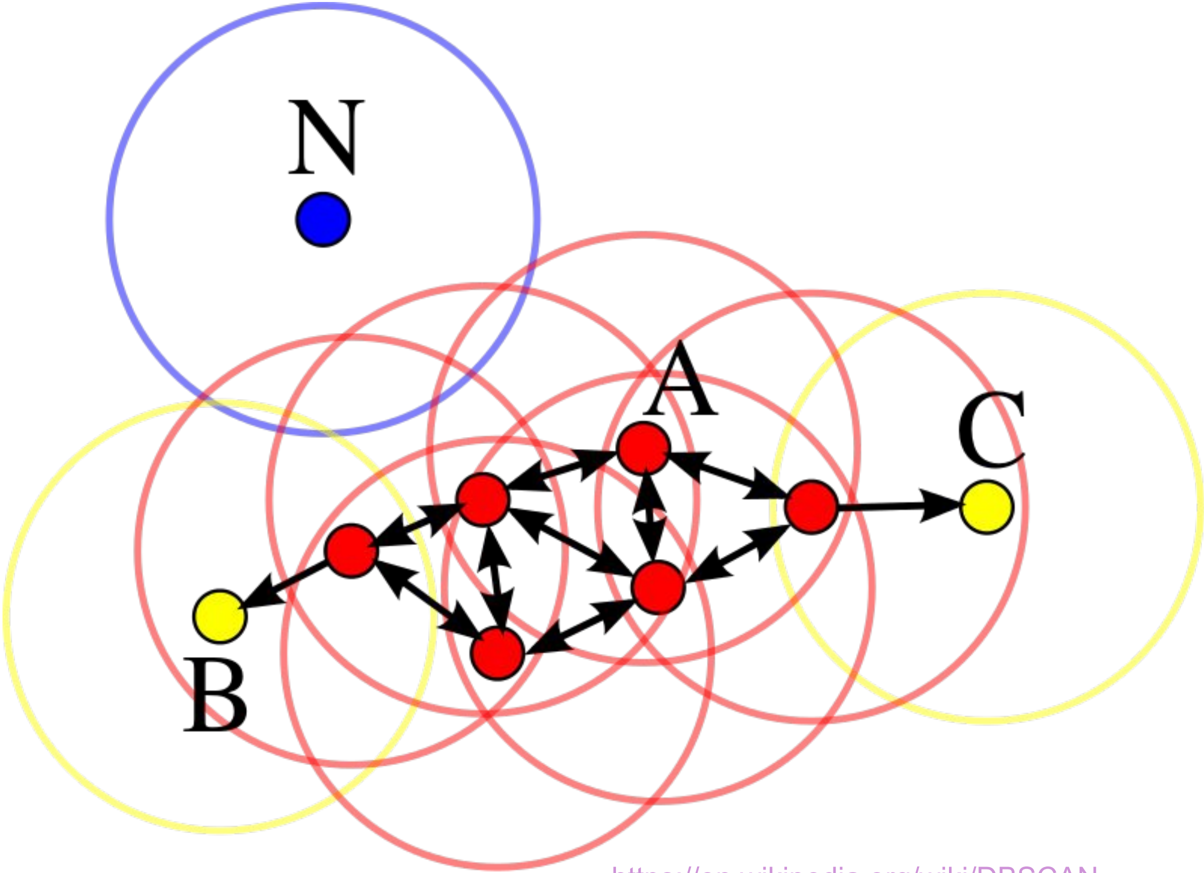
- **Epsilon (Eps)**: Maximum radius of neighbourhood
- **Minimum number of points (MinPts):** Minimum number of points in the Eps-neighbourhood of a data point
    - Then that data point will be called a **core** point

Different types of data points:

- **Core**
- **Border**: within epsilon distance does not meet MinPts criteria
    - But there is at least 1 core point within the epsilon distance
- **Noise (Outlier)**: Not assigned to any clusters

Visualizing different types of data points in DBSCAN

A: Core
B and **C**: Border
N: outlier



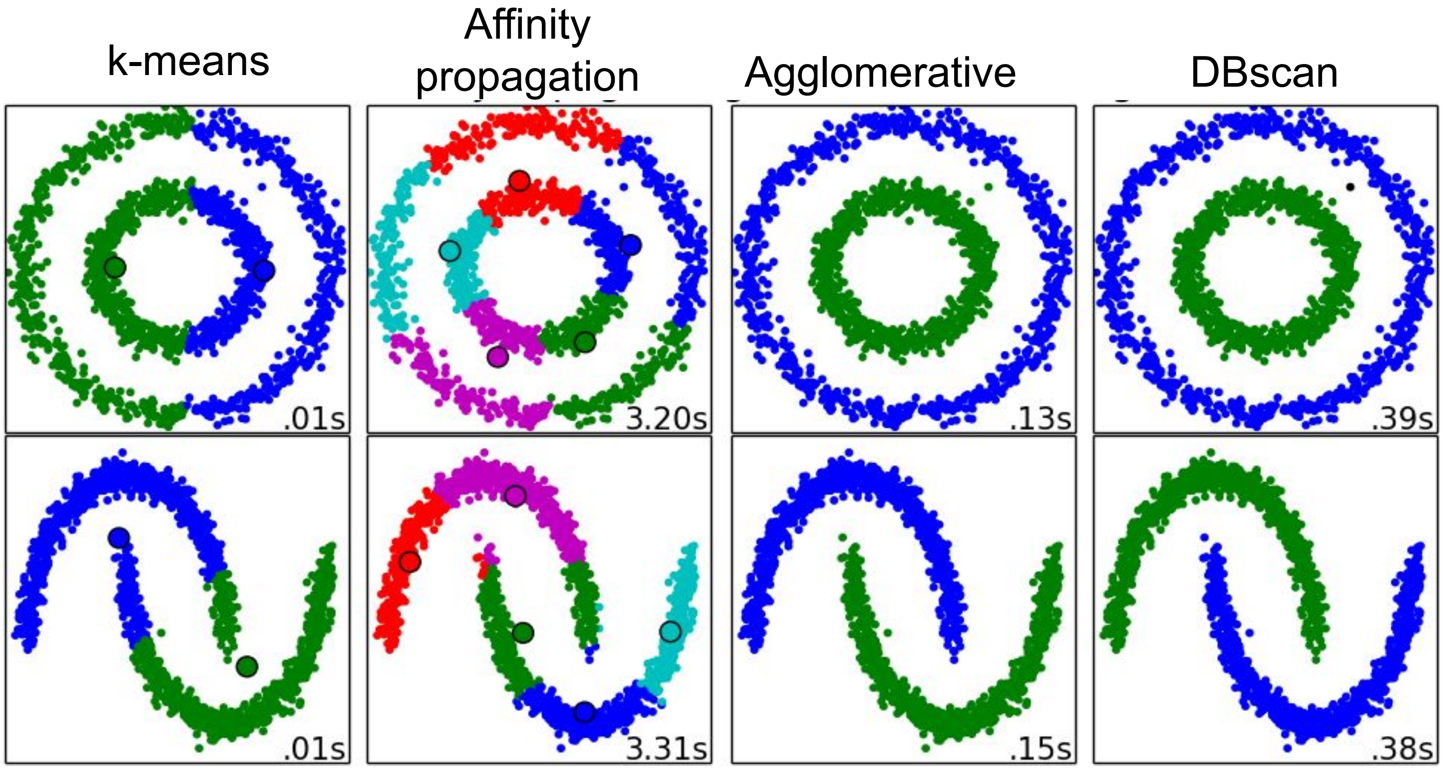
https://en.wikipedia.org/wiki/DBSCAN

Some disadvantages of DBSCAN

- May not work well with clusters of similar densities
  - But great for separating clusters of low and high densities

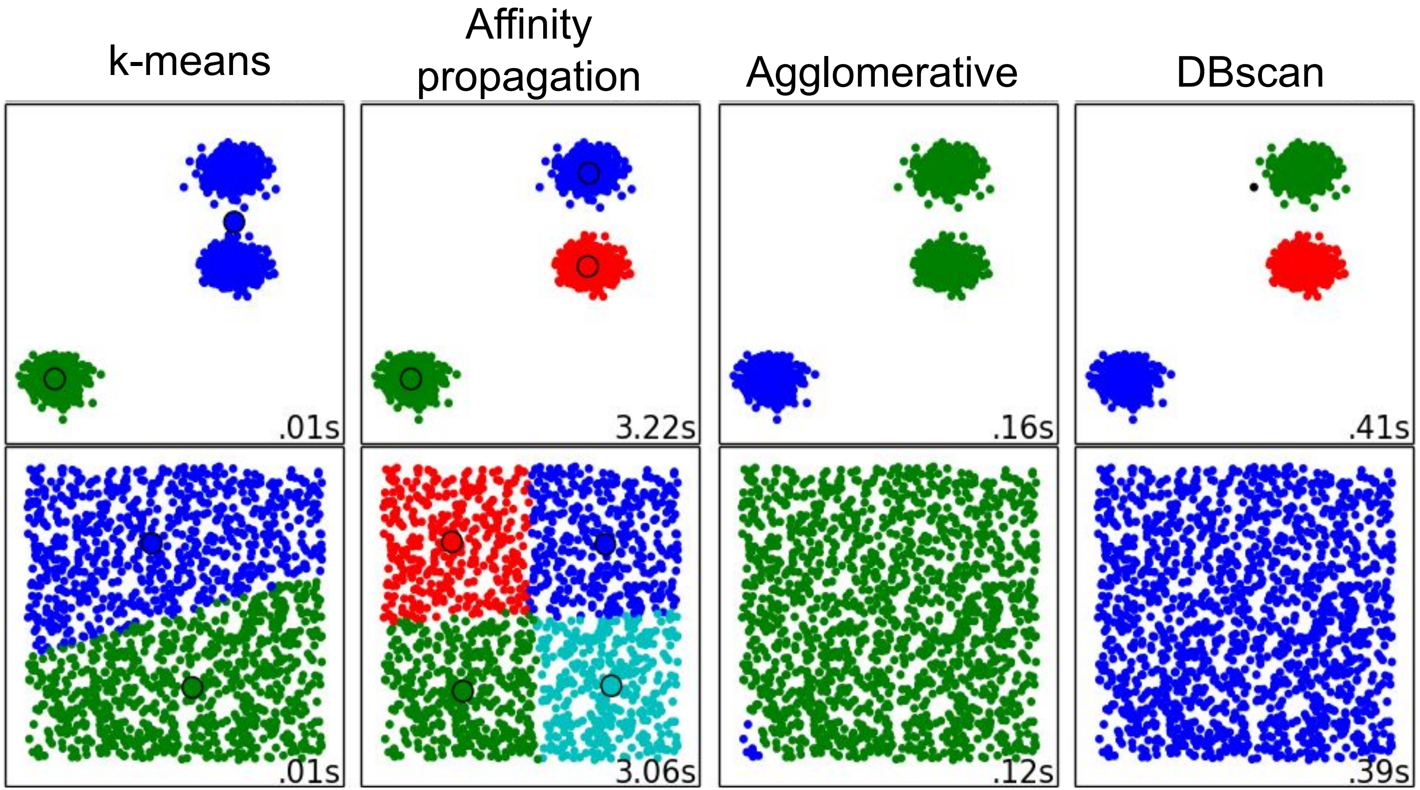
- May not be great with very high dimensional data

Comparison of clustering methods



https://scikit-learn.org/0.15/auto_examples/cluster/

Comparison of clustering methods



https://scikit-learn.org/0.15/auto_examples/cluster/