GL Applied Data Science Program

Unsupervised Learning - Clustering

Overview

Overview of this week / module:

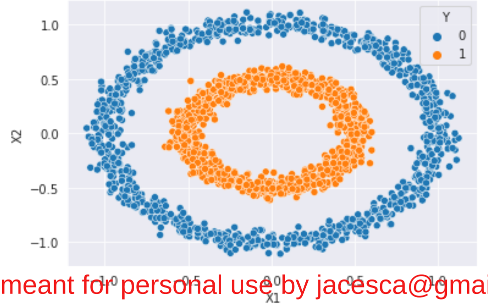
- Data collection and visualization for exploratory data analysis
- Network analysis
- Unsupervised learning - clustering

Overview of this lecture:

- Clustering methods
- Community detection in networks

Case study: clustering

- Find groups, so that elements within cluster are very similar and elements between clusters are very different
- **Examples:**
  - Find customer groups to adjust advertisement
  - Find subtypes of diseases to fine-tune treatment
- Our eye is very good at identifying cluster



Clustering

Clustering

N samples, k clusters: k^N possible assignments

- E.g., $N = 100$, $k = 3$: $3^{100} = 5 * 10^{47}$!!

  $\Rightarrow$ impossible to search through all assignments

We will discuss:

- k-means clustering
- Gaussian mixture models
- Hierarchical clustering
- DBSCAN

K-means clustering

- Choose the number of clusters K
- Minimize the sum of the pairwise distances between samples within each cluster (also known as the *Within-Groups Sum of Squares*)

K-means clustering

- Choose the number of clusters K
- Minimize the sum of the pairwise distances between samples within each cluster (also known as the *Within-Groups Sum of Squares*)
  - One can show that this is equivalent to minimizing the sum of the distances to the cluster means
- Exact solution of minimization problem is computationally infeasible
  - Use greedy algorithm with random restarts to avoid local optima
- Leads to spherical shaped clusters of similar radii

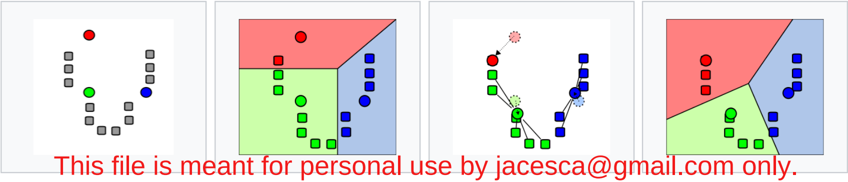
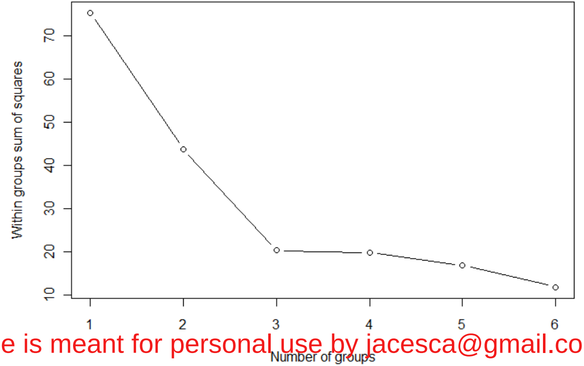


Image source: Wikipedia

Choosing the number of clusters

- Run K-means clustering for several number of groups K
- Plot Within-Groups Sum of Squares versus the number of groups
- Choose number of groups after the last big drop of the curve

Example:



Partitioning around medoids (PAM)

- K-Means: Cluster centers can be arbitrary points in space

  $\Rightarrow$ very sensitive to outliers!

Partitioning around medoids (PAM)

- K-Means: Cluster centers can be arbitrary points in space

  $\Rightarrow$ very sensitive to outliers!

- Robust alternative: Partitioning around medoids (PAM)
  - Cluster center must be an observation ("medoid")
  - More robust against outliers
  - Also gives a representative object for each cluster (e.g., for easy interpretation)

Hierarchical: ingle linkage.

DBSCAN

Gaussian mixture model

- Soft version of K-means clustering, where each sample is attributed to a cluster with a certain probability
  - This allows for points between clusters to belong to multiple clusters

- Distribution of samples within each cluster is modeled by a Gaussian
  - This allows for ellipsoidal shaped clusters and the number of clusters can be determined in a statistically sound way (for example using the so-called *Bayesian Information Criterion*)

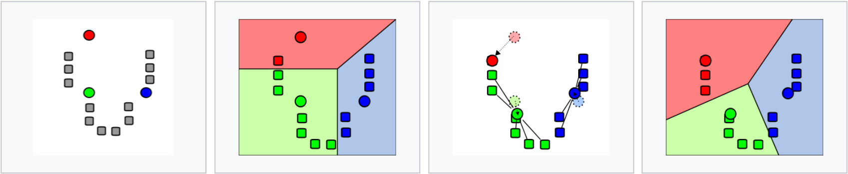
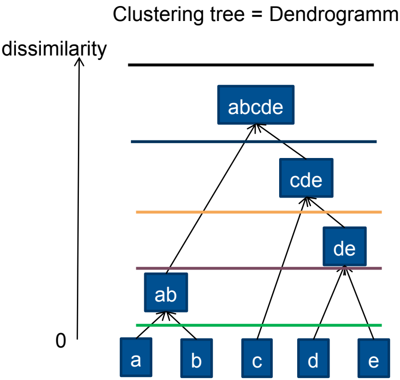


Image source: Wikipedia

Hierarchical clustering

- Agglomerative clustering: Build up clusters from individual observations
- Divisive clustering: Start with whole group of observations and split off clusters



Advantage of hierarchical clustering:

- Solve clustering for all possible numbers of cluster $1, 2, \ldots, n$ at once
- Choose desired number of clusters later

Distance measures between clusters

How we define distances between clusters can have a huge effect on what kinds of clusters we obtain!

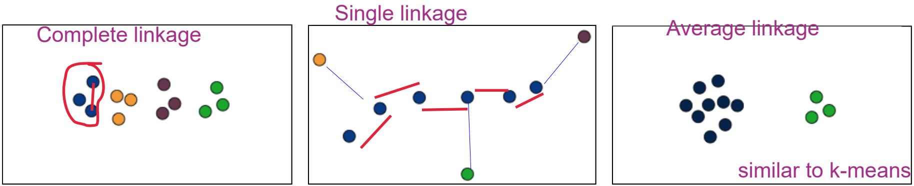
- single linkage (i.e., minimum distance)
- complete linkage (i.e., maximum distance)
- average linkage (i.e., average distance)

Distance measures between clusters

How we define distances between clusters can have a huge effect on what kinds of clusters we obtain!

- single linkage (i.e., minimum distance) Good at identifying long elongated clusters.
- complete linkage (i.e., maximum distance) Good at identifying small and badly separated clusters
- average linkage (i.e., average distance) Good at identifying large, well separated clusters.

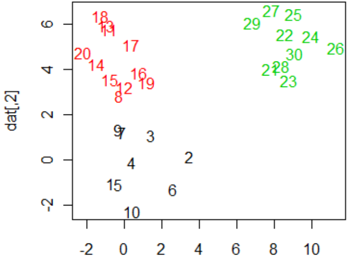
Which clustering output belongs to which choice of linkage?

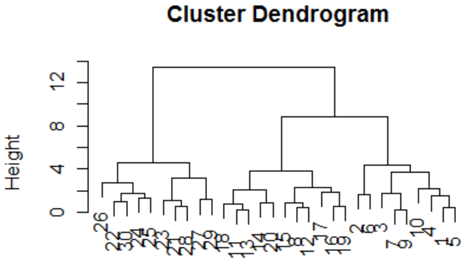


Choosing the number of clusters

- No strict rule
- Find the largest vertical “drop” in the tree

Example:





DBSCAN

- Uses 2 parameters: minPts (minimum number of points) and ϵ (radius of neighborhood)
- Core points have at least minPts within distance ϵ
- Clusters are defined by looking at all points reachable from a core point

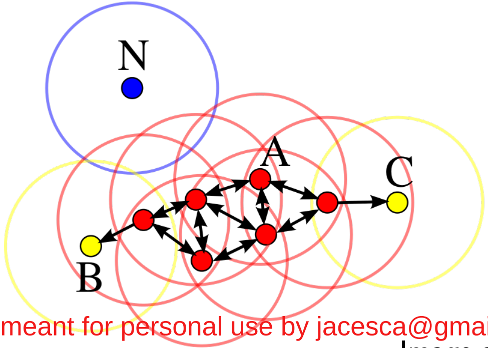


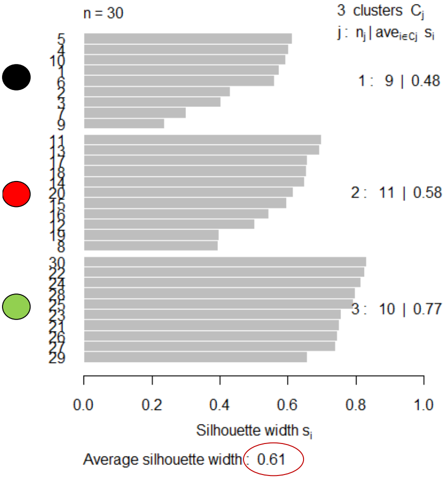
Image source: Wikipedia

Quality of clustering: Silhouette plot

Compute for each sample $x^{(i)}$:

- $a(x^{(i)})$ = average distance between $x^{(i)}$ and all other points in its cluster
- $b(x^{(i)})$ = average distance between $x^{(i)}$ and the closest cluster it does not belong to
- $S(x^{(i)}) \in [-1, 1]$ with

$$S(x^{(i)}) = \frac{(b(x^{(i)}) - a(x^{(i)}))}{\max(a(x^{(i)}), b(x^{(i)}))}$$

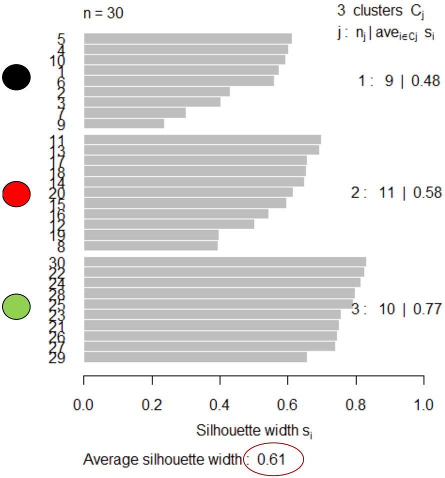


Quality of clustering: Silhouette plot

Compute for each sample $x^{(i)}$:

- $a(x^{(i)})$ = average distance between $x^{(i)}$ and all other points in its cluster
- $b(x^{(i)})$ = average distance between $x^{(i)}$ and the closest cluster it does not belong to
- $S(x^{(i)}) \in [-1, 1]$ with

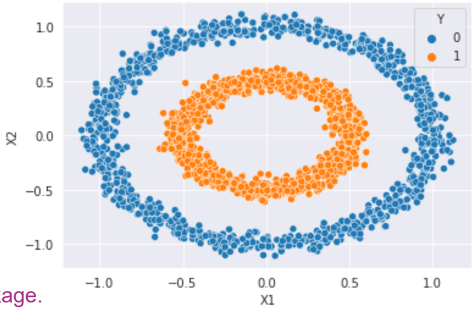
$$S(x^{(i)}) = \frac{(b(x^{(i)}) - a(x^{(i)}))}{\max(a(x^{(i)}), b(x^{(i)}))}$$



Note: $S(x^{(i)})$ large (0.5 is often used as cut-off): well clustered; $S(x^{(i)})$ small: badly clustered; $S(x^{(i)}) < 0$: assigned to wrong cluster

Case study: clustering

Which clustering methods are able to identify the two clusters?

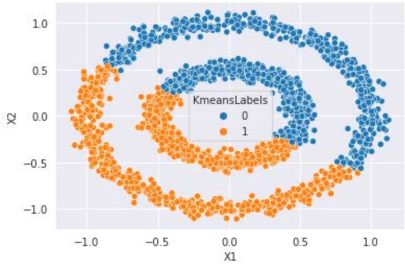


Hierarchical: single linkage.

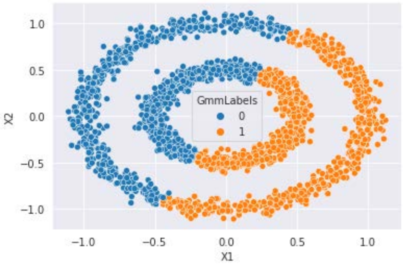
DBSCAN

Case study: clustering

k-means clustering



Gaussian mixture model



Hierarchical clustering (single linkage)

DBSCAN

Community detection

Community detection:

- detect subsets of nodes that are more densely connected between each other in the network than outside the community

Clustering

- determine subsets of points that are 'close' to each other given a pairwise distance or similarity measure defined by the network
- examples for vertex similarity measures: hop distance, number of different neighbors, correlation between adjacency matrix columns,...
- can use clustering methods discussed so far based on these similarity measures to identify communities in a network

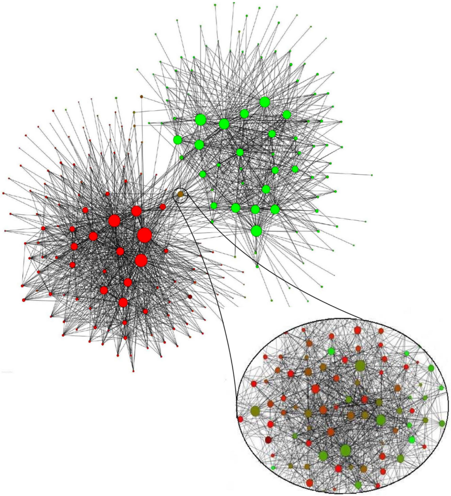
Other methods: Divisive and agglomerative algorithms

- **Algorithm of Girvan and Newman** (2002): iteratively remove edges with highest betweenness centrality
  - Intuition: intercommunity edges have a large value of edge betweenness, because many shortest paths connecting vertices of different communities will pass through them

Other methods: Divisive and agglomerative algorithms

- **Algorithm of Girvan and Newman** (2002): iteratively remove edges with highest betweenness centrality
  - Intuition: intercommunity edges have a large value of edge betweenness, because many shortest paths connecting vertices of different communities will pass through them
- **Louvain method** by Blondel et al. (2008): iteratively merge pairs of clusters that are connected by more edges than expected if the edges were randomly distributed (also known as modularity score)
  - Is extremely fast and provides decomposition of network into communities for different levels of of organization

Louvain method



Belgian mobile phone network with 2M customers (red: French-speaking, green: Dutch-speaking).

References

- For clustering
  - Chapter 14 in

    T. Hastie, R. Tibshirani, & J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

- For community detection in networks:
  - V. D. Blondel, et al. *Fast unfolding of communities in large networks*. Journal of Statistical Mechanics: Theory and Experiment 10, 2008.
  - S. Fortunato. *Community detection in graphs*. Physics Reports 486, 2010.
  - Lecture notes on Laplacian and spectral clustering (prominent method not discussed in this module) by T. Roughgarden & G. Valiant:

    http://web.stanford.edu/class/cs168/l/l11.pdf