Clustering and Community Detection

Mikhail Belyaev and Maxim Panov

Skoltech
Skolkovo Institute of Science and Technology

2020

Overview

- Clustering:
  - Agglomerative clustering
  - K-means
  - Mixture models
- Community detection
  - Spectral clustering
  - Modularity based clustering

Clustering Problem: Documents

Finding topics:

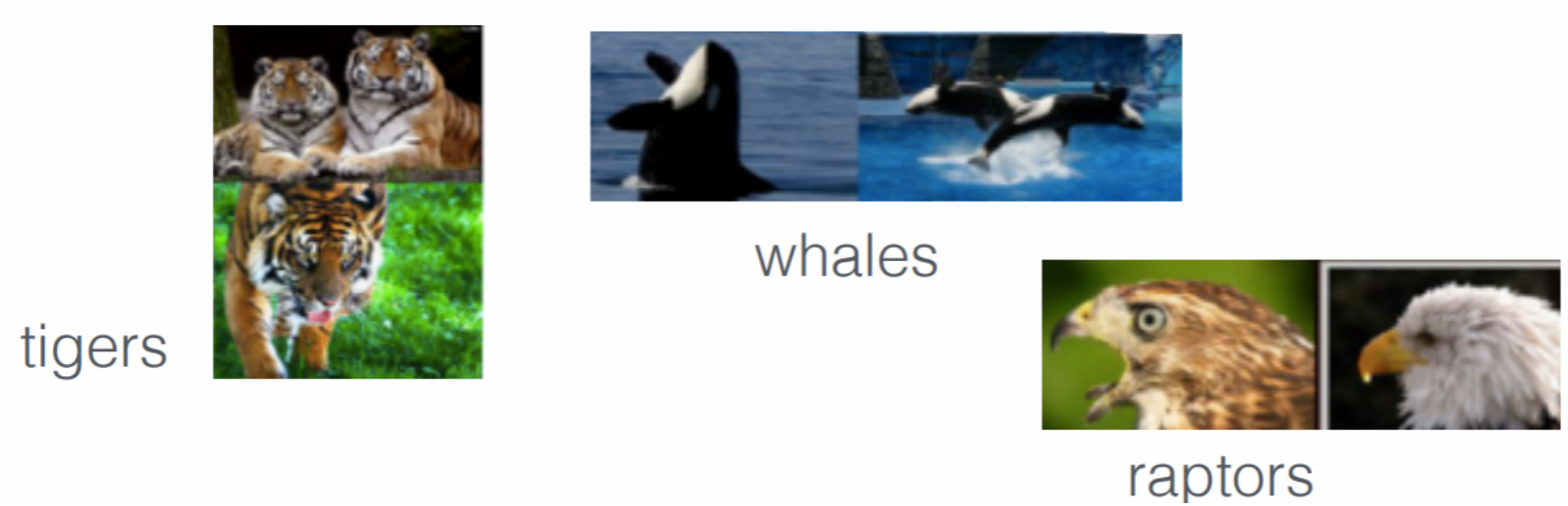
- Texts are everywhere and majority of them are unlabeled $\rightarrow$ unsupervised methods are needed for the analysis.
- Represent a document by a vector $(x_1, x_2, \ldots, x_k)$, where $x_i = 1$ iff the i-th word (in some order) appears in the document.
- Documents with similar sets of words may be about the same topic.

Clustering Problem: Images

What is a cluster?

Goal: partitioning data in maximally homogeneous, maximally distinguished subsets.

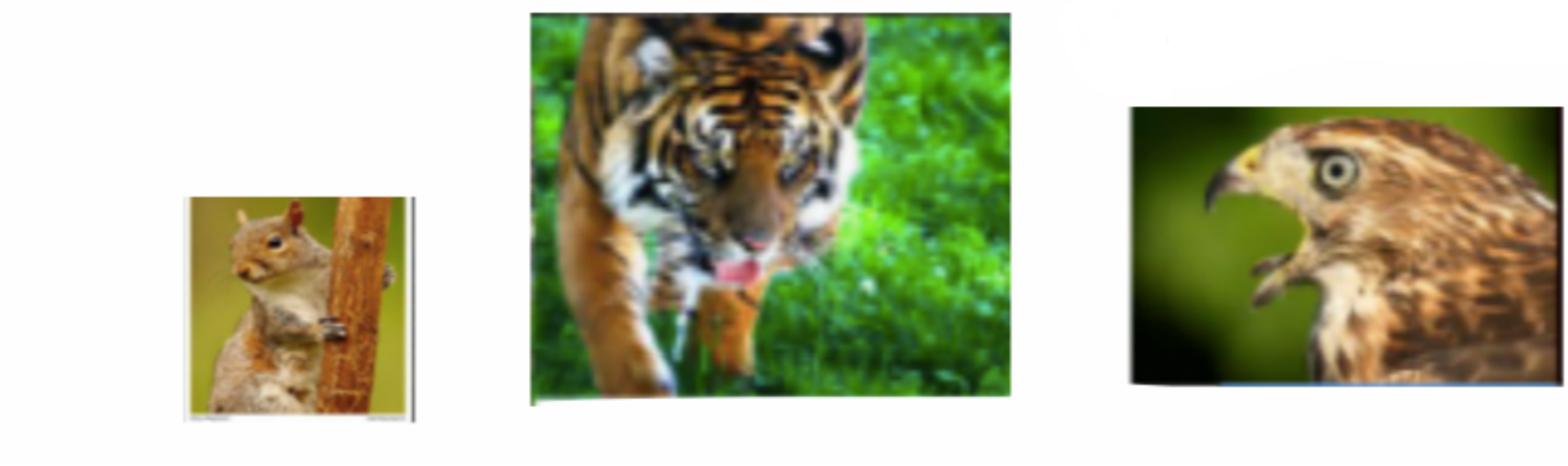
- **Internal criterion:** members of the cluster should be similar to each other (**inter-cluster compactness**).



What is a cluster?

Goal: partitioning data in maximally homogeneous, maximally distinguished subsets.

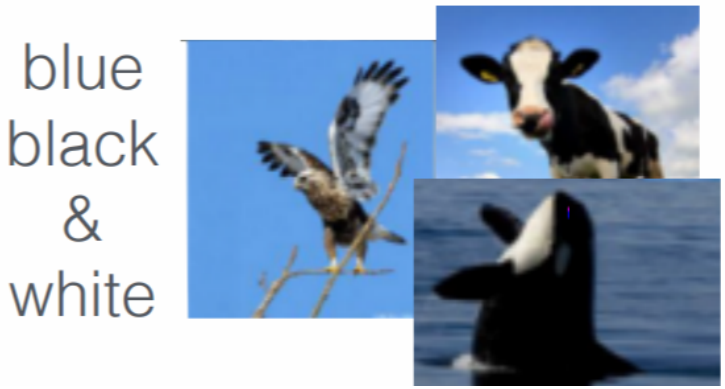
- **External criterion**: objects outside the cluster should be dissimilar from the objects inside the cluster (**intra-cluster distance**).



What is a cluster?

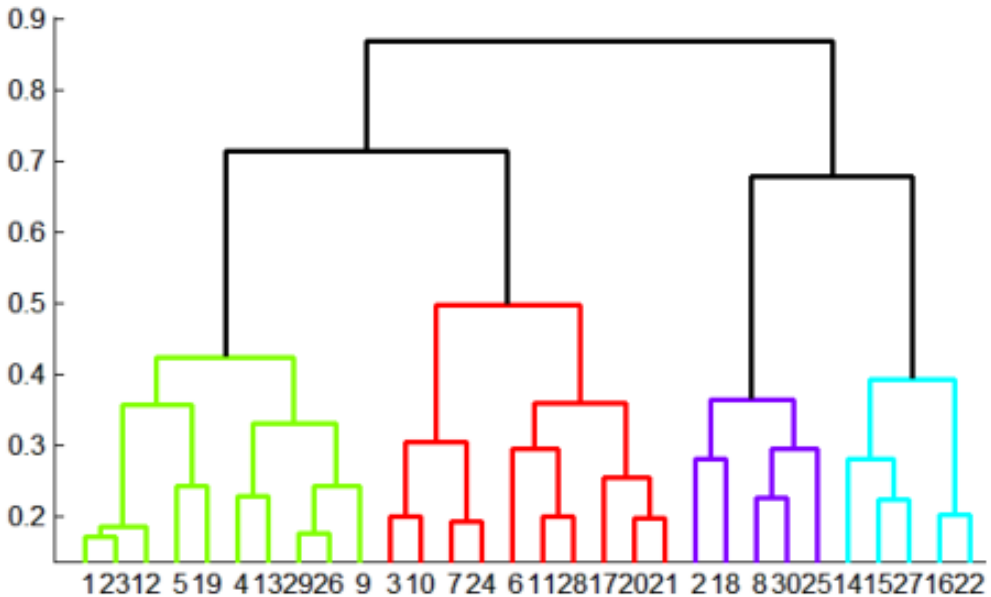
Goal: partitioning data in maximally homogeneous, maximally distinguished subsets.

- **Internal criterion:** members of the cluster should be similar to each other (**inter-cluster compactness**).
- **External criterion**: objects outside the cluster should be dissimilar from the objects inside the cluster (**intra-cluster distance**).



Hierarchical clustering

- **Agglomerative (bottom up)**:
  - Initially, each point is a cluster;
  - Repeatedly combine the two "nearest" clusters into one.
- **Divisive (top down)**:
  - Start with one cluster and recursively split it.

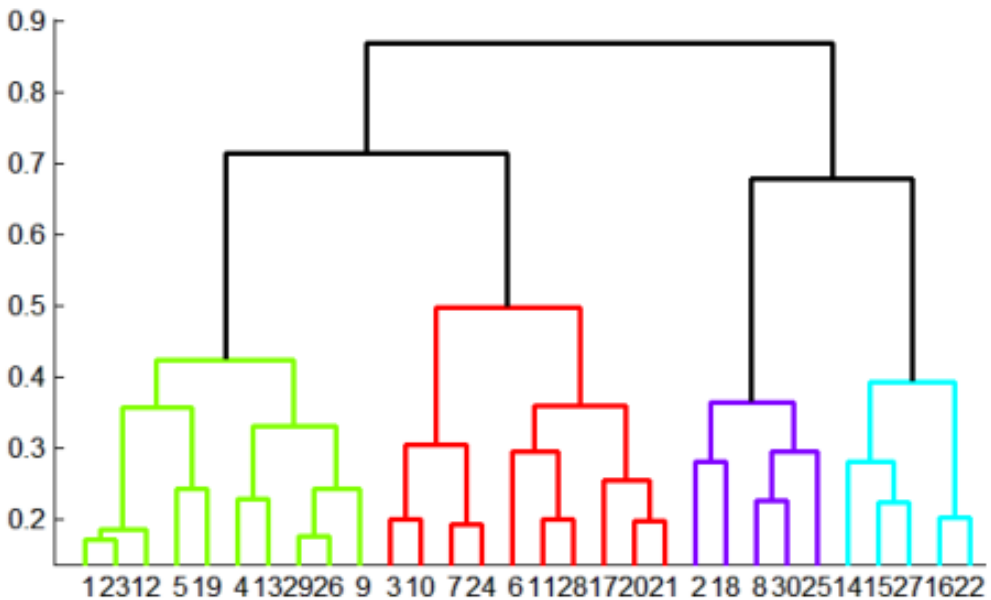


Hierarchical clustering

Key operation: Repeatedly combine two nearest clusters.

Three important questions:

1. How do you represent a cluster of more than one point?
2. How do you determine the "nearness" of clusters?
3. When to stop combining clusters?



Hierarchical clustering

Key operation: Repeatedly combine two nearest clusters.

1. How do you represent a cluster of more than one point?
   - **Key problem:** As you merge clusters, how do you represent the "location" of each cluster, to tell which pair of clusters is closest?
   - **Euclidean case**: each cluster has a centroid = average of its points.
2. How do you determine the "nearness" of clusters?
   - Measure cluster distances by distances of centroids.

What distance to pick?

Minkowski family of distances:

$$D(x, y) = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots + |x_n - y_n|^p}.$$

It can be checked that for any $p \geq 1$:

- $D(x, y) \geq 0$,
- $D(x, x) = 0$,
- $D(x, y) = D(y, x)$,
- $D(x, y) \leq D(x, z) + D(z, y)$.

Manhattan distanse

Minkowski family of distances:

$$D(x, y) = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots + |x_n - y_n|^p}.$$

In case of $p = 1$:

$$D(x, y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|.$$

It is nicknamed **Manhattan distance** (blue):

Euclidean distanse

Minkowski family of distances:

$$D(x,y) = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots + |x_n - y_n|^p}.$$

In case of $p = 2$:

$$\sqrt[2]{|x_1 - y_1|^2 + |x_2 - y_2|^2 + \cdots + |x_n - y_n|^2}.$$

Euclidean distance (green):

K-means

- Randomly **initialize** k centers:

$$\mu^0 = (\mu_1^0, \ldots, \mu_k^0).$$

- **Classify:** Assign each point $j \in \{1, \ldots, m\}$ to nearest center:

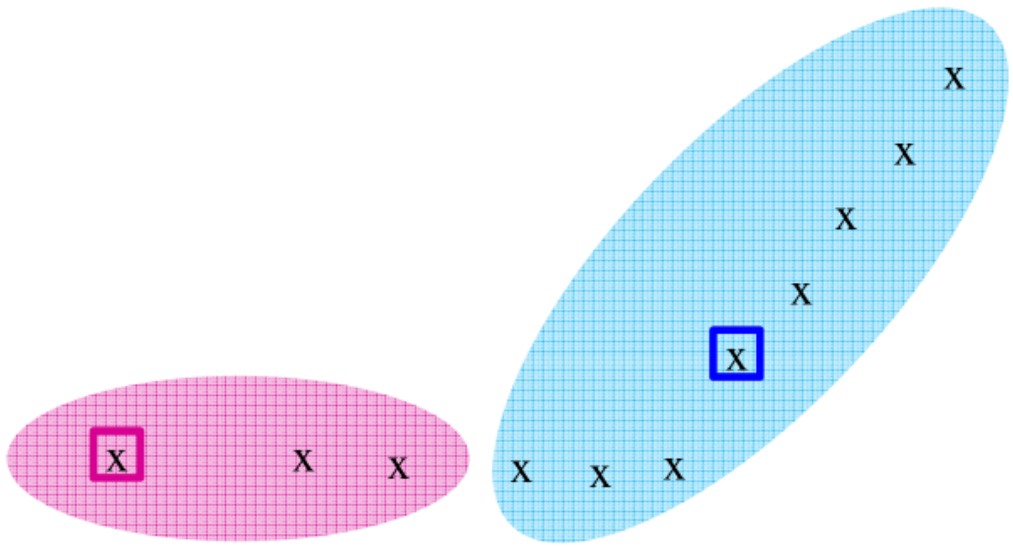
$$z^j = \arg\min_i \|x_j - \mu_i^t\|_2^2.$$

- **Recenter:** μ_i becomes centroid of its points:

$$\mu_i^{t+1} = \arg\min_\mu \sum_{j:z^j=i} \|x_j - \mu\|_2^2.$$

  - Equivalent to μ_i average of its points!

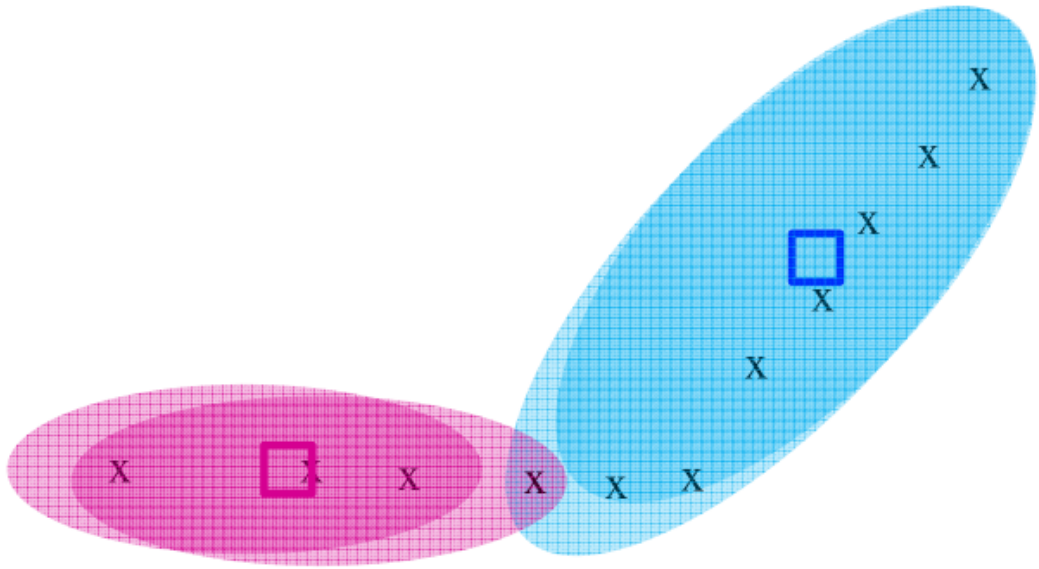
K-means



x … data point
□ … centroid

Clusters after round 1

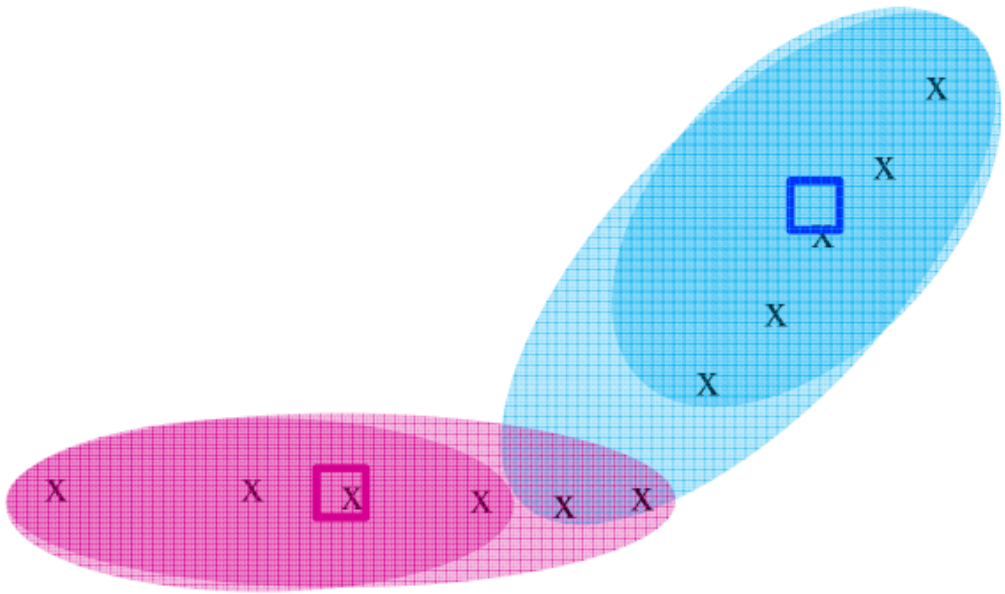
K-means



x … data point
□ … centroid

Clusters after round 2

K-means



x … data point
□ … centroid

Clusters at the end

K-means

Assumptions:

- Known number of clusters;
- Clusters of approximately same size and density;
- Spherical form.

Cluster validity

- For supervised classification we have a variety of measures to evaluate how good our model is: accuracy, precision, recall.
- For cluster analysis, the analogous question is how to evaluate the goodness of the resulting clusters?
- We need measures to compare:
  - clustering algorithms;
  - two sets of clusters.

Measures of cluster validity

- **External measure:** used to measure the extent to which cluster labels match externally supplied class labels.
  - Example: Jaccard index.
- **Internal measure:** used to measure the goodness of a clustering structure without respect to external information.
  - Example: Sum of Squared Errors (SSE).

External measures: Jaccard Index

$$J = \frac{a}{a + b + c},$$

where

- a is the number of pairs of points with the same label in C and assigned to the same cluster in P;
- b is the number of pairs with the same label, but in different clusters;
- c is the number of pairs in the same cluster, but with different class labels.

The index produces a result in the range $[0, 1]$, where a value of 1.0 indicates that C and P are identical.

External measures: Rand Index

$$RI = \frac{a + d}{a + b + c + d},$$

where

- a, b, c are as above
- d denotes the number of pairs with a different label in C that were assigned to a different cluster in P
- The index produces a result in the range $[0, 1]$, where a value of 1.0 indicates that C and P are identical.
- A high value for this measure generally indicates a high level of agreement between a clustering and the true classes.

Internal measures: Silhouette Coefficient

Consider an i-th individual point

- $a(i)$ = average distance of the i-th point to the points in its cluster.
- $b(i)$ = min (average) distance of the i-th point to points in other clusters.
- The silhouette coefficient for the point is then given by

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

- Property: $-1 \leq s(i) \leq 1$.

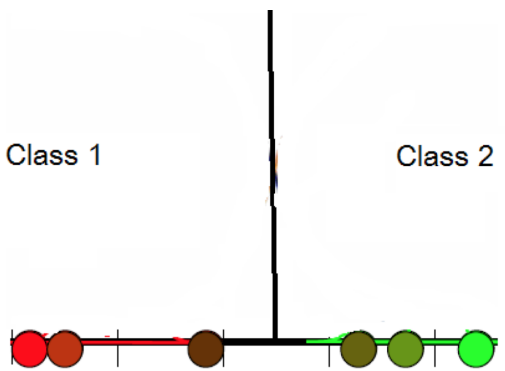
Internal Measure: Silhouette Coefficient

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- If $s(i)$ is close to 1, sample i is well-clustered and it was assigned to a very appropriate cluster.
- If $s(i)$ is close to zero, sample i could be assign to another closest cluster as well, and the sample lies equally far away from both clusters.
- If $s(i)$ is close to -1, sample i is misclassified.
- We can consider average $\frac{\sum_i s(i)}{m}$ of $s(i)$ for all objects in the whole dataset:
  - the larger it is, the better the clustering is.

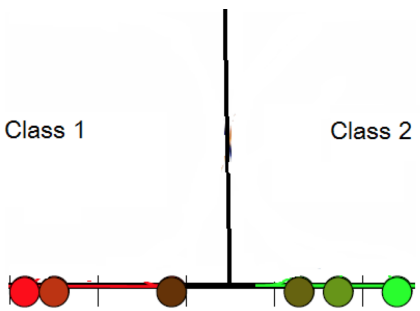
K-means and Hard Assignment

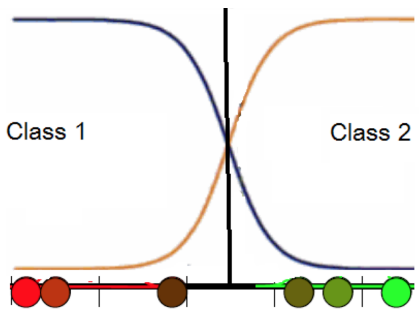
- K-means performs hard assignment.
- Each data point is assigned to one and only one cluster.
- Appropriate for points close to cluster centroids.
- Not appropriate for points midway between the two cluster centroids.



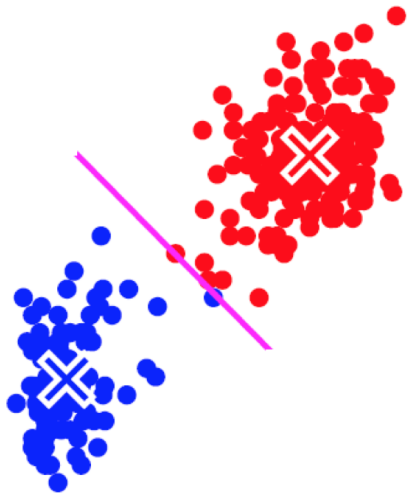
Probabilistic (Soft) Assignment

- Each data point is (partially) assigned to clusters with certain probabilities.
- One point might be (partially) assigned to multiple clusters.
- The midway point is assigned to either cluster with probability 0.5.

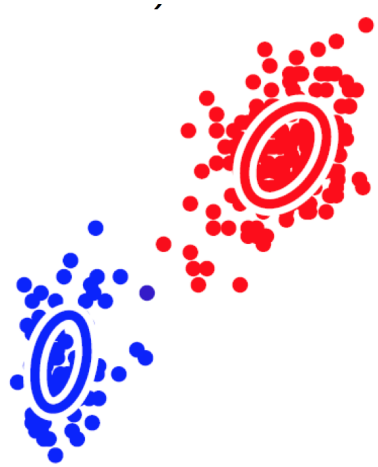




Hard/Soft Assignment: 2D example

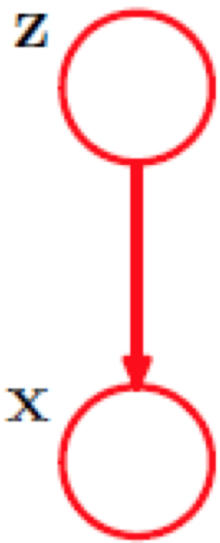


Hard Assignment



Soft Assignment
(Ellipses: contour of probability functions).

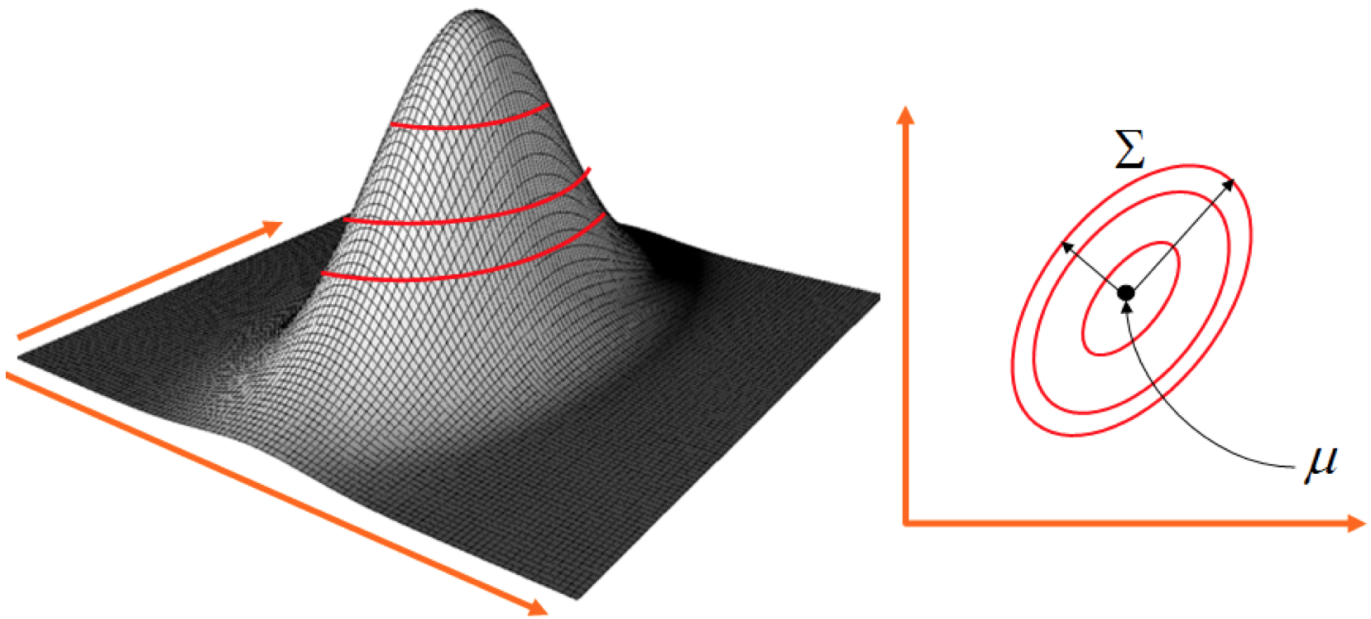
Mixture Models



- K clusters: $1, 2, \ldots, K$.
- Randomly drawn object
  - x: attribute values of the object, observed.
  - z: class of the object, latent variable (not observed).
- $P(z)$: distribution of z
  - $\pi_k = P(z = k)$: probability that the object is from class k.
- $p(\mathrm{x}|z)$: conditional distribution of attribute values
  - $p(\mathrm{x}|z = k)$: distribution for objects from class k.

Multivariate Gaussian Distribution

- $\boldsymbol{\mu}$: center of contour lines
- $\boldsymbol{\Sigma}$: orientation and size of contour lines



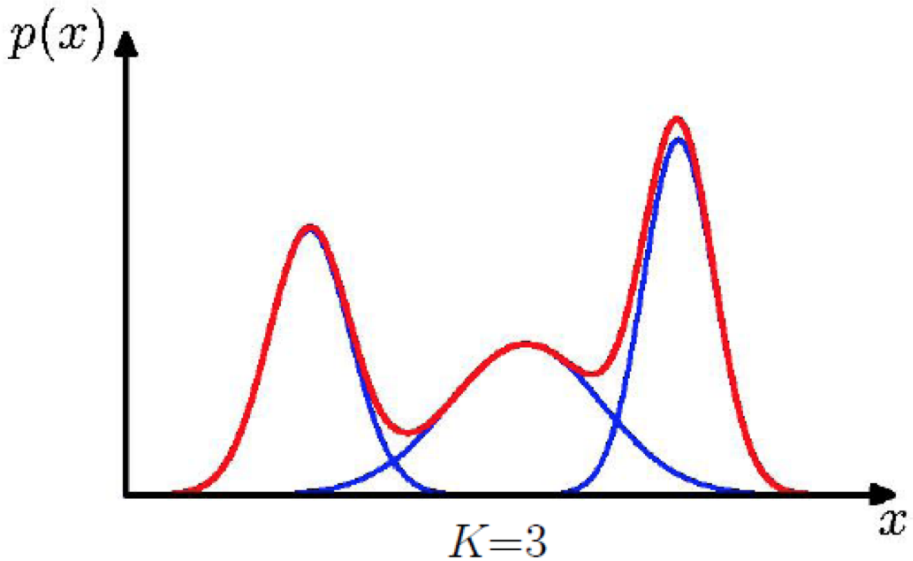
Gaussian Mixture Models

- Mixture distribution:

$$p(\mathrm{x}) = \sum_{k=1}^{K} \pi_k p(\mathrm{x}|z = k).$$

- Each component is a Gaussian distribution:

$$p(\mathrm{x}|z = k) = \mathcal{N}(\mathrm{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$



Problem Statement

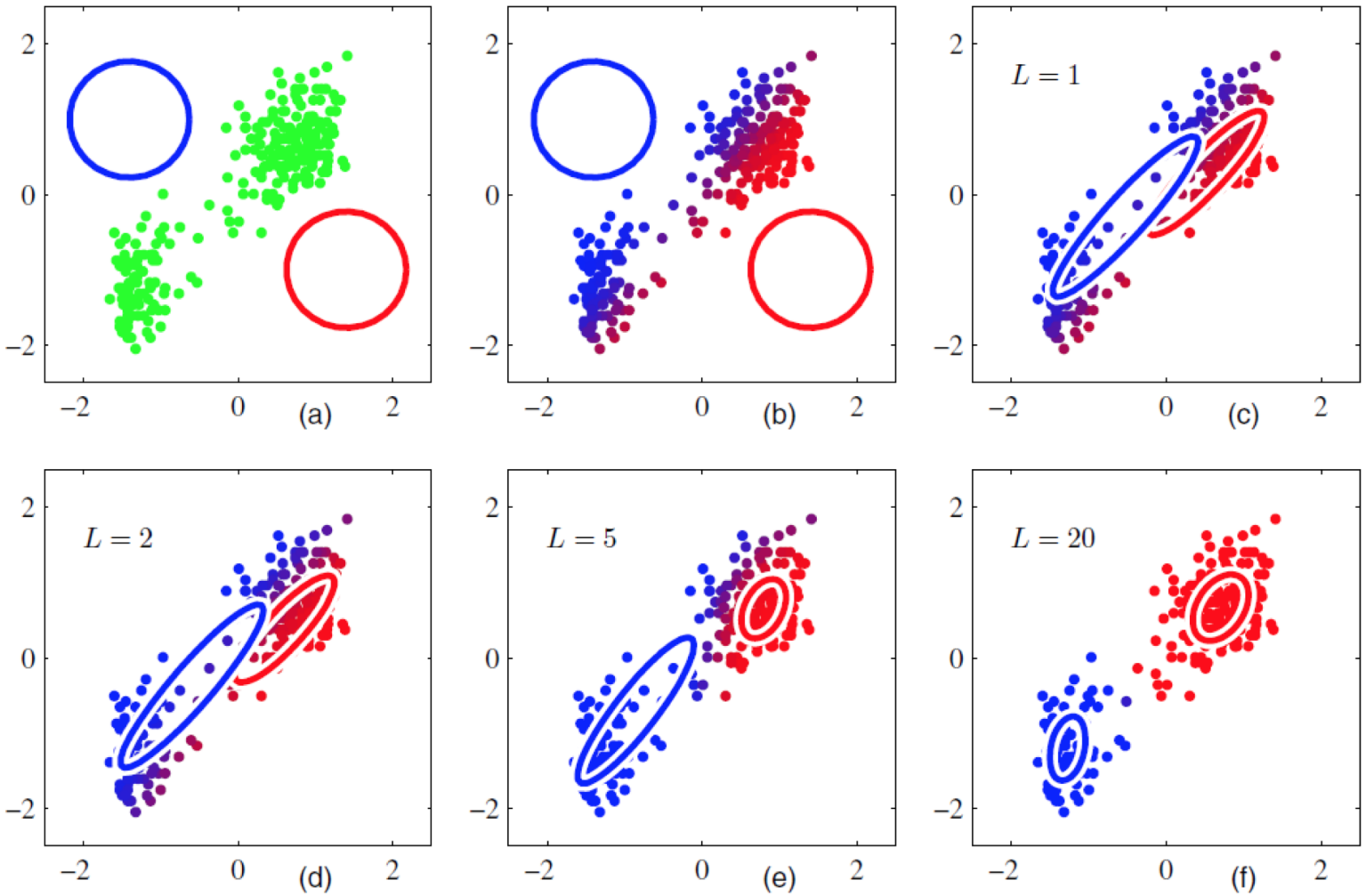
- Given
  - unlabeled data $\{x_1, \ldots, x_m\}$;
  - number of clusters K.

- Find a K-component Gaussian mixture model:
  - mixing coefficients $\{\pi_1, \ldots, \pi_K\}$;
  - components parameters $\{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$

  by maximizing data likelihood.

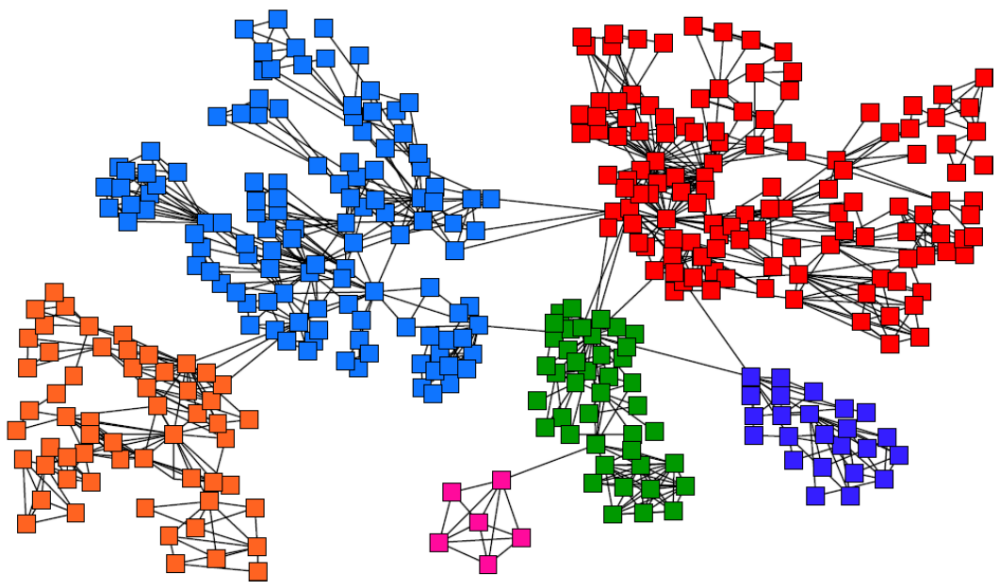
Solution: Expectation-Maximization algorithm (EM).

EM: 2D illustration



Community detection

- **Graph** $G(E, V)$:
  - Nodes v_j
  - Edge weights $w_{ij} > 0$.
- **Problem**: Want to partition graph such that edges between groups have low weights.

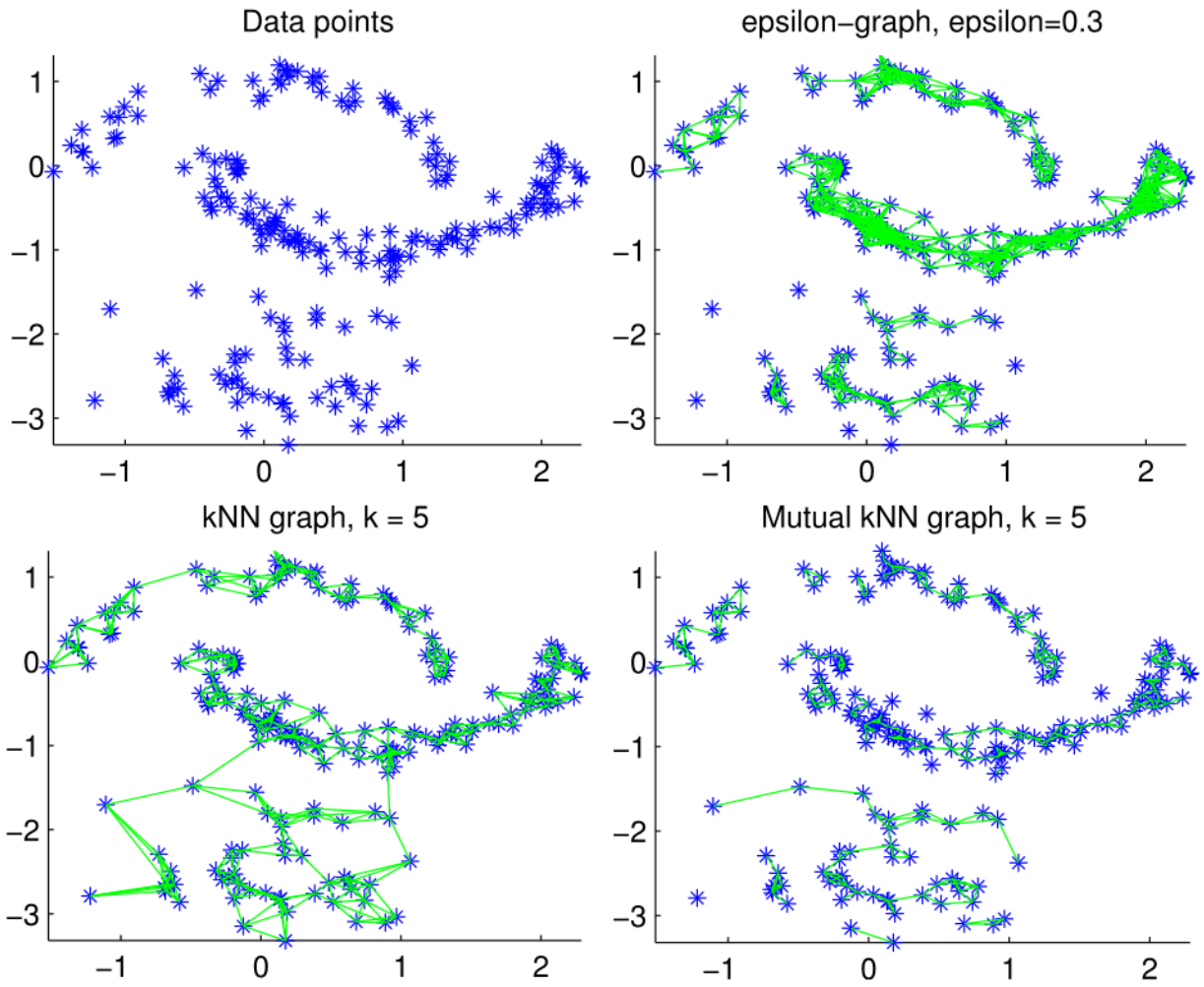


Similarity graphs

Types of graphs:

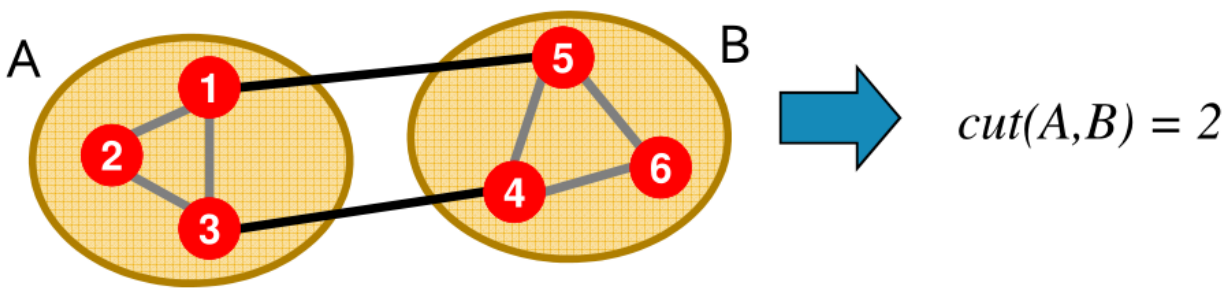
- ε-**neighborhood**:
  - Only include edges with distances $< \varepsilon$;
  - Treat as unweighted: $w_{ij} = Const$.
- **k-NN**:
  - Connect v_i and v_j if v_j is a k-NN of v_i.
  - Weighted by similarity $w_{ij} = s_{ij}$.
  - Directed or undirected.
- **Mutual k-NN**:
  - Same as k-NN, but only include mutual k-NN.

Similarity graphs



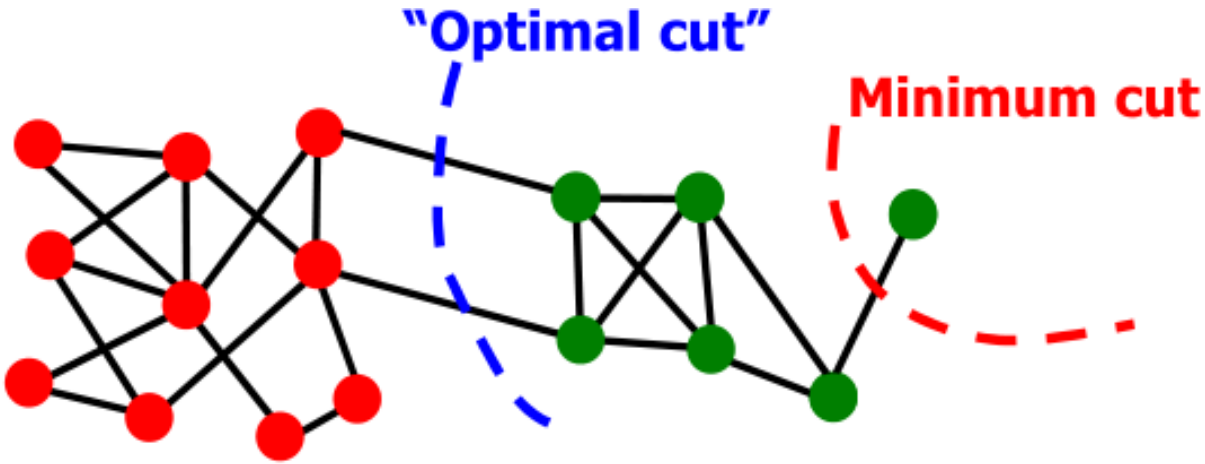
Graph cuts

- **Problem:** Partition graph such that edges between groups have low weights
- **Define:** $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$.
- **MinCut problem:** $Cut(A_1, \ldots, A_k) = \sum_{i=1}^{k} W(A_i, \bar{A}_i)$.
- **Choose:** $A_1, \ldots, A_k = \arg\min_{A_1, \ldots, A_k} Cut(A_1, \ldots, A_k)$.



MinCut

Problem: MinCut favors isolated clusters.



Solution:

- Ratio cuts (RatioCut);
- Normalized cuts (Ncut);
- Lead to "balanced" clusters.

Graph terminology

Two measures of size of a subset:

- Cardinality:

$$|A| = \#\text{ of vertices in } A.$$

- Volume:

$$vol(A) = \sum_{i \in A} \sum_{j=1}^{N} w_{ij}.$$

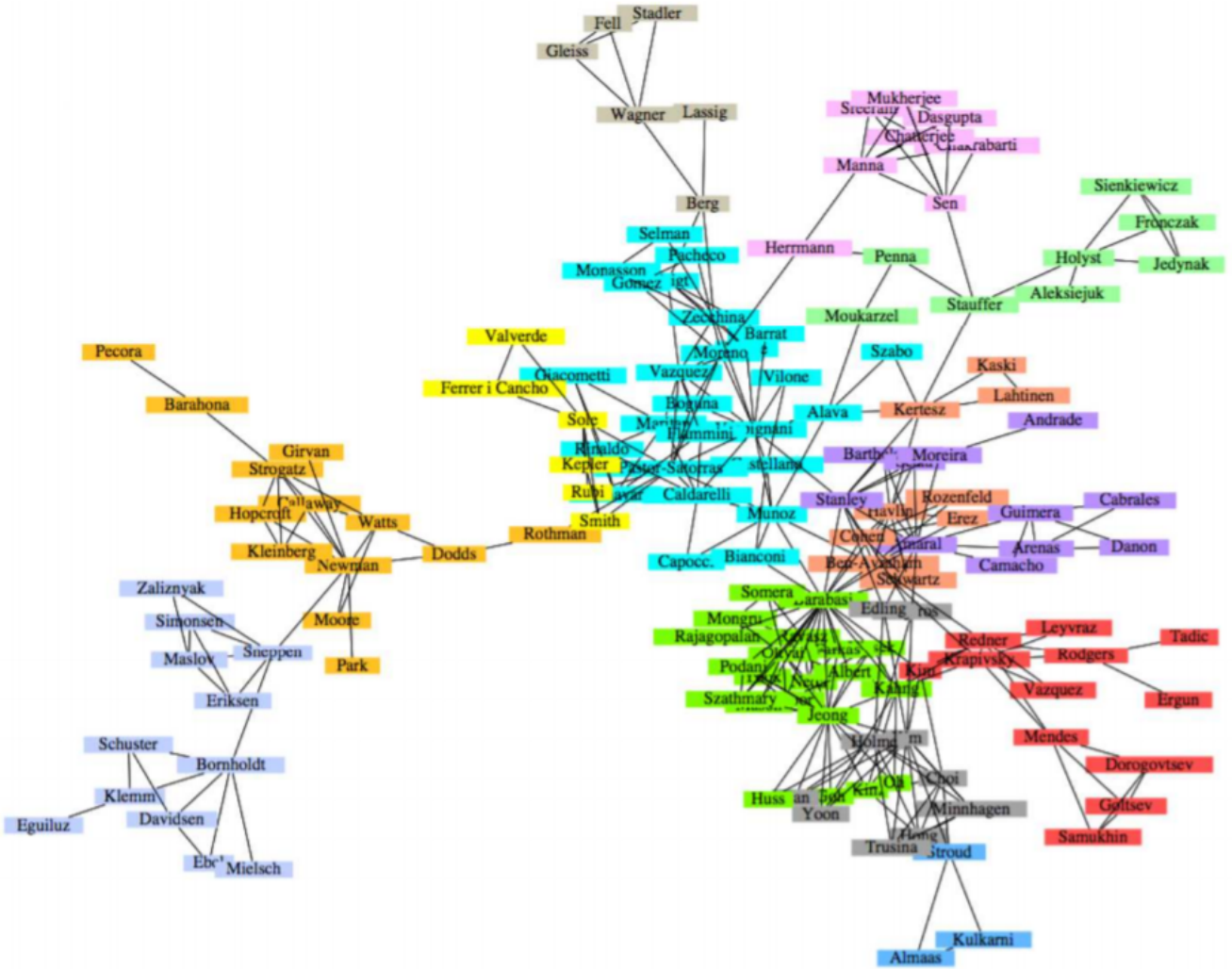
Cuts Accounting for Size

- Ratio cuts (RatioCut)
  - $k = 2$: $RatioCut(A, \bar{A}) = Cut(A, \bar{A})\left(\frac{1}{|A|} + \frac{1}{|\bar{A}|}\right)$.
  - General k: $RatioCut(A_1, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{Cut(A_i, \bar{A}_i)}{|A_i|}$.
- Normalized cuts (Ncut)
  - $k = 2$: $NCut(A, \bar{A}) = Cut(A, \bar{A})\left(\frac{1}{Vol(A)} + \frac{1}{Vol(\bar{A})}\right)$.
  - General k: $NCut(A_1, \ldots, A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{Cut(A_i, \bar{A}_i)}{Vol(A_i)}$
- Problem is NP-hard!
- We need to look at relaxation (solution = Spectral clustering).

Segmentation of Lena

Segmentation of Lena

Citation network



Community detection problems

Objects of study:

- social networks,
- citation/co-authorship networks,
- designing network protocols,
- biological networks,
- ...

Modularity

$$Q = \frac{1}{2m} \sum_{(i,j) \in E} \left(w_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j),$$

where

- d_i is a degree of node i;
- C_i is a community of node i;
- $\delta(C_i, C_j)$ is a delta function;
- $m = |E|$ is a total number of edges in a graph.

Interpretation: difference between the fraction of edges inside the community and its expectation in random graph with fixed node degrees.

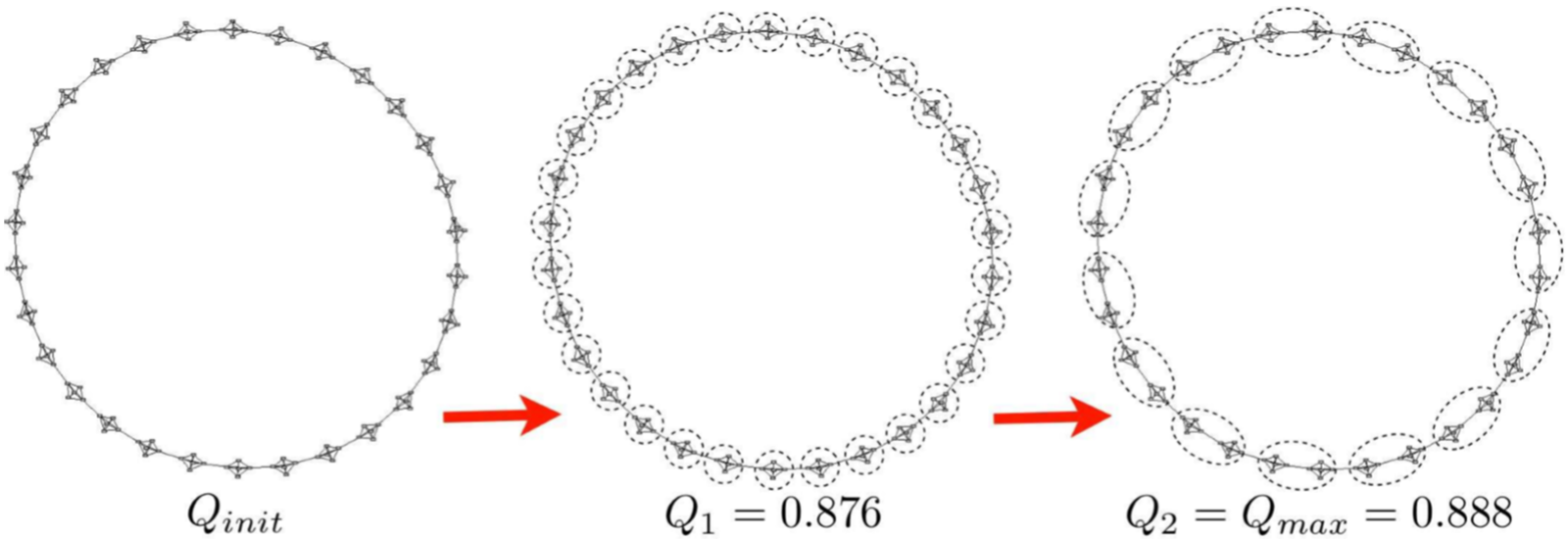
Louvain

Modularity:

$$Q = \frac{1}{2m} \sum_{(i,j) \in E} \left(w_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j),$$

- **Idea:** optimize modularity (discrete optimization problem).
- **Efficient implementation:** Louvain community detection algorithm.
- **Problem:** low resolution.

Low resolution of modularity



Thank you for your attention!