

# Clustering and Community Detection

Maxim Panov



2017

# Overview

- ▶ Clustering:
  - ▶ Agglomerative clustering
  - ▶ K-means
  - ▶ Mixture models
- ▶ Community detection
  - ▶ Spectral clustering
  - ▶ Modularity based clustering
  - ▶ Detecting overlapping communities

## Clustering Problem: Galaxies

- ▶ A catalog of 2 billion “sky objects” represents objects by their radiation in 7 dimensions (frequency bands).
- ▶ **Problem:** Cluster into similar objects, e.g., galaxies, nearby stars, quasars, etc.
- ▶ **Sloan Digital Sky Survey:**



# Clustering Problem: Documents

Finding topics:

- ▶ Represent a document by a vector  $(x_1, x_2, \dots, x_k)$ , 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.

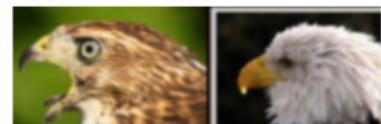
- ▶ **Internal criterion:** members of the cluster should be similar to each other (**inter-cluster compactness**).



tigers



whales



raptors

## What is a cluster?

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

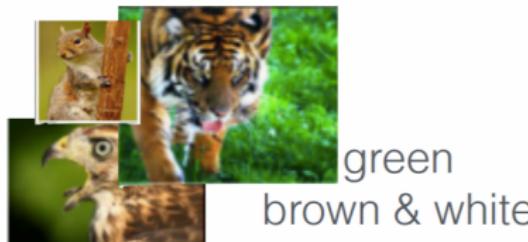
- ▶ **External criterion:** objects outside the cluster should be dissimilar from the objects inside the cluster (**intra-cluster distance**).



# What is a cluster?

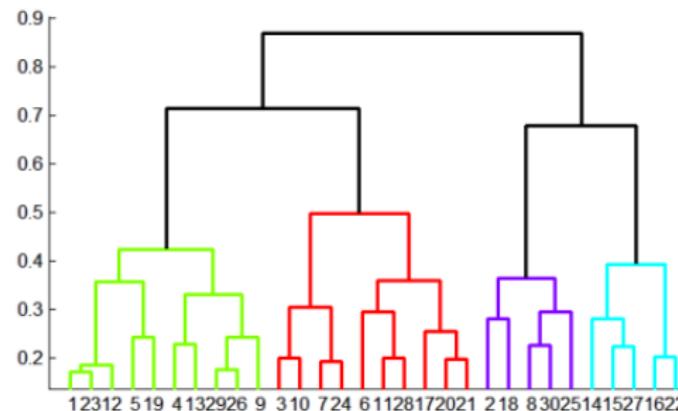
**Goal:** partitioning data in maximally homogeneous, maximally distinguished subsets.

- ▶ **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.

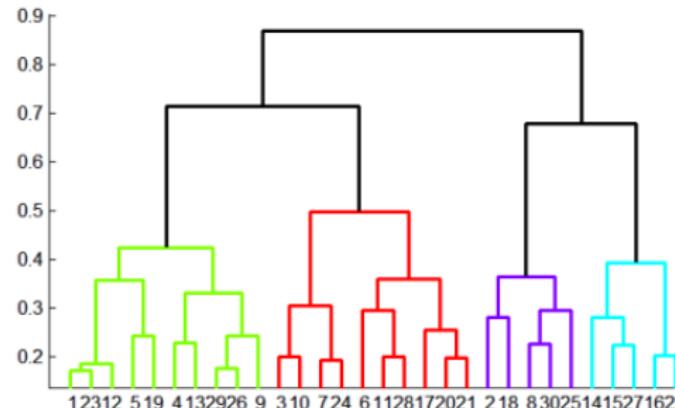


# Hierarchical clustering

**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 distance

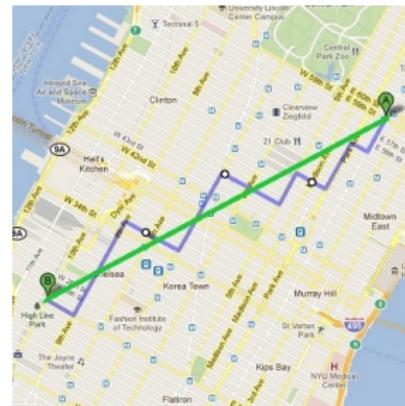
Minkowski family of distances:

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

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 distance

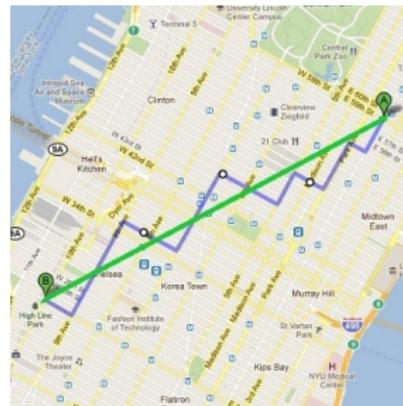
Minkowski family of distances:

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

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:

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

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

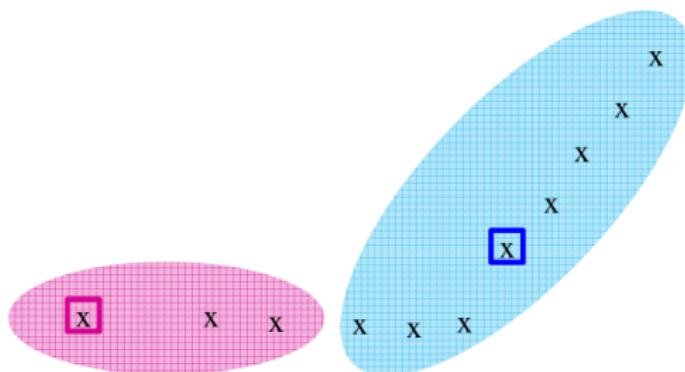
$$z^j = \arg \min_i \|x_j - \mu_i^t\|_2^2.$$

- ▶ **Recenter:**  $\mu_i$  becomes centroid of its points:

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

- ▶ Equivalent to  $\mu_i$  average of its points!

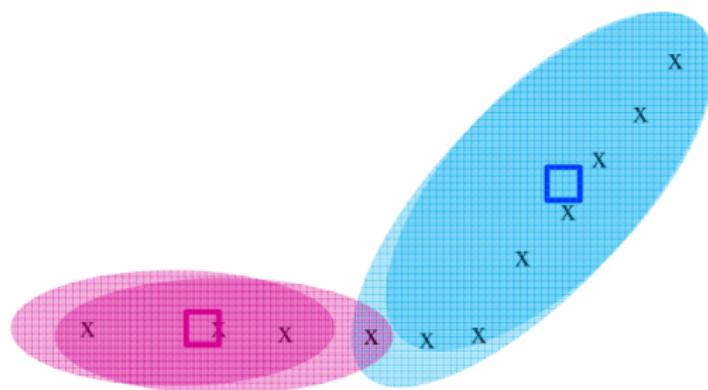
# K-means



X ... data point  
█ ... centroid

Clusters after round 1

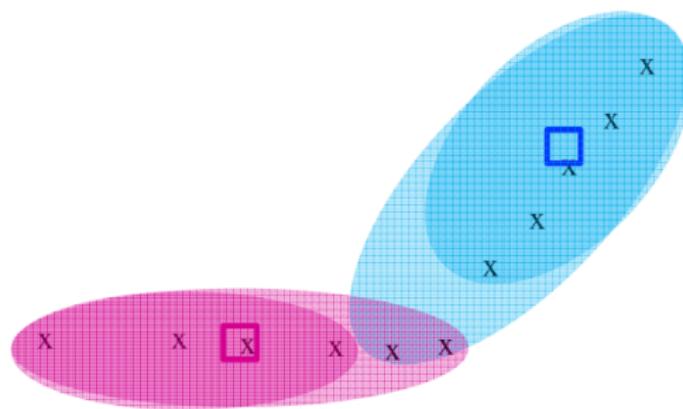
## K-means



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: entropy.
- ▶ **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

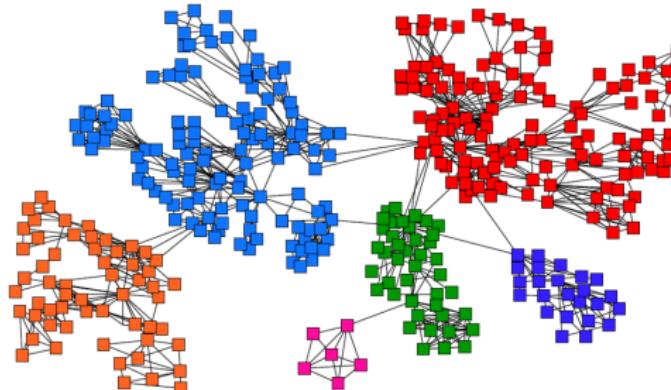
$$RI = \frac{a + d}{a + b + c + d},$$

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.

# 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.

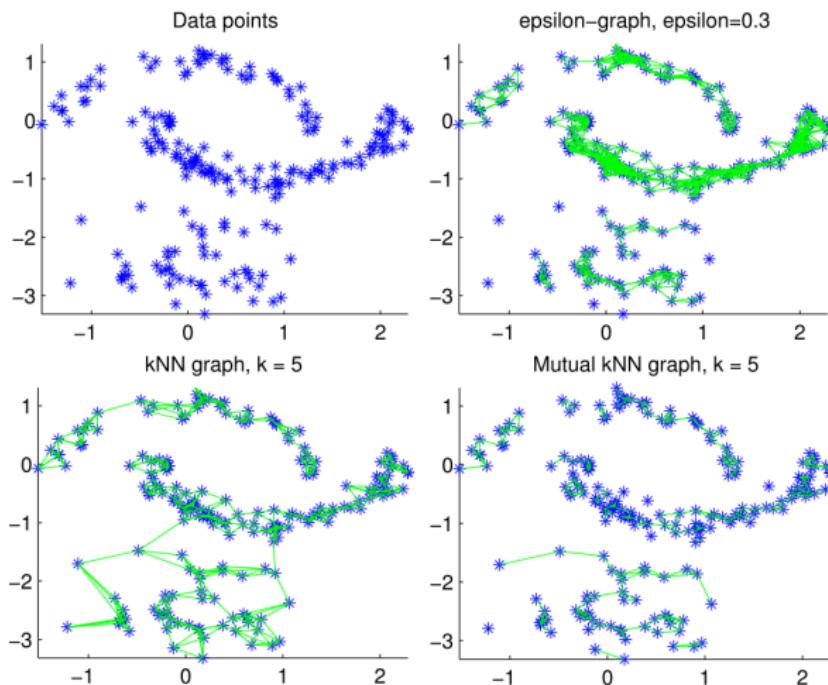


# Similarity graphs

Types of graphs:

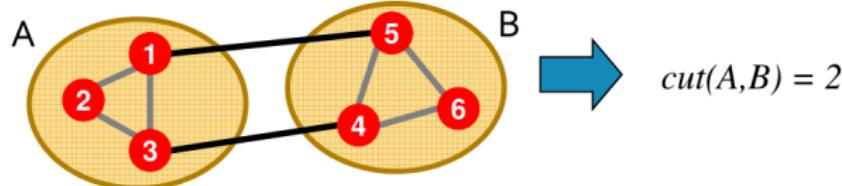
- ▶  **$\varepsilon$ -neighborhood:**
  - ▶ Only include edges with distances  $< \varepsilon$ ;
  - ▶ Treat as unweighted:  $w_{ij} = \text{Const.}$
- ▶ **k-NN:**
  - ▶ Connect  $v_i$  and  $v_j$  if  $v_j$  is a k-NN of  $v_i$ .
  - ▶ Weighted by similarity  $w_{ij} = s_{ij}$ .
  - ▶ Directed or undirected.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

# Similarity graphs



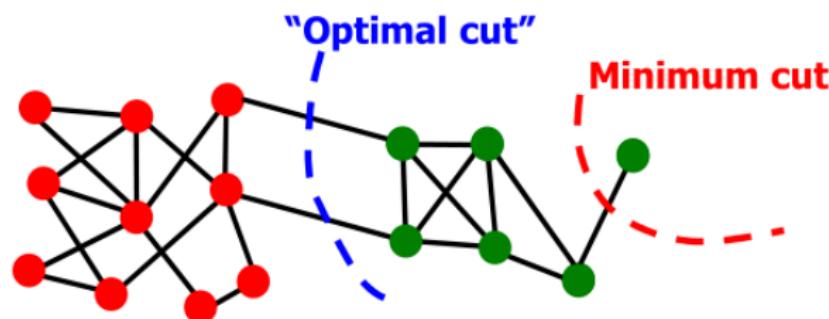
# Graph cuts

- ▶ **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:**  $Cut(A_1, \dots, A_k) = \sum_{i=1}^k W(A_i, \bar{A}_i)$ .
- ▶ **Choose:**  $A_1, \dots, A_k = \arg \min_{A_1, \dots, A_k} Cut(A_1, \dots, A_k)$ .



# 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:

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

# Cuts Accounting for Size

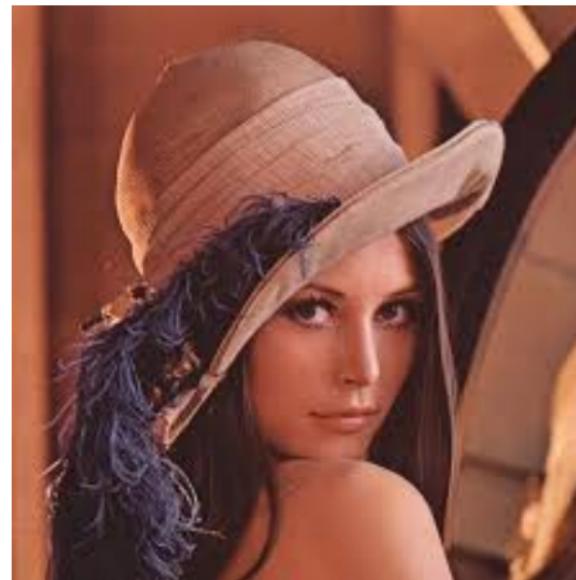
- ▶ Ratio cuts (RatioCut)
  - ▶  $k = 2$ :  $\text{RatioCut}(A, \bar{A}) = \text{Cut}(A, \bar{A}) \left( \frac{1}{|A|} + \frac{1}{|\bar{A}|} \right)$ .
  - ▶ General  $k$ :  $\text{RatioCut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k \frac{\text{Cut}(A_i, \bar{A}_i)}{|A_i|}$ .
- ▶ Normalized cuts (Ncut)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

Clustering

└ Community detection

└ Spectral clustering

## Segmentation of Lena



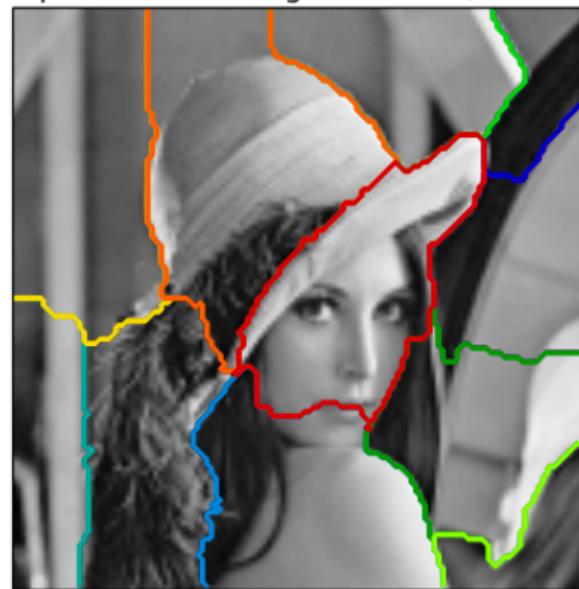
Clustering

└ Community detection

└ Spectral clustering

# Segmentation of Lena

Spectral clustering: discretize, 28.62s

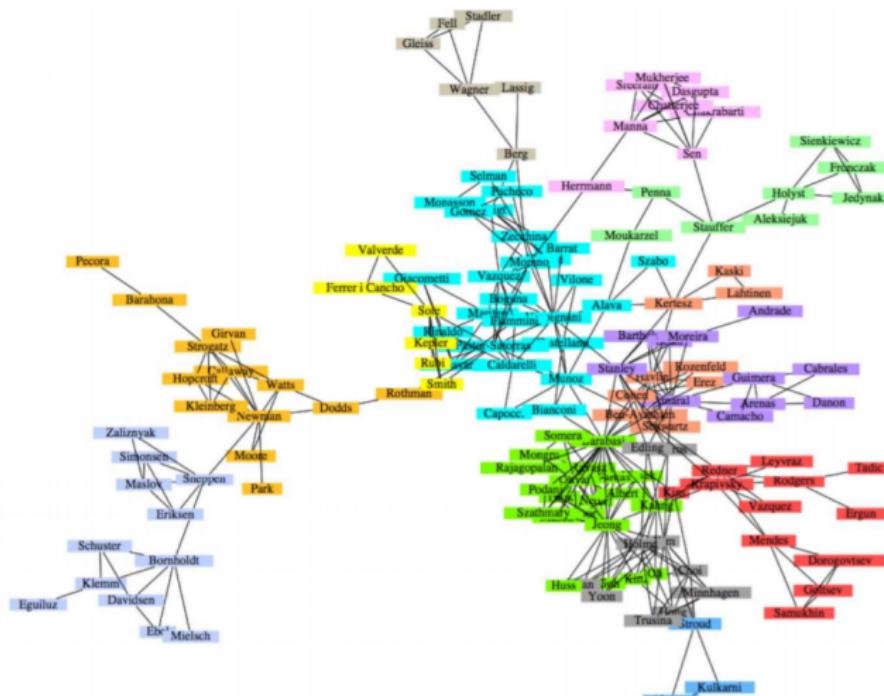


## Clustering

### └ Community detection

#### └ Community detection problems

# 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.

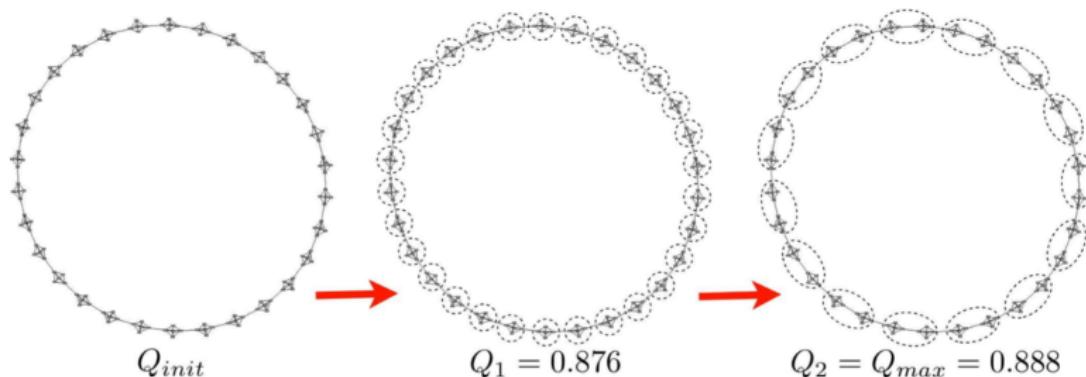
# Louvain

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!