Clusters and Communities

Internet Analytics (COM-308)

Prof. Matthias Grossglauser
School of Computer and Communication Sciences

EPFL

Overview

- Clustering:
  - Given: set of points with a distance metric
  - Find sets of points that are close to each other, but far from other points
- Community detection:
  - Given: network
  - Find sets of nodes that are highly interconnected, but poorly connected to other nodes
- Many important applications:
  - Data analysis: finding structure, modes in data distribution
  - Problem decomposition and resource allocation: where to put warehouses; group formation in social networks
  - …

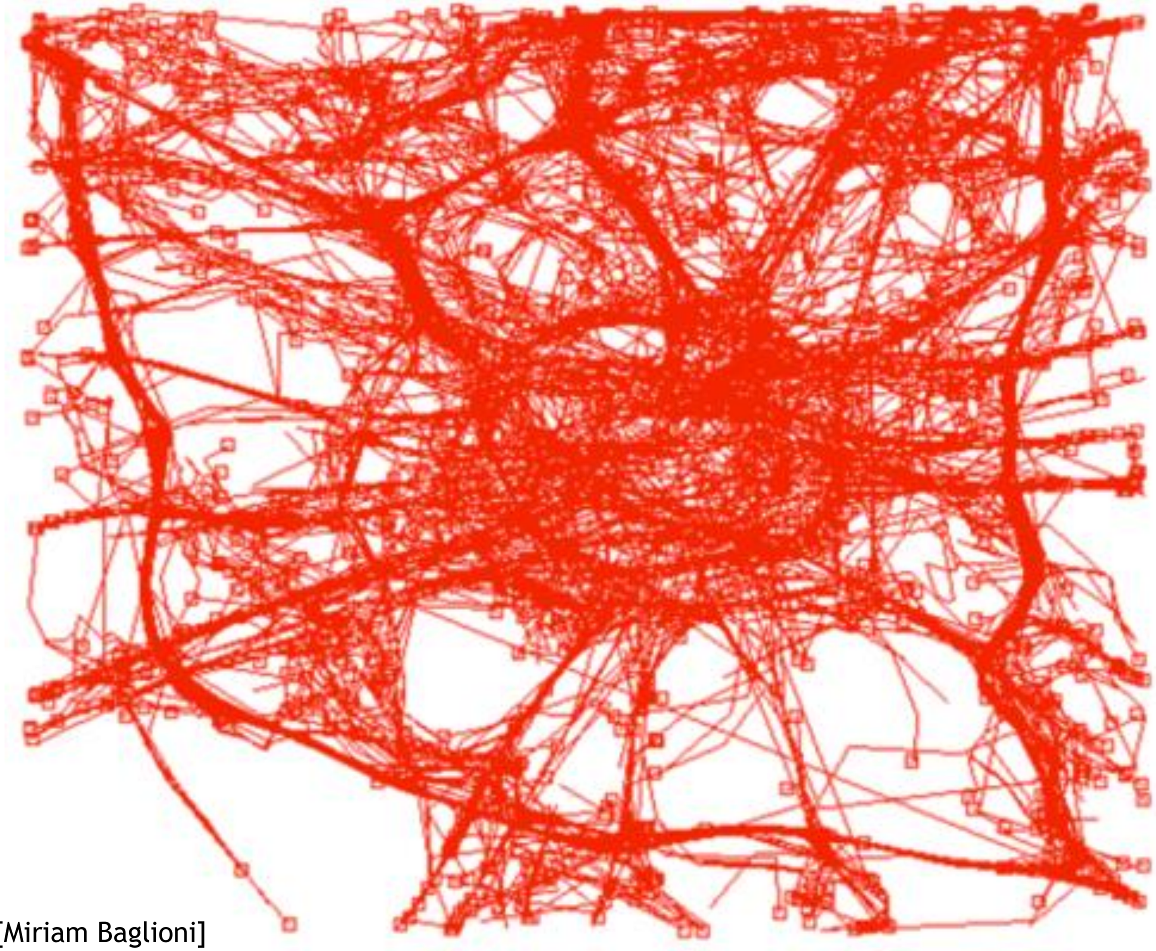
Clustering: goal and definition

- Definition:
  - Given: set of points (vectors) with a distance function (metric space)
  - Find: partition (hard or soft) of points into clusters; plus potentially more information (characterization of clusters)
- Find organization in data:
  - Image compression: each pixel has a color → find small set of colors so that each pixel is close to one color

[Stackexchange.com]

Clustering: goal and definition

- Find organization in data:
  - Mobility: point = GPS trace with noise → find representative set of trajectories



[Miriam Baglioni]

K-means clustering algorithm

- Input:
  - N data points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$
  - K: number of clusters
- Output:
  - K cluster centers $\boldsymbol{\mu}_k$
  - r_{nk}: point-cluster assignment indicator
    - $r_{nk} = 1$ means point $\boldsymbol{x}_n$ is in cluster k
- Cost function:
  - $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$
- Optimal K-means: NP-hard
- Solution: iterative heuristic to approximate solution

K-means: iterative approximation

- Initialize $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K)$
- Until convergence (J does not decrease):
  - Minimize J w.r.t. $\{r_{nk}\}$:
    - $r_{nk} = 1$ only for
      $$k = \operatorname{argmin} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|$$
  - Minimize J w.r.t. $\boldsymbol{\mu}$:
    - Set gradient of J w.r.t. $\boldsymbol{\mu}_k$ to zero
    - $2 \sum_n r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$ (for each k)
    - Solve for $\boldsymbol{\mu}$: $\boldsymbol{\mu}_k = \dfrac{\sum_n r_{nk} \boldsymbol{x}_n}{\sum_n r_{nk}}$

E-step:
Attribute each $\boldsymbol{x}_n$
to closest center

M-step:
New cluster center $\boldsymbol{\mu}_k$
= center of mass of
points of cluster k

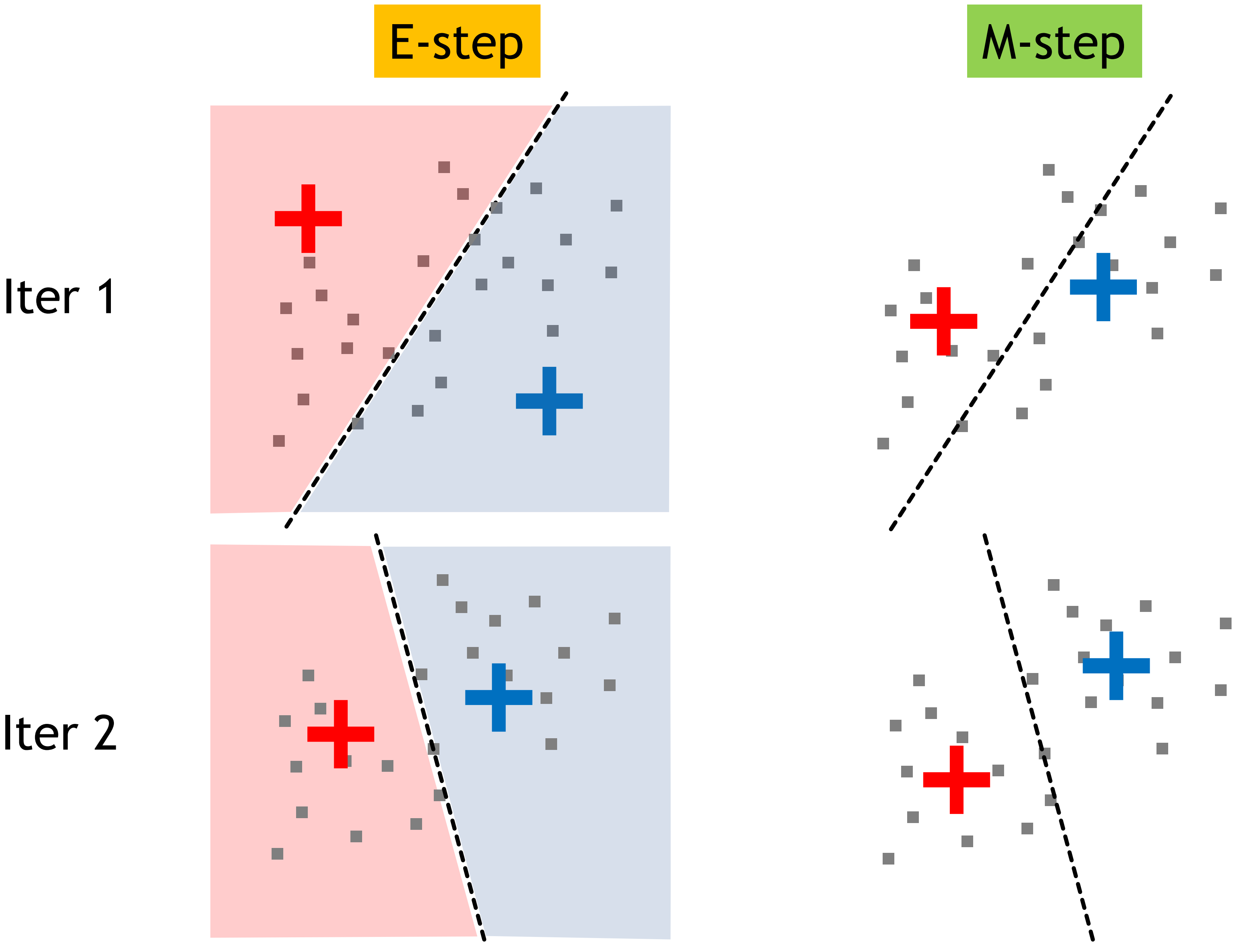
Voronoi tessellation

[baeldung.com]

K-means converges in finite # steps

- Proof:
  - J non-increasing in both steps
    - First step: each $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ either stays the same or decreases if r_{nk} changes
    - Second step: convex → new J global min as function of $\{\mu_k\}$
  - There are finitely many configurations r_{nk} → if we ever returned to a configuration of $\{r_{nk}\}$ already visited (in a finite # of steps), we'd end up with the same J in step 2 as last time in step 2 → must have already converged
- But: there is no guarantee of convergence to globally minimal J over both $\{r_{nk}\}, \{\boldsymbol{\mu}_k\}$

K-means: example



From K-means to Mixtures of Gaussians

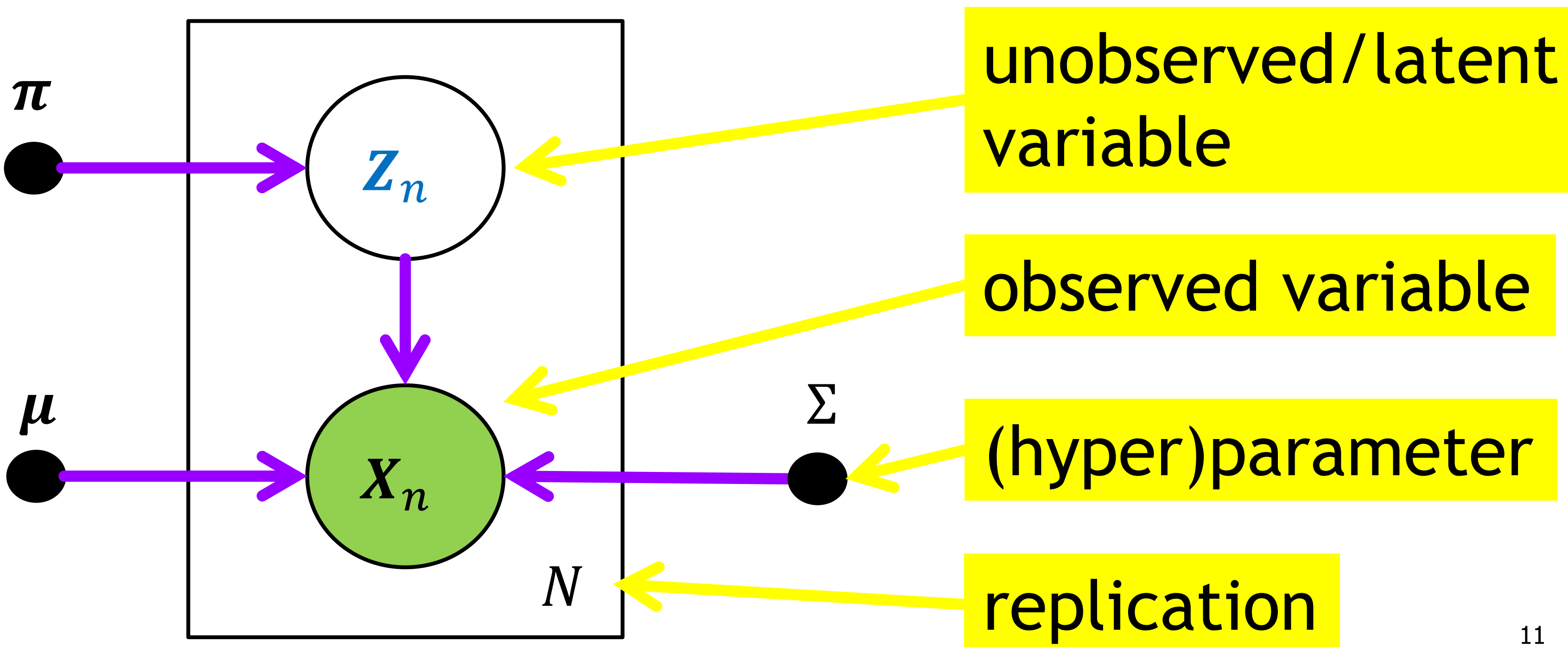
- K-Means: Can be generalized to non-Euclidean distance functions
- Limitations:
  - Each point attributed to exactly one cluster
  - Not a generative model: cannot “simulate” data based on learned $\{r_{nk}\}, \{\boldsymbol{\mu}_k\}$ (no distribution for new values)
- Improvement:
  - Soft attribution: a point can belong to several clusters
  - Generative model: distribution over points

Gaussian mixture model (GMM)

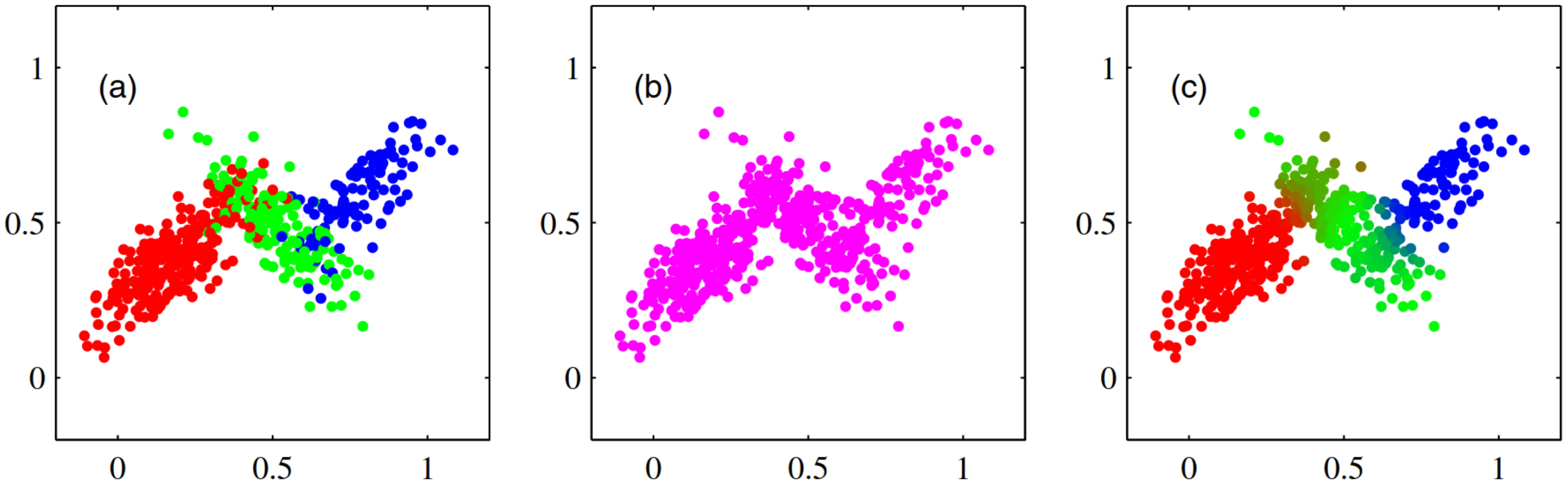
- GMM: distribution of single data point

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \, N(\boldsymbol{x}; \mu_k, \Sigma_k)$$

- Random variable Z_{nk}: point n belongs to cluster k
  - $p(Z_{nk} = 1) = \pi_k$: mixing coefficients



GMM is a generative model



[Pattern Recognition and Machine Learning: Ch. Bishop, 2006]

GMM

- Latent variable $\boldsymbol{Z}_n = (Z_{n1}, Z_{n2}, \dots, Z_{nk}, \dots, Z_{nK})$:
  - $p(\boldsymbol{Z}_n) = \prod_k {\pi_k}^{Z_{nk}}$
  - "One-hot": $\boldsymbol{Z}_n = (0,0,0, \dots, 1, \dots, 0,0)$
- Conditional distribution of data point:
  - $p(\boldsymbol{X}_n | Z_{nk} = 1) = N(\boldsymbol{X}_n; \mu_k, \Sigma_k)$, or equivalently
  - $p(\boldsymbol{X}_n | \boldsymbol{Z}_n) = \prod_k N(\boldsymbol{X}_n; \mu_k, \Sigma_k)^{Z_{nk}}$
- Data distribution (unconditional):
  - $p(\boldsymbol{X}_n) = \sum_{\boldsymbol{z}_n} p(\boldsymbol{Z}_n) p(\boldsymbol{X}_n | \boldsymbol{Z}_n) =$
  - $= \sum_{\boldsymbol{z}_n} \prod_k (\pi_k N(\boldsymbol{X}_n; \mu_k, \Sigma_k))^{Z_{nk}} = \sum_k \pi_k N(\boldsymbol{X}_n; \mu_k, \Sigma_k)$
- Conclusion:
  - Gaussian mixture can be viewed as follows: choose a cluster k with distribution $\boldsymbol{\pi}$; then generate a point according to Gaussian $N(\boldsymbol{X} | \mu_k, \Sigma_k)$ of the chosen cluster

Fitting single Gaussian to data

- Data vectors $\boldsymbol{X}_1, \dots, \boldsymbol{X}_N$
- Model as $N(\boldsymbol{\mu}, \Sigma)$
- Maximum likelihood:
  - $\mu = \frac{1}{N}\sum_n \boldsymbol{X}_n$: empirical mean
  - $\Sigma = \frac{1}{N}\sum_n (\boldsymbol{X}_n - \boldsymbol{\mu})(\boldsymbol{X}_n - \boldsymbol{\mu})^T$: empirical covariance

GMM: posterior

- Finding clusters = computing posterior, ie, distribution of $\boldsymbol{Z}_n$ given data $\boldsymbol{X}_n$
- Def: $\gamma_{nk} = p(Z_{nk} = 1|\boldsymbol{X}_n)$
- Bayes' theorem:
  - $$\gamma_{nk} = \frac{p(Z_{nk}=1)p(\boldsymbol{X}_n|Z_{nk}=1)}{\sum_j p(Z_{nj}=1)p(\boldsymbol{X}_n|Z_{nj}=1)} = \frac{\pi_k N(\boldsymbol{X}_n;\boldsymbol{\mu}_k,\Sigma_k)}{\sum_j \pi_j N(\boldsymbol{X}_n;\boldsymbol{\mu}_j,\Sigma_j)}$$
- Interpretation:
  - For fixed $\{\boldsymbol{\mu}_k, \Sigma_k\}$,
    $\boldsymbol{\pi}$ is the prior for the cluster $\boldsymbol{Z}_n$ of point $\boldsymbol{X}_n$, and
    $\boldsymbol{\gamma}_n$ is its posterior

GMM: ML estimator for μ

- Log-likelihood for n data points $(X_1, \ldots, X_N)$:
  - $L = \log p(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N | \boldsymbol{\pi}, \mu, \Sigma) = \sum_n \log \sum_k \pi_k N(\boldsymbol{x}_n; \boldsymbol{\mu}_k, \Sigma_k)$
- Maximizing w.r.t. $\boldsymbol{\mu}$:
  - $\nabla_{\boldsymbol{\mu}_k} L = 0 \Rightarrow$

$$\sum_n \underbrace{\frac{\pi_k N(\boldsymbol{X}_n; \boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j N(\boldsymbol{X}_n; \boldsymbol{\mu}_j, \Sigma_j)}}_{= \gamma_{nk} \text{: posterior of } Z_{nk} \text{ given } \boldsymbol{X}_n} \Sigma_k^{-1} (\boldsymbol{X}_n - \boldsymbol{\mu}_k) = 0$$

- Solution:
  - $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_n \gamma_{nk} \boldsymbol{X}_n$ (roughly, weighted center of mass)
  - with $N_k = \sum_n \gamma_{nk}$ (roughly, # of points in class k)

GMM: ML estimator for Σ and π

- Maximizing w.r.t. $\boldsymbol{\Sigma}$:
  - $\Sigma_k = \frac{1}{N_k} \sum_n \gamma_{nk} (\boldsymbol{X}_n - \boldsymbol{\mu}_k)(\boldsymbol{X}_n - \boldsymbol{\mu}_k)^T$
  - (roughly, weighted empirical covariance matrix within class k)
- Maximizing w.r.t. $\boldsymbol{\pi}$:
  - $\pi_k = \frac{N_k}{N}$
  - (roughly, number of points attributed to cluster k)

EM algorithm for GMM

E-step:
Compute posterior of latent variables Z given parameters from M-step:

$$\gamma_{nk} = \frac{\pi_k N(X_n;\mu_k,\Sigma_k)}{\sum_j \pi_j N(X_n;\mu_j,\Sigma_j)}$$

M-step:
Compute new parameters using distribution of latent variables from E-step:

$$\mu_k = \frac{1}{N_k}\sum_n \gamma_{nk} X_n$$

$$\Sigma_k = \frac{1}{N_k}\sum_n \gamma_{nk} (X_n - \mu_k)(X_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

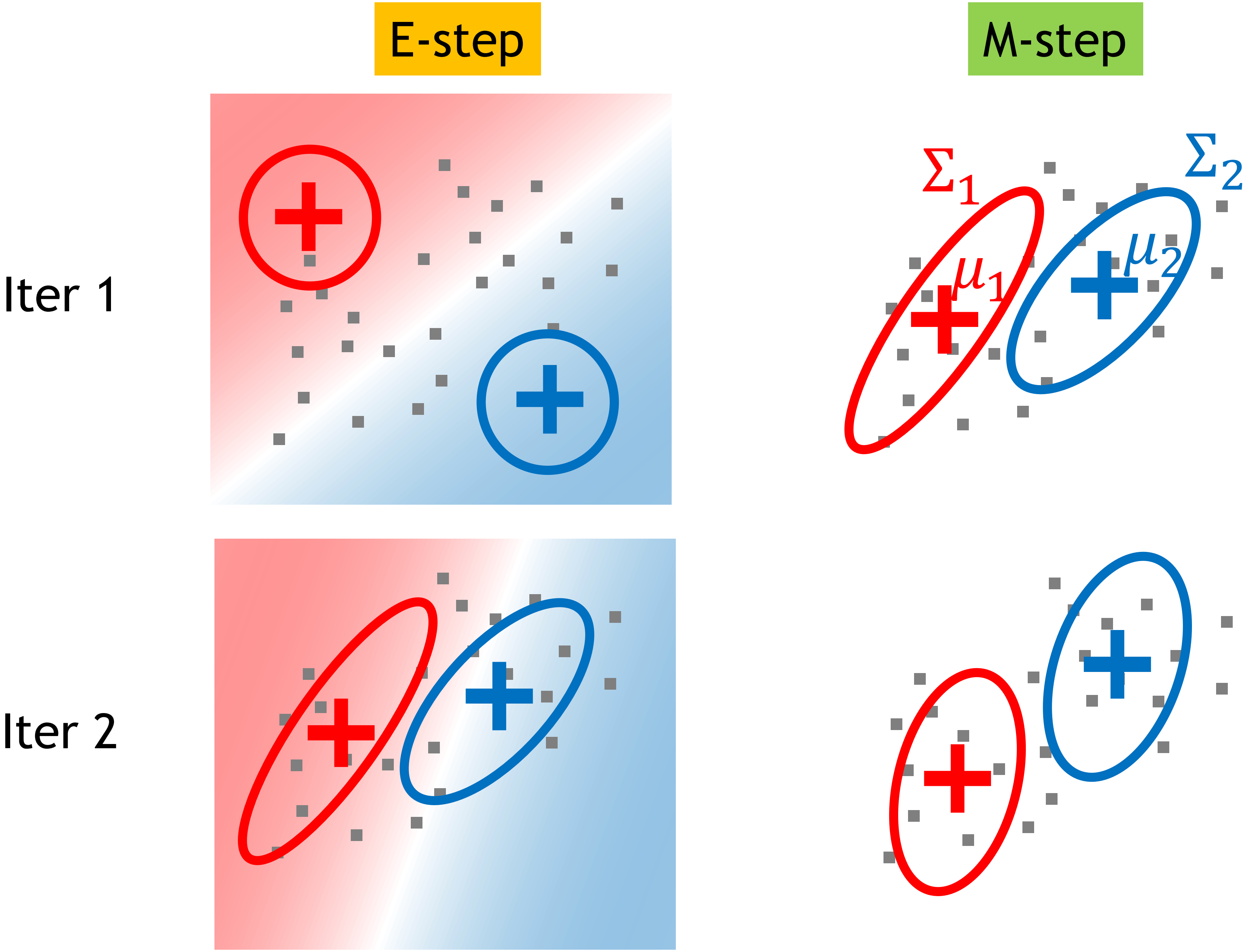
E-step 2

M-step 2

E-step 3

Until convergence of likelihood
$L = \log p(X_1, \dots, X_n | \pi, \mu, \Sigma)$

EM for GMM: example



K-means vs GMM

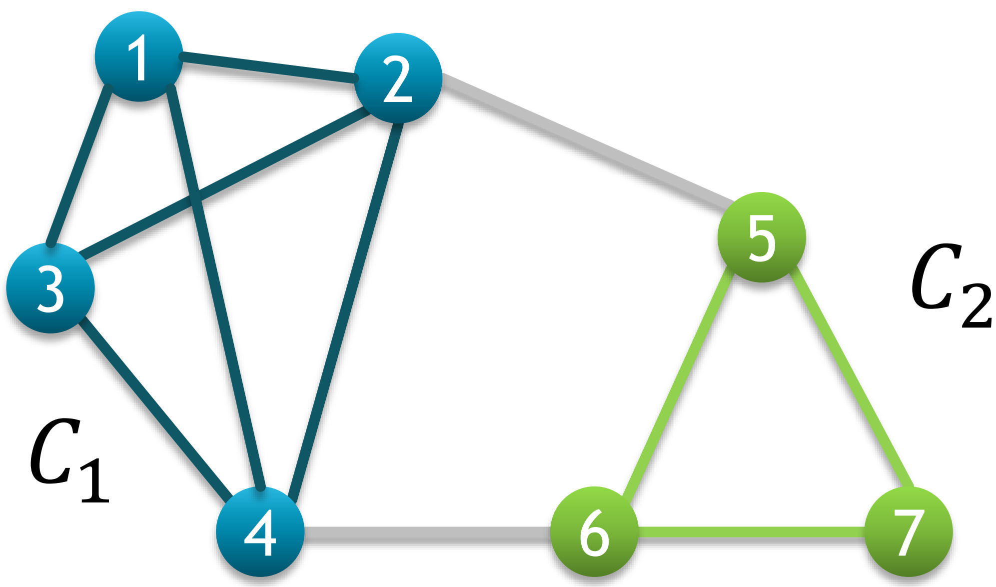
	K-Means	GMM
Membership	Hard	Soft
Generative	No	Yes
E-step updates	r_{nk} = closest center	$\gamma_{nk} = P(Z_{nk} \mid X_n)$
M-step updates	μ_k	μ_k, Σ_k, π_k
Convergence	Guaranteed	Guaranteed
Optimal	Not guaranteed	Not guaranteed
Characterization	Centers, membership	Centers, weights, shapes

Community detection: goal and def

- Find organization in graphs:
  - Email or phone graph → find organizational units
  - Citation networks → scientific topics and their relationships
  - Social networks → groups with shared interests (language, etc.)
- Definition:
  - Given: a network $G(V, E)$
  - Find: a partition (hard or soft), or a hierarchy, such that node in same community are more “meshed” than other nodes

Modularity: strength of communities

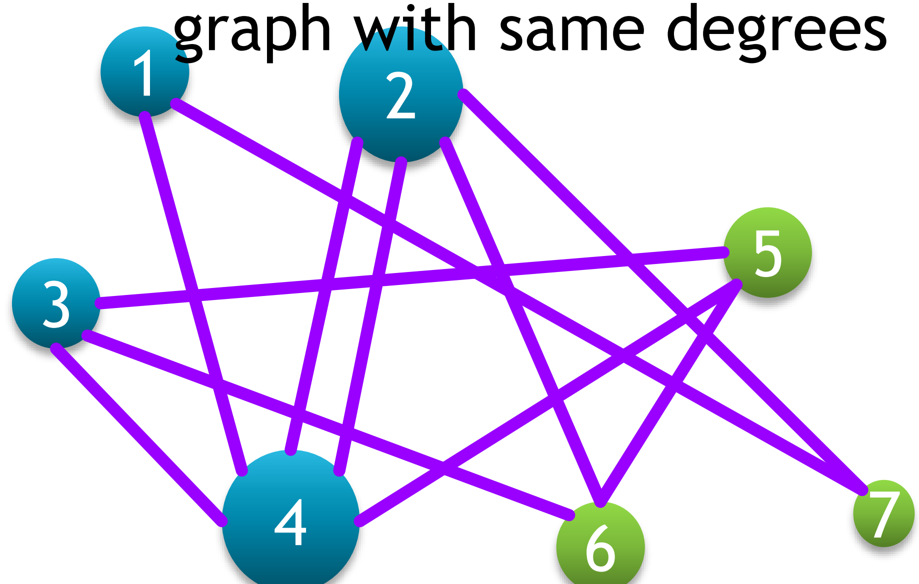
- Def:
  - Partitioning of nodes into communities: $\{C_i\}$
  - $Q = \frac{1}{2m} \sum_{C_i \in C} \sum_{u,v \in C_i} \left(\mathbb{1}_{uv} - \frac{d_u d_v}{2m} \right)$



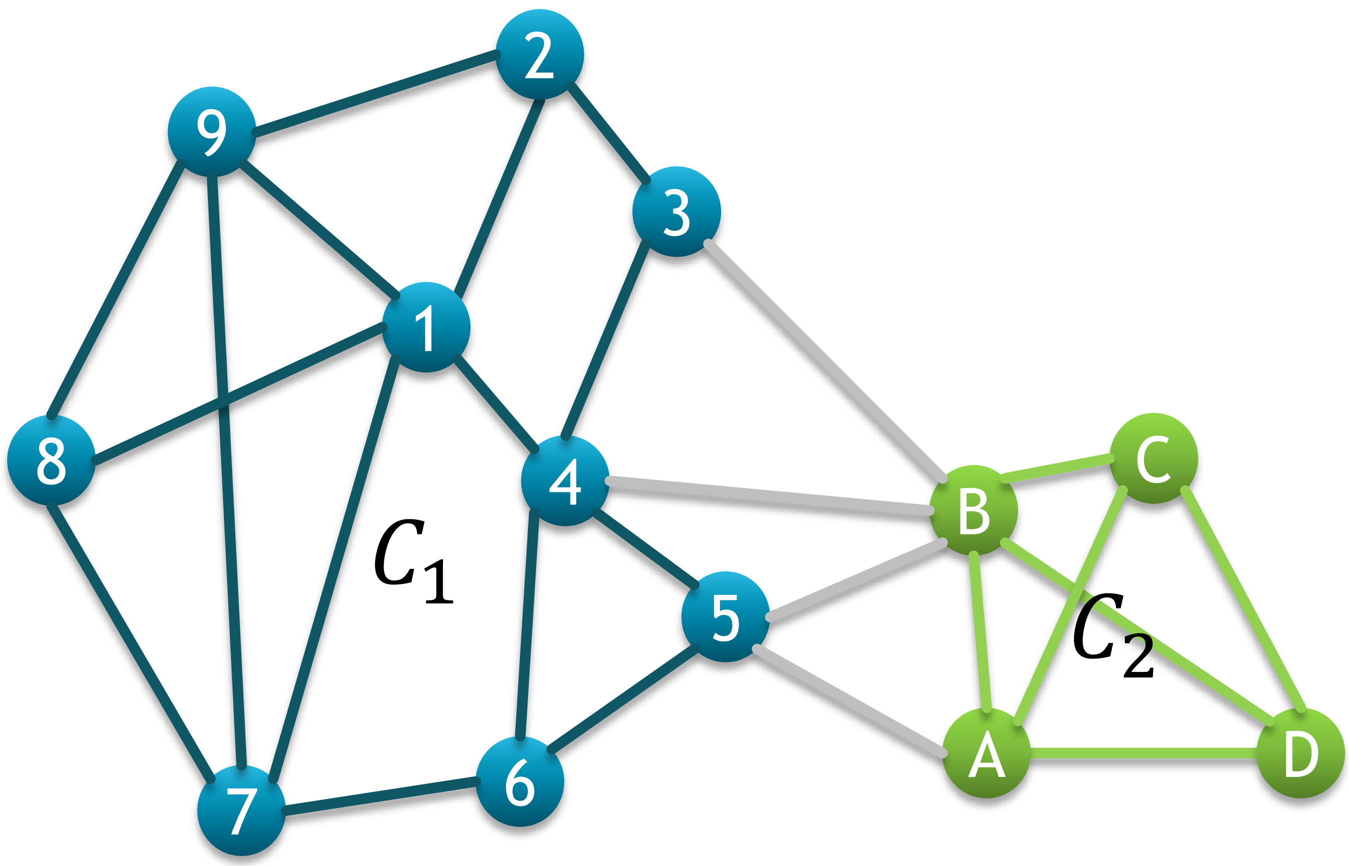
Note: inner sum is over all *ordered* (u, v), and includes (u, u)

Modularity: interpretation

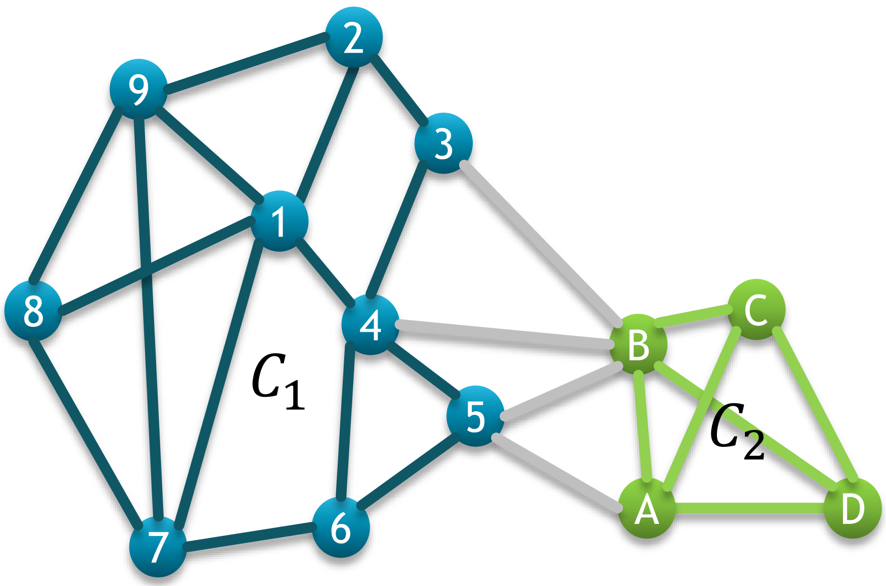
- Number of stubs: $2m$
- Expected # edges (u, v) in C_i: $\frac{1}{2}\sum_{u,v \in C_i} \frac{d_u d_v}{2m} = e_i$
- Actual # edges in community C_i: $\frac{1}{2}\sum_{u,v \in C_i} \mathbb{1}_{uv} = m_i$
- Modularity: compares actual graph with unstructured graph with same node weights
- $Q = \frac{\sum_{C_i}(m_i - e_i)}{m}$
  - m_i: # edges in community C_i
  - e_i: expected # edges in community C_i in random graph with same degrees



Modularity: example



Modularity: example

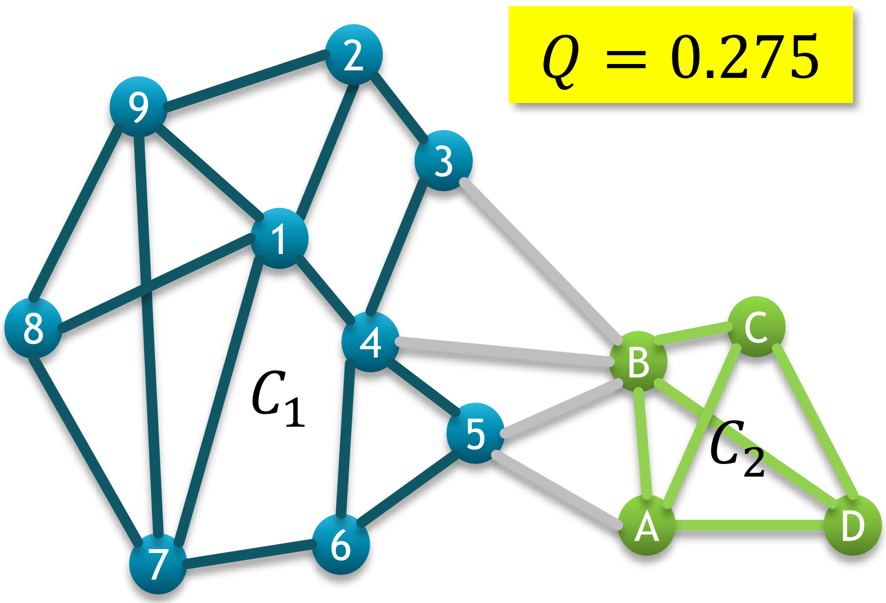


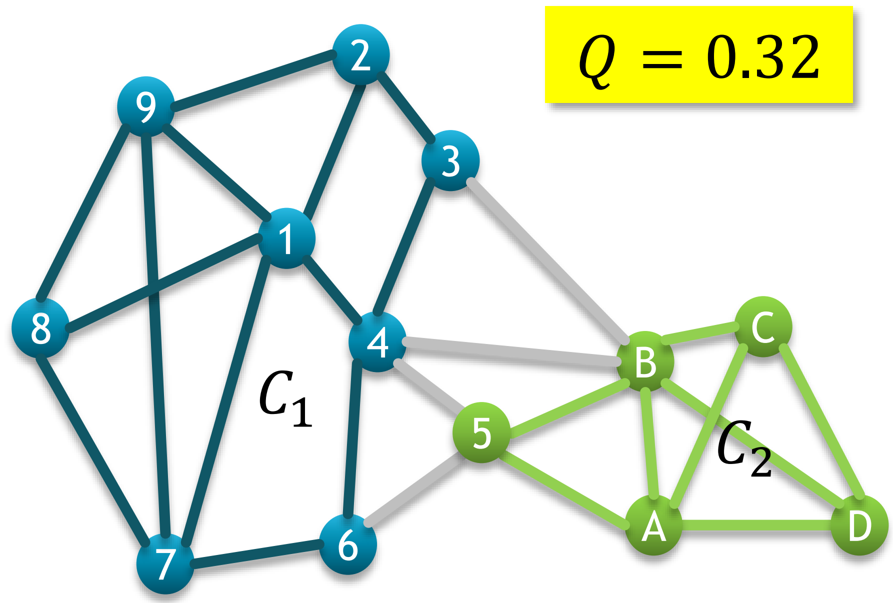
C_1:
- $m_1 = 15$
- $e_1 = 11.56$

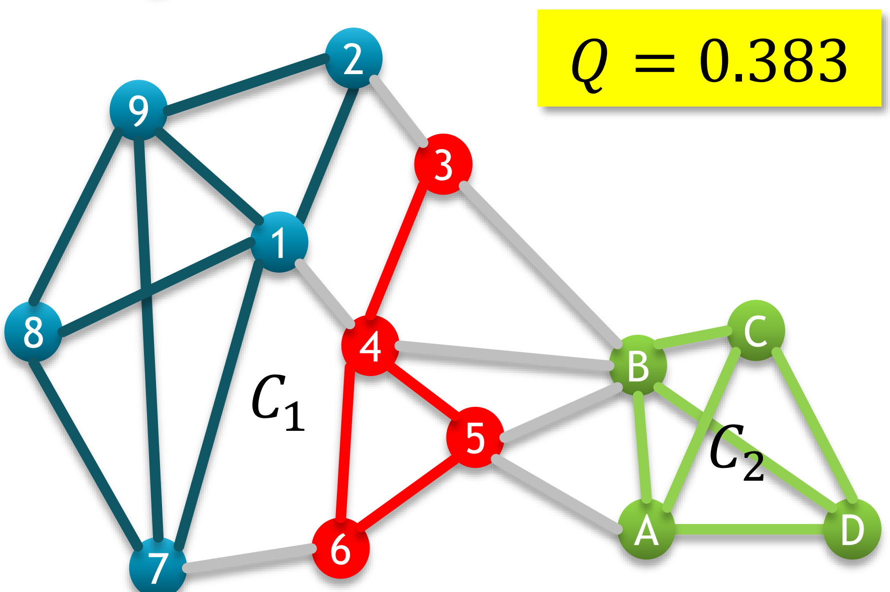
C_2:
- $m_2 = 6$
- $e_2 = 2.56$

- $m = 25$
- Edge $(4{,}8)$:
  - $d_4 = 5; d_8 = 3$
  - Expected # $= \frac{15}{50} \sim 0.3$
- $Q = \frac{(15+6)-(11.56+2.56)}{25} = 0.275$

Modularity: example

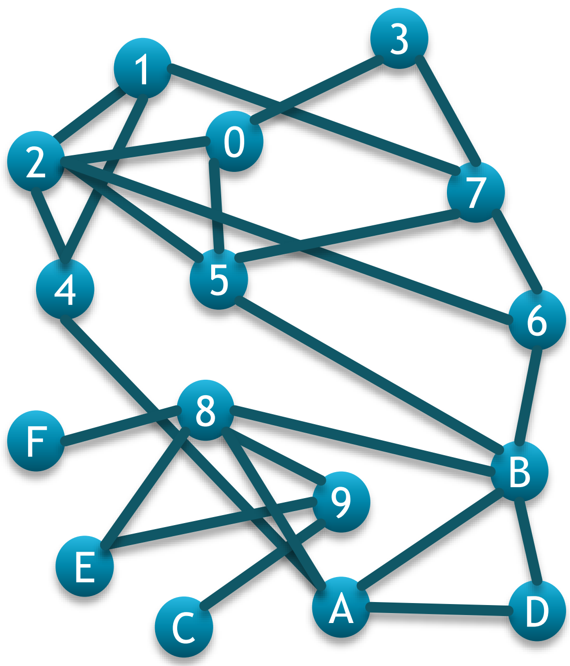






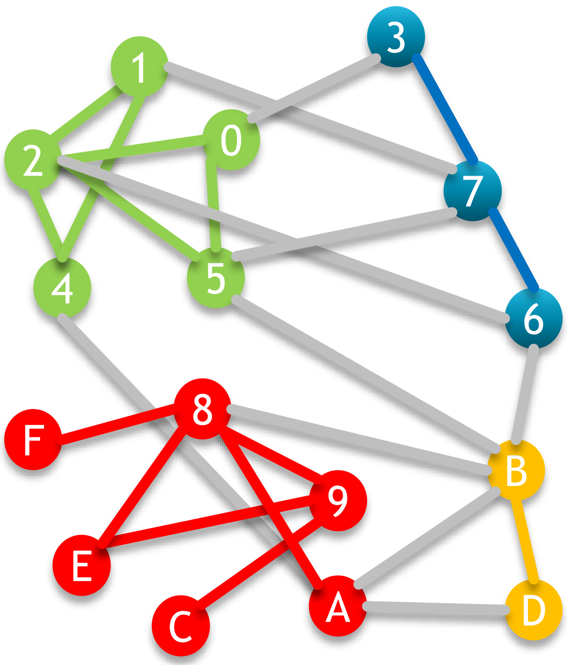
- Max-modularity is NP-hard
- Need efficient heuristics

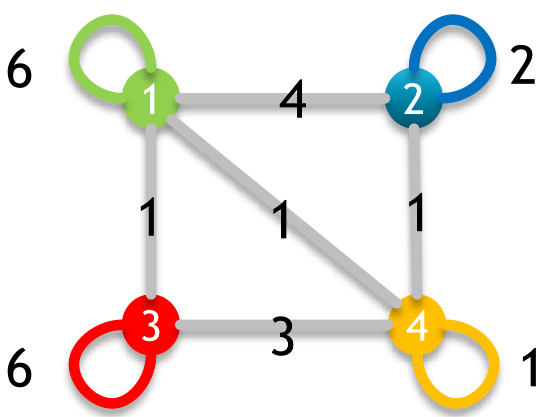
Louvain method

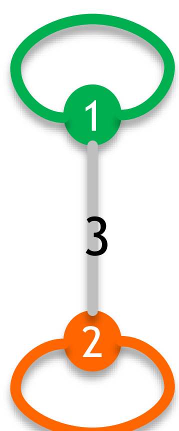


- Idea:
  - Building hierarchy of communities
  - Bottom-up: start with each node a separate community, then coalesce communities
- For every node u:
  - Compare modularity if u is added to the community of a neighbor v
  - Choose neighbor v that increases modularity most; if none, leave u in current community

Louvain method

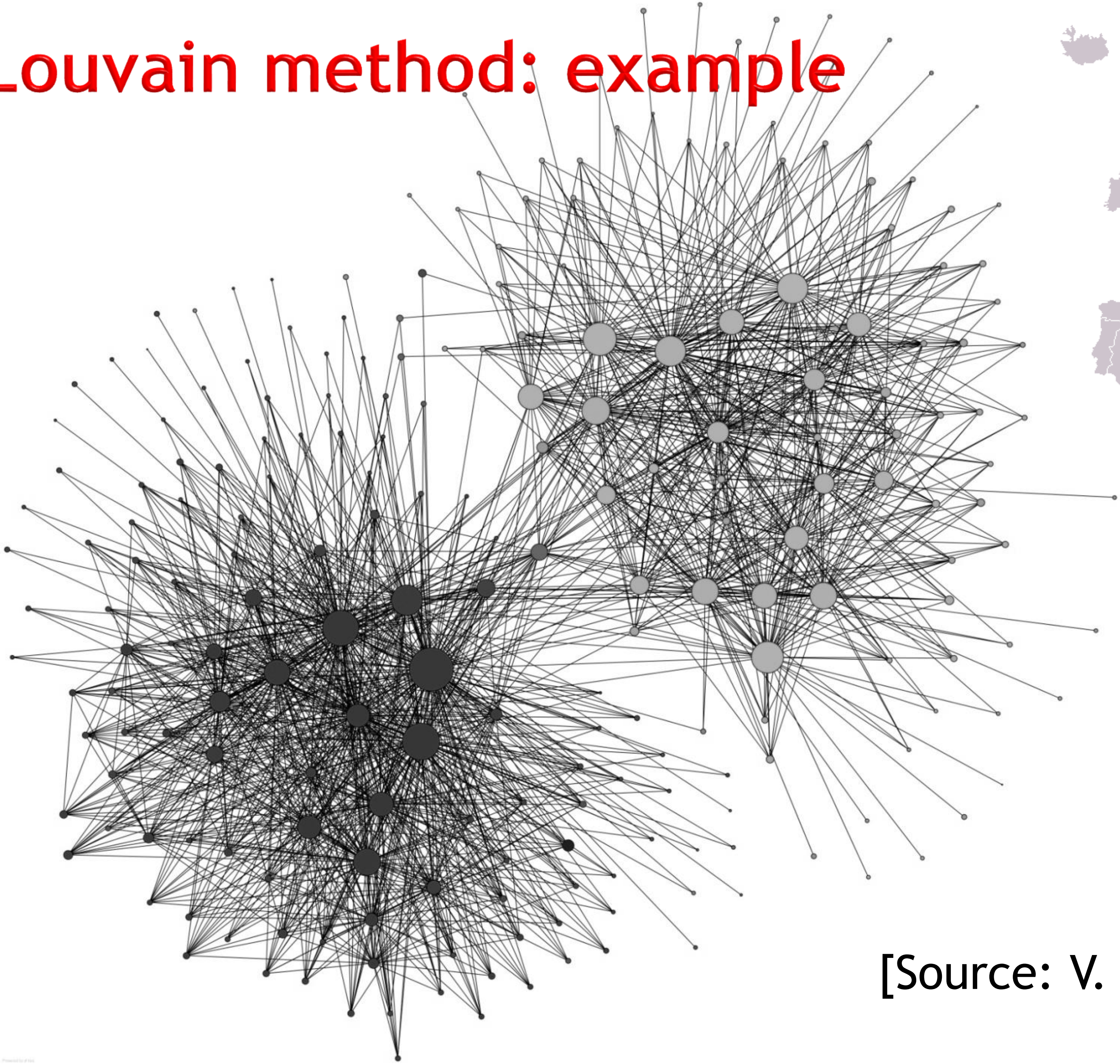






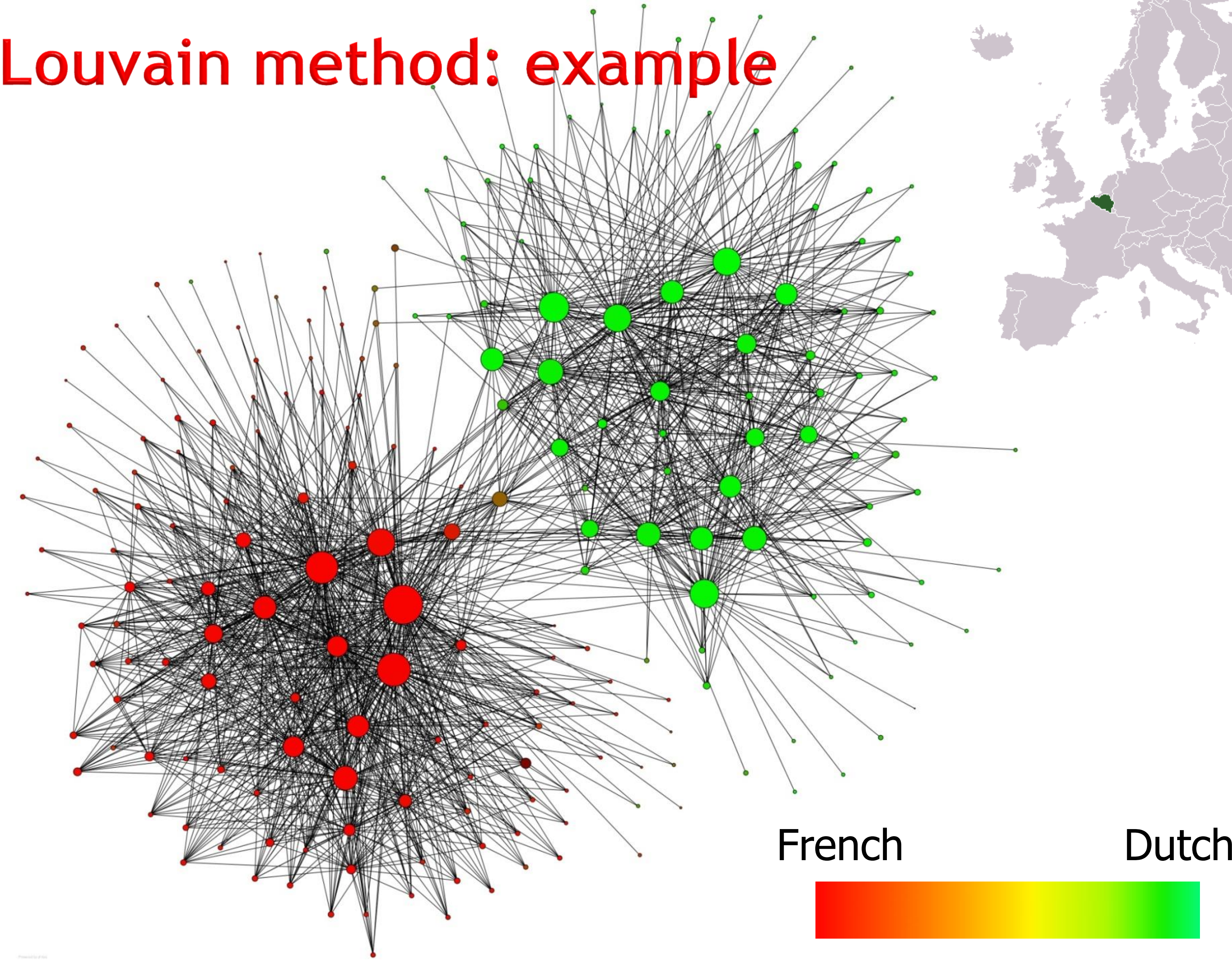
- Iterate through all nodes u (possibly several times) until no more modularity increases possible
  - Local maximum
- Form a new graph capturing the network of communities
  - Link weights = # edges between communities
  - Self-loops: internal edges
- Repeat the procedure until convergence

Louvain method: example

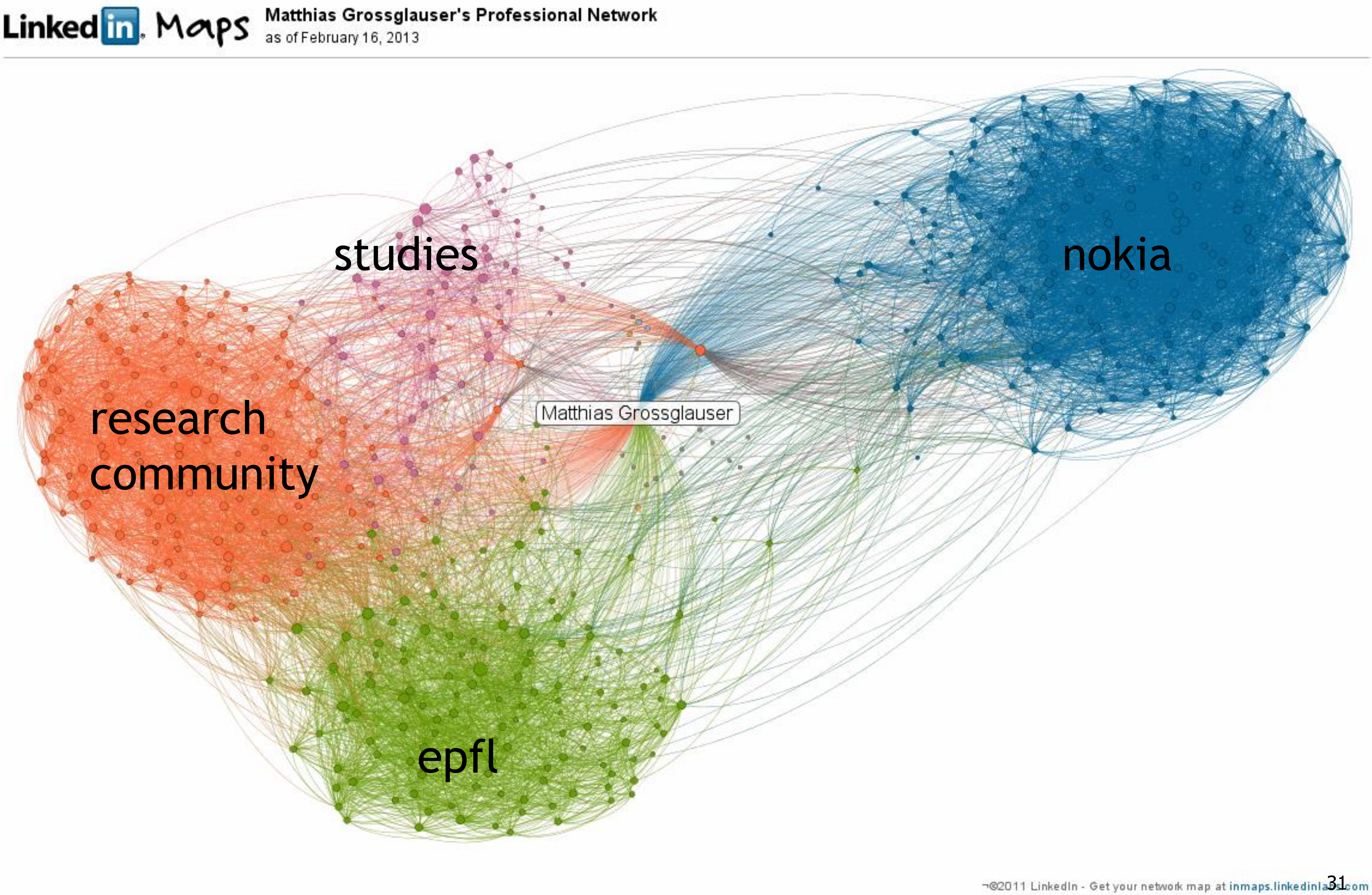


[Source: V. Blondel]

Louvain method: example



Louvain method in LinkedIn Labs



Summary

- Unsupervised techniques for grouping data
- Clusters: set of points close in distance, far from other clusters
  - Criteria: distances to center (K-means), likelihood (GMM)
  - GMM: Gaussian parameters characterize cluster
- Community: set of nodes with high edge density, low edge density to other communities
  - Criterion = modularity
- In general, no optimal solutions
  - Exponential computational cost
- Heuristics:
  - Expectation Maximization for mixture models
  - Louvain method: build hierarchy bottom-up

References

- [Ch. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006 (chapter 9)]
- [V. Blondel, lecture notes on community detection, 2013]
- [M. Newman, Networks, Oxford UP, 2010]