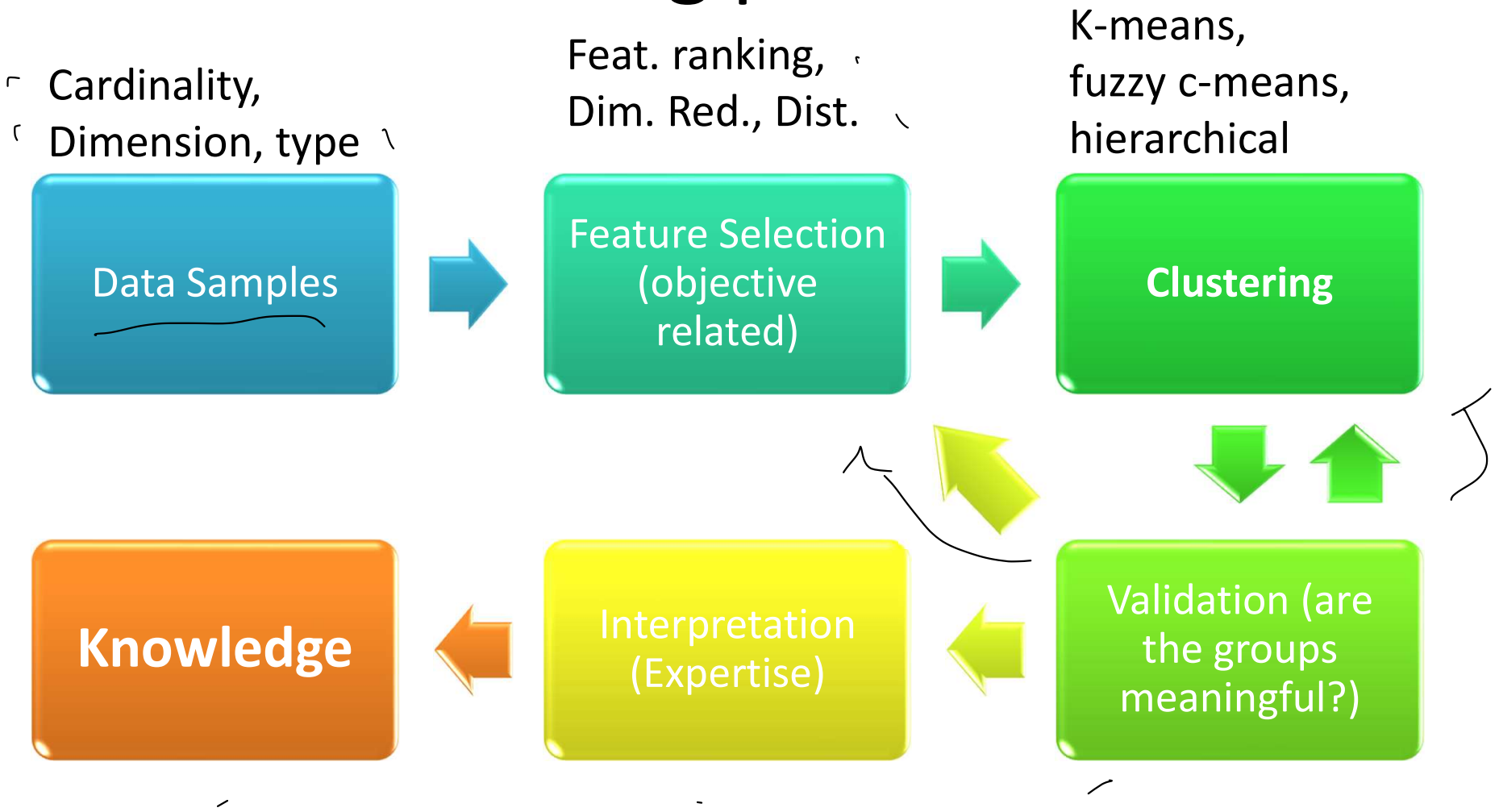
Modern Clustering algorithms

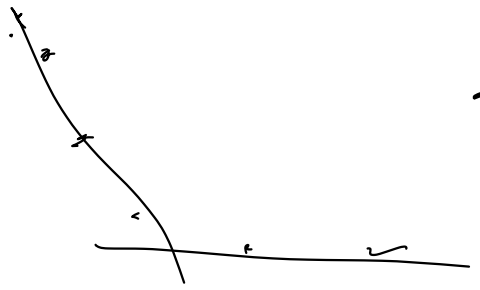
Unsupervised Machine Learning

Clustering procedure



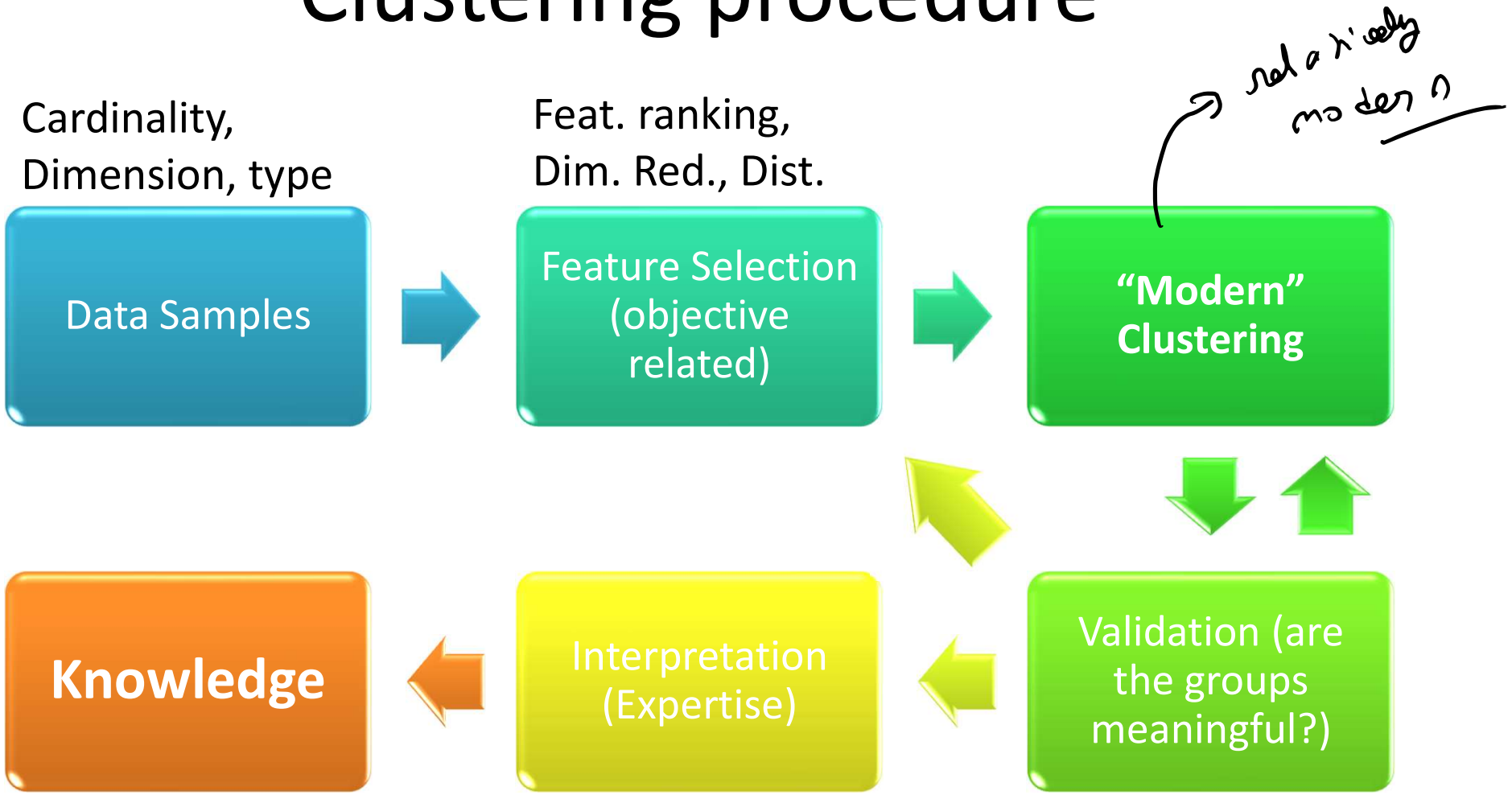
Some problems with the discussed algorithms

- K-means and fuzzy c-means are not well suited to deal with arbitrary shape (non-convex) clusters.
- Which intercluster distance shall we use for hierarchical clustering? (Euclidean choices)
- What happens with noisy data sets?
- Which should be the final number of clusters?



→ what happens if "no elbow" in scree plots?

Clustering procedure



Reviews on clustering algorithms

- There are hundreds of clustering algorithms, each of them with their advantages and withdrawals.
  - Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.
  - Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, *16*(3), 645-678.
  - Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), 651-666.
  - Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms.*Annals of Data Science*, *2*(2), 165-193.
  - Amit Saxena, *et al* (2017). A review of clustering techniques and developments,Neurocomputing, 267, 664-681.

A selected list of clustering algorithms

scikit-learn

- K-means
- Fuzzy c-means
- Agglomerative hierarchical
- Spectral Clustering
- Gaussian Mixture
- Affinity Propagation
- Density Based Algorithms:
  - DBSCAN
  - Density Peaks
  - Mean-Shift clustering

A selected list of clustering algorithms

- K-means
- Fuzzy c-means
- Agglomerative hierarchical

Classical

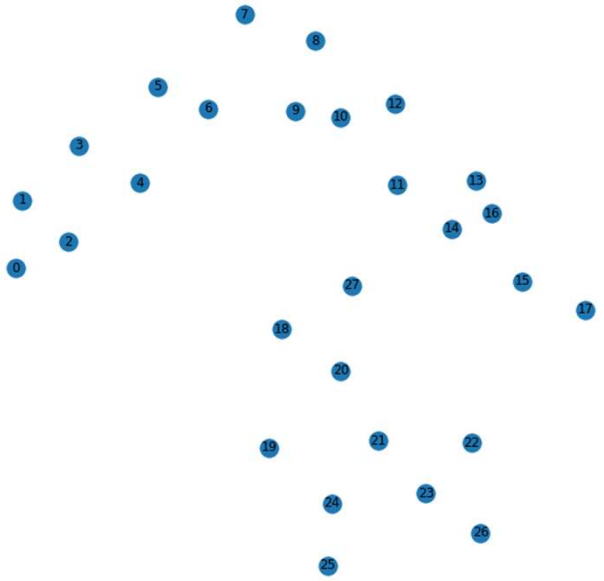
- Spectral Clustering
- Gaussian Mixture
- Affinity Propagation
- Density Based Algorithms:
  - DBSCAN
  - Density Peaks
  - Mean-Shift clustering

Modern

Spectral clustering

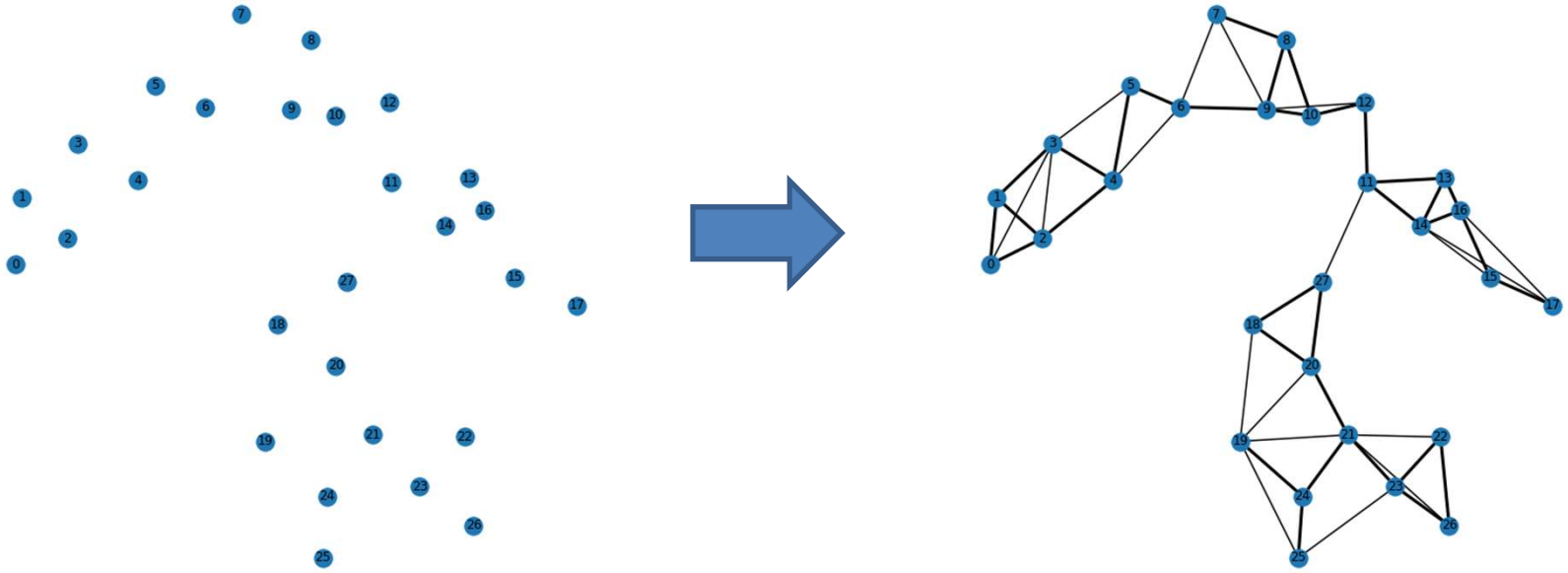
Modern clustering algorithms (I)

A different paradigm



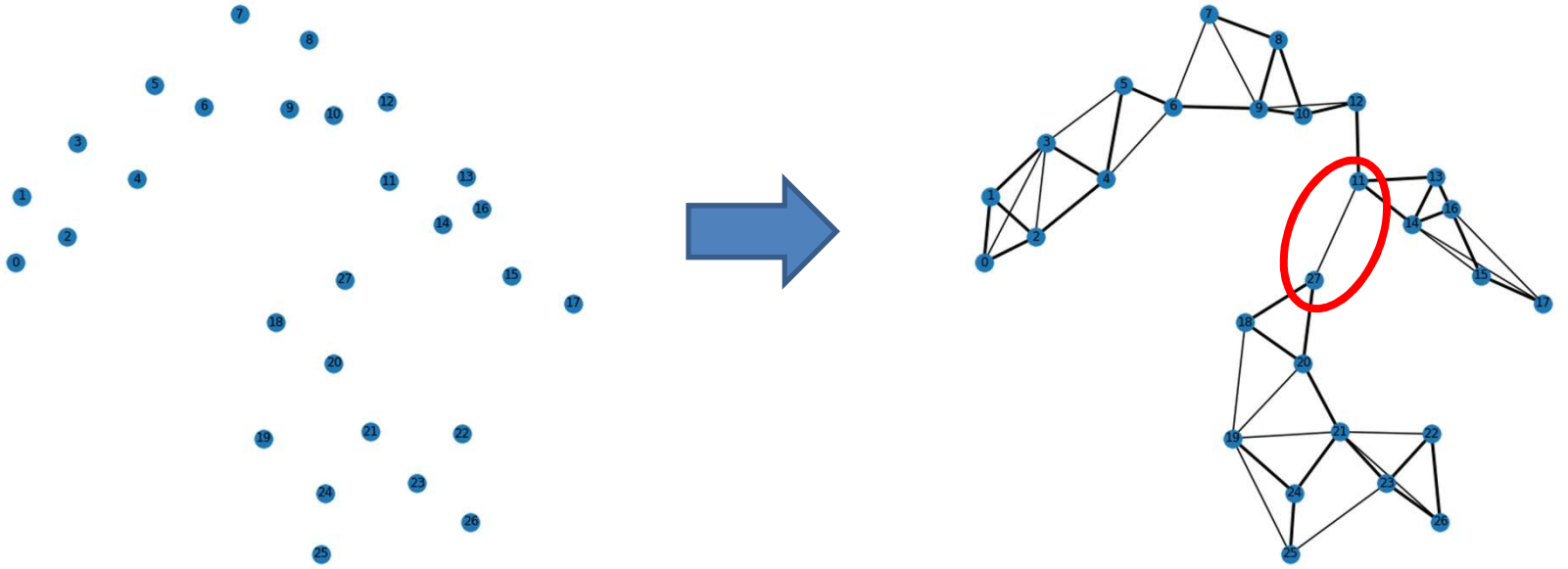
→ instead of directly applying clustering algos. on the data, we first transform it into a graph

A different paradigm



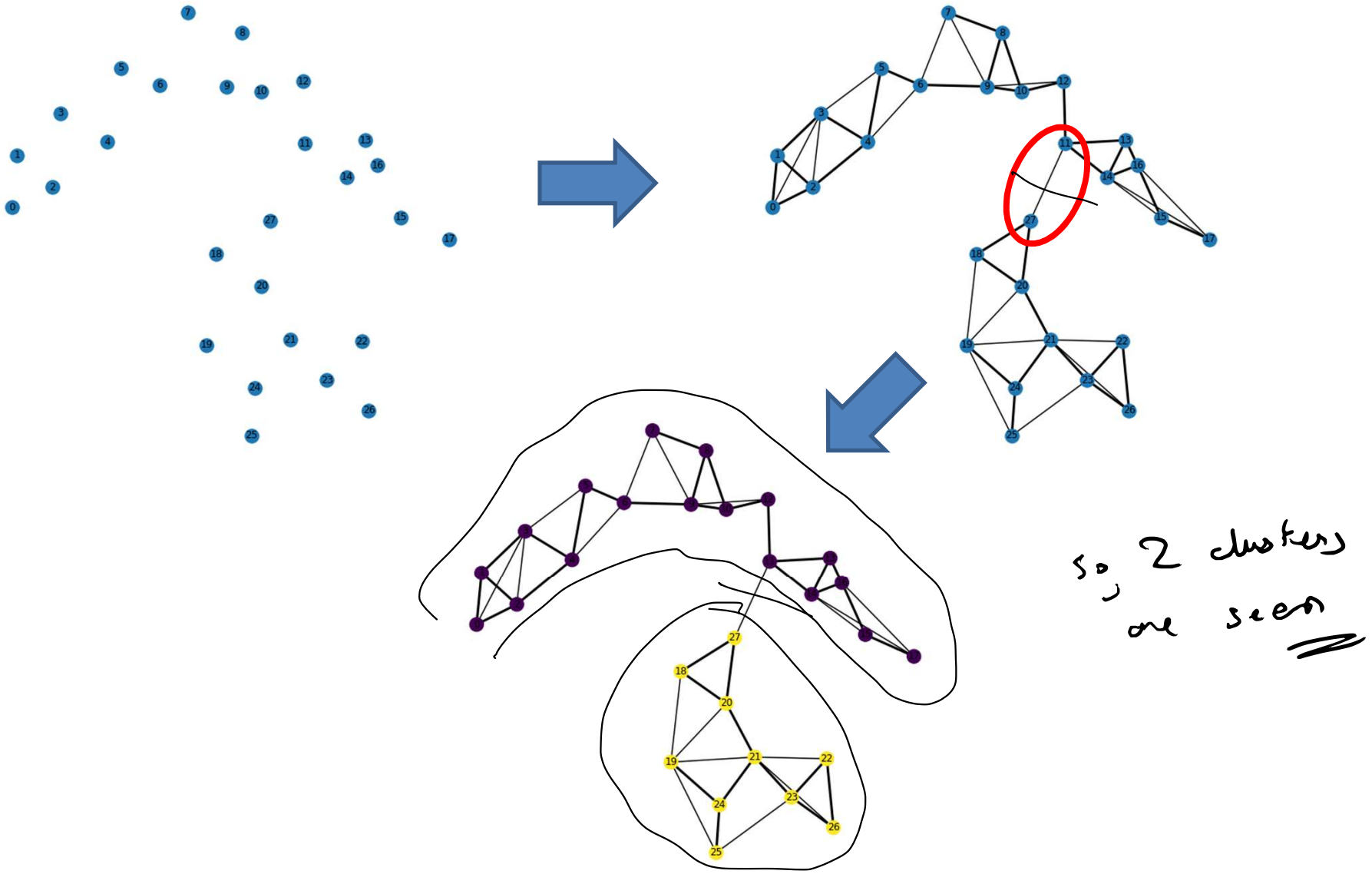
① Transform into a graph

A different paradigm

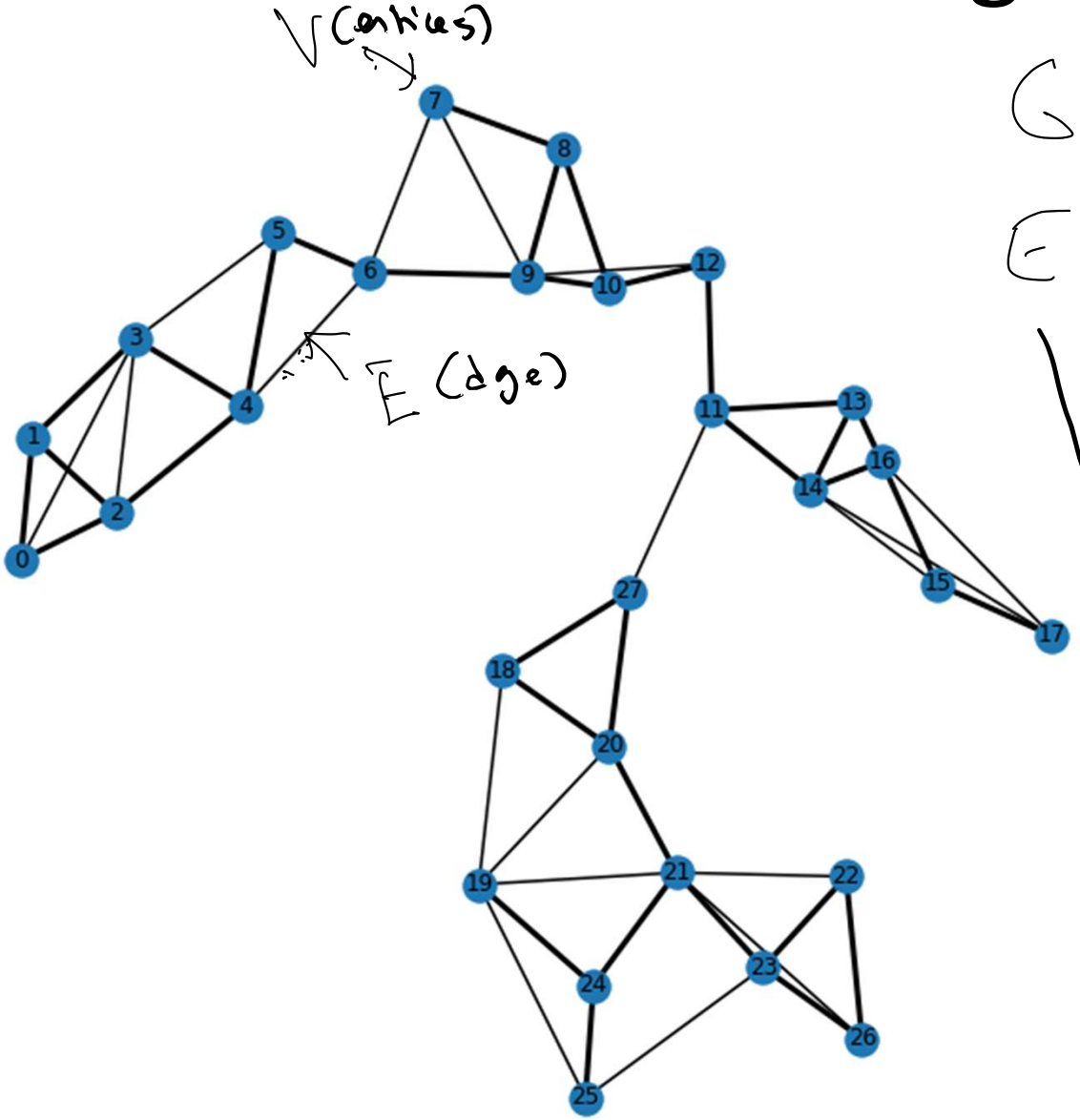


② Optimal way of ~~cut~~ cutting the graph

A different paradigm



Undirected Weighted Graph



$G(E, V)$ → graph defined by edges & vertices

$E \{(i,j), S_{ij} \geq \emptyset\}$

→ no arrows

→ no weights for edges

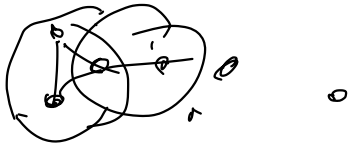
each edge is defined by the 2 vertices ((i) & (j)),

4 = weight.

if weight = 0, i & j are not connected.

Obtaining the (similarity) graph

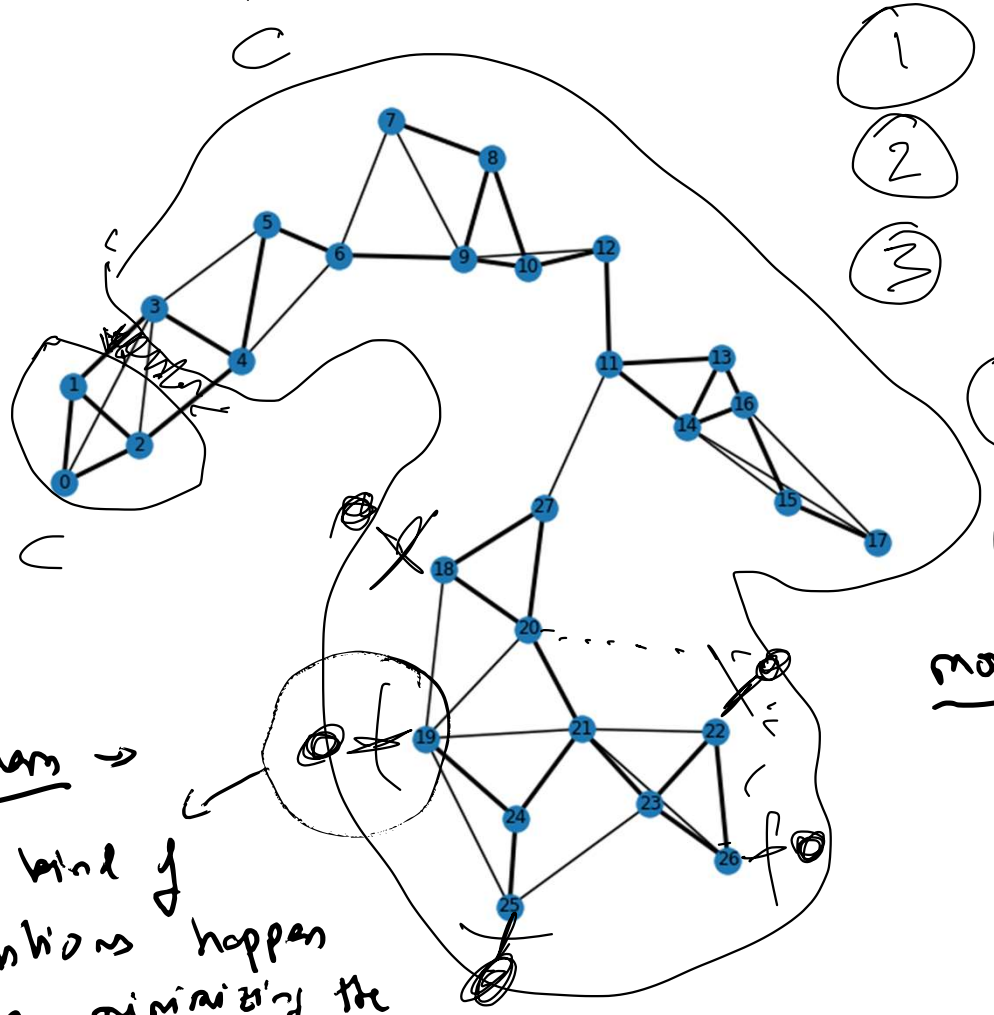
- **ε-ball graph:** Join vertices within a radius ε.



- ***k*-NN graph:** Join the k-Nearest Neighbor vertices.

- **Fully connected graphs:** Join all the vertices, for instance: $S_{ij} = e^{-\frac{d_{ij}^2}{2\sigma^2}}$

The k-way cut: a naive approach



1. Cluster C set of vertices
2. $\bar{C}$ all the vertices $\notin C$
3. $Cut(C, \bar{C}) = \sum_{i \in C} \sum_{j \in \bar{C}} S_{ij}$
4. $Cut(C_1, C_2 \dots C_K) = \frac{1}{2} \sum_{\ell} Cut(C_\ell, \bar{C}_\ell)$
5. Minimize k-way Cut

Problem →

these kind of partitions happen when minimizing the cut (5) & we get single points / noise in the result

more explanation of above approach :-

(1) → define a cluster

(2) → opposite of 'C' - all vertices not belonging to C

(3) → define the cut → all vertices belonging to C & $\bar{C}$ (complement of C) as a sum.

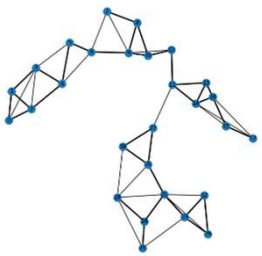
(4) → generalize (3) to 'k' clusters

→ So, to recap: the problem is we don't take size of clusters into account →

Balancing the k-way cut: Ratio Cut

- $|C_\ell|$ = Number of vertices of C_ℓ
- $\text{RatioCut}(C_1, C_2, \dots C_k) = \sum_\ell \frac{\text{Cut}(C_\ell, \bar{C}_\ell)}{|C_\ell|}$

define a different "cut" → this weights less the clusters with lower population.



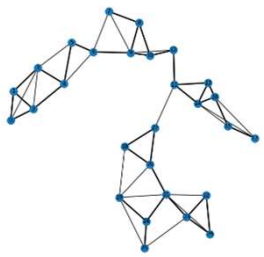
Another way of taking into account the size of the clusters: Normalized cut

→ Degree of vertex i: $d_i = \sum_j s_{ij}$

→ $Vol(C_\ell) = \sum_{i \in C_\ell} d_i$ (Vol = $\sum$ of a cluster)

→ $Ncut(C_1, C_2, \dots C_K) = \sum_\ell \frac{Cut(C_\ell, \bar{C}_\ell)}{Vol(C_\ell)}$

↳ (Normalized cut)



→ Some def needed for next part :-

The path to Spectral clustering: Matrices involved

(Similarity matrix) $\$$: Symmetric $n \times n$

$n \equiv$ # of data pts.

diagonal = 0, elements are S_{ij}. S_{ij} : graph is undirected

$$\begin{pmatrix} 0 & S_{ij} & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{pmatrix}$$

so it is symmetric

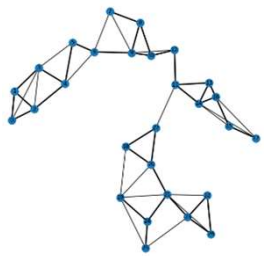
(Degree matrix) $\mathbb{D} = \{ D_{ii} = d_i \,,\, D_{ij} = \emptyset_{\forall j \neq i} \}$

diagonal terms = degree of matrix

non-diagonal = 0

(Laplacian matrix) $\mathbb{L} = \mathbb{D} - \$$

→ at the core of all spectral clustering



The path to Spectral clustering: Important properties of the Laplacian Matrix

① Symmetric $L = D - S$ ∴ L is symm.

(Sym, Sym)

② Positive semi-definite: $\lambda_1 = 0 \geq \lambda_2 \geq \lambda_3$

this is def^n :-

③ $\forall$ v as vector dim = n . We want to compute $\{v^T L v\}$

$d_i = \sum_j S_{ij}$

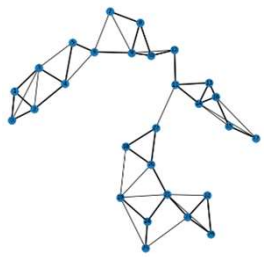
$d_j = \sum_i S_{ij}$

$$v^T L v = v^T D v - v^T S v = \sum_i d_i v_i^2 - \sum_{ij} v_i v_j S_{ij}$$

$$= \frac{1}{2}\left(\sum_i v_i^2 d_i - 2\sum_{ij} v_i v_j S_{ij} + \sum_j v_j^2 d_j\right) =$$

$$\frac{1}{2}\sum_{ij} v_i^2 S_{ij} - 2\sum_{ij} v_i v_j S_{ij} + \sum_{ij} v_j^2 S_{ij} =$$

$$\frac{1}{2}\sum_{ij} S_{ij}\left(v_i^2 - 2 v_i v_j + v_j^2\right) = \frac{1}{2}\sum_{ij} S_{ij}\left(v_i - v_j\right)^2$$



"indicator vector"
tells us if a single vertex belongs to a cluster (C) or its complement ($\bar{C}$)

Let's define a vector f with following properties

The RatioCut for k=2

(a) Minimize $\frac{Cut(C,\bar{C})}{|C|}$

(b) f

$$\begin{cases} f_i = \sqrt{\frac{|\bar{C}|}{|C|}} & \text{if } i \in C \\ f_i = -\sqrt{\frac{|C|}{|\bar{C}|}} & \text{if } i \in \bar{C} \end{cases}$$

(c) $f^T L f = \frac{1}{2}\sum_{ij} S_{ij}(f_i - f_j)^2 = \frac{1}{2}\sum_{\substack{i\in C \\ j\in \bar{C}}} S_{ij}\left(\sqrt{\frac{|\bar{C}|}{|C|}} + \sqrt{\frac{|C|}{|\bar{C}|}}\right)^2$

$+\frac{1}{2}\sum_{\substack{i\in \bar{C} \\ j\in C}} S_{ij}\left(-\sqrt{\frac{|C|}{|\bar{C}|}} - \sqrt{\frac{|\bar{C}|}{|C|}}\right) =$

if i & j belong to diff clusters

(d) $cut(C,\bar{C}) \cdot \left(\frac{|\bar{C}|}{|C|} + \frac{|C|}{|\bar{C}|} + 2\right) = n\, RatioCut(C,\bar{C})$

C can be derived from some algebra

→ when $\{i \in C, j \in C\}$, & $\{i \in \bar{C}, j \in \bar{C}\}$, the terms cancel out

The RatioCut for k=2

$\Rightarrow$ minimize $\frac{Cut(C, \bar{C})}{|C|}$ equivalent

minimize $f^T L f$, given: $\|f\| = \sqrt{n}$ & $f \perp \mathbb{1}$

condition (1) condition (2)

condition (2): this avoids the trivial sol^n, where all values of f are the same.

The RatioCut for arbitrary k

→ k indicator vectors, H $\quad i = 1 \dots n$ (define "h")

① $H_{ij} = \dfrac{1}{\sqrt{|C_j|}}$ if $i \in C_j$ $\quad j = 1 \dots k$ → properties of "h"

$H_{ij} = 0$ otherwise

→ (h_p) is the indicator vector.

② $H^T H = I$ $\quad H_{(:,p)} = h_p$

$h_p^T L h_p = \dfrac{cut(C_p, \bar{C}_p)}{|C_p|} = (H^T L H)_{pp}$ → (product of these matrices is the diagonal (p,p))

$\therefore$ RatioCut$(C_1, \dots C_k) = Tr(H^T L H)$ → Trace = RatioCut

given :- $H^T H = I$

(similar to what we did for $k = 2$)

The RatioCut for arbitrary k

→ for indicator vectors, we impose certain values like

$$h_{ij} = \frac{1}{\sqrt{|C_j|}}$$

& we don't know how to minimize that

Solving the problem: Relaxation

- Relaxation: Remove the constraint of discrete optimization and accept that the indicator matrix (H) has real values.
- The minimization becomes tractable using Lagrange multipliers. It becomes an eigenvalue-eigenvector problem (see PCA derivation). → $LH = \lambda H$ (solve this eq.)
- Once it is solved, we need to discretize → k-means using the eigenvectors as coordinates.
- There's no warranty that this solution is similar to the ~~unrelax~~ non-relaxed one!!!

The Unnormalized spectral clustering algorithm

1. Construct the similarity graph *
2. Compute the Laplacian $L = D - S$
3. Compute the first k eigenvectors $Lh = \lambda h$
4. Use the coordinates as input for the k-means algorithm and obtain the clusters

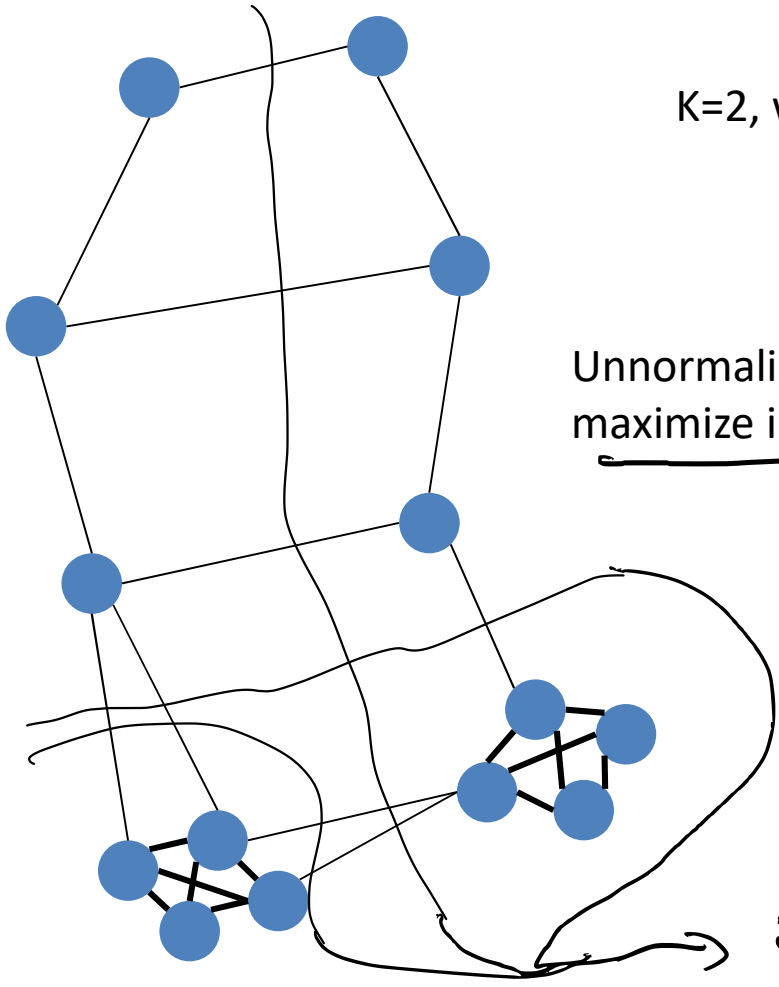
$Lh = \lambda h \Rightarrow H \begin{pmatrix} H_{11} \dots H_{1k} \\ \vdots \quad\quad \vdots \\ H_{n1} \dots H_{nk} \end{pmatrix}$ $\vec{x}_i = (H_{i1}, \dots, H_{ik})$

$i = 1 \dots n$

k-means

(new set of coords. for k-means)

A problem related to RatioCut



K=2, where should we cut?

Unnormalized spectral clustering does not maximize intracluster similarities.

All of these 3 are possible. Which do we do?

→ There's no simple sol^n ∵ ratiocut is not well-defined for certain problems.

tries to maximize intra cluster similarities

The Normalized spectral clustering

$$\min \; Ncut = \frac{Cut(C,\bar{C})}{Vol(C)} \qquad Ncut(C_1 \dots C_K) = \frac{1}{2}\sum \frac{Cut(C_i,\bar{C}_i)}{Vol(C_i)}$$

(a) $\frac{1}{Vol(C_j)} = H_{ij}$ $i \in C_j$; otherwise $H_{ij} = \phi$

$$\min \left(Tr\left(H^T L H\right)\right) \text{ given } \left(H^T D H\right) = I \quad *$$

(minimizing Ncut is same as minimizing the trace)

(b) $T = D^{1/2} H$

$$\min \left(Tr\left(H^T L H\right)\right) = \min \left(Tr\left(T^T \underbrace{D^{-1/2} L D^{-1/2}}_{L_{sym}} T\right)\right)$$

cond :- $\left(T^T T\right) = I$

:: Find problem :- $\min \left(Tr\left(T^T L_{sym} T\right)\right)$ given $\left(T^T T\right) = I$

→ Now it is again an eigval problem, & we do same as unnormalized algo, (replace "L" with "L_{sym}")

* for Ratio Cut the condition was $H^T H = I$

Spectral Clustering

- Closely related with kernel k-means, under certain conditions, they optimize the same loss.
- With a different transformation, the Laplacian can be transformed in a random walk matrix and related with a Markov state model analysis.
- The value of k can be inferred from a gap in the spectrum of eigenvalues.

Spectral Clustering problems

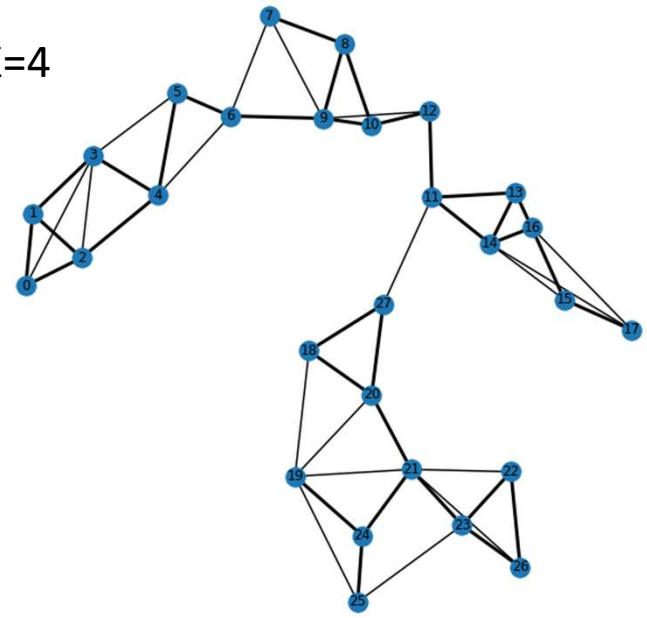
- How to construct the similarity matrix?

K-NN graph.

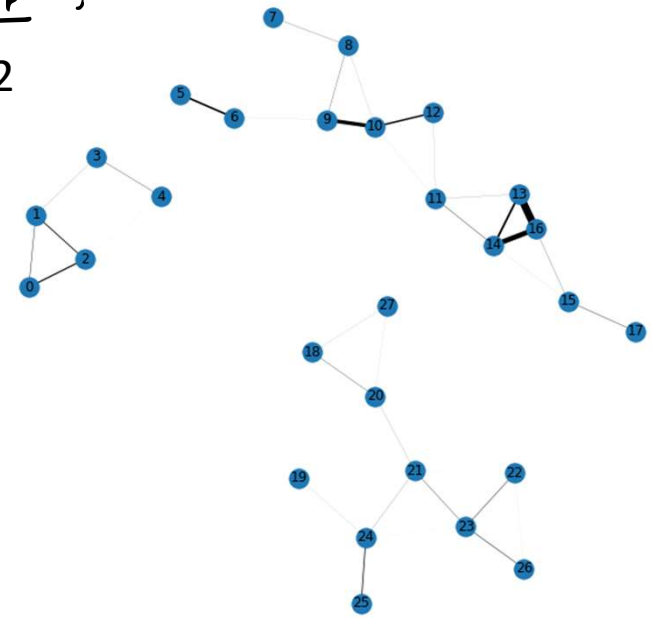
fully connected $S_{ij} = e^{\frac{-d_{ij}}{2\sigma}}$

graph :-

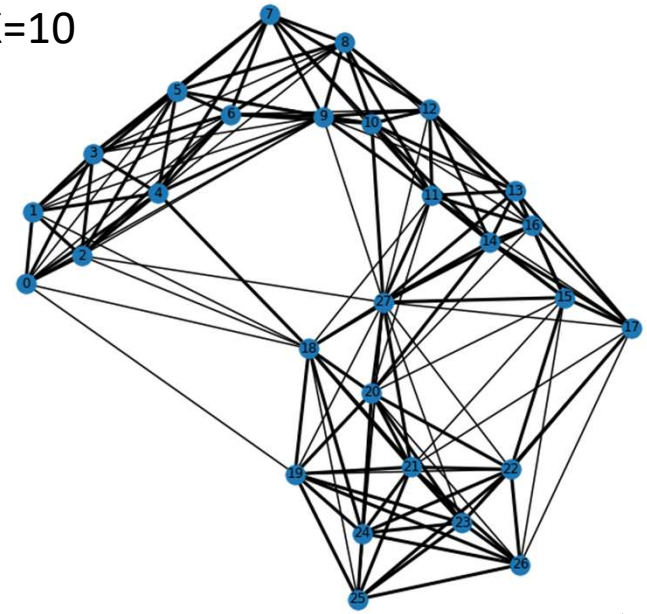
K=4



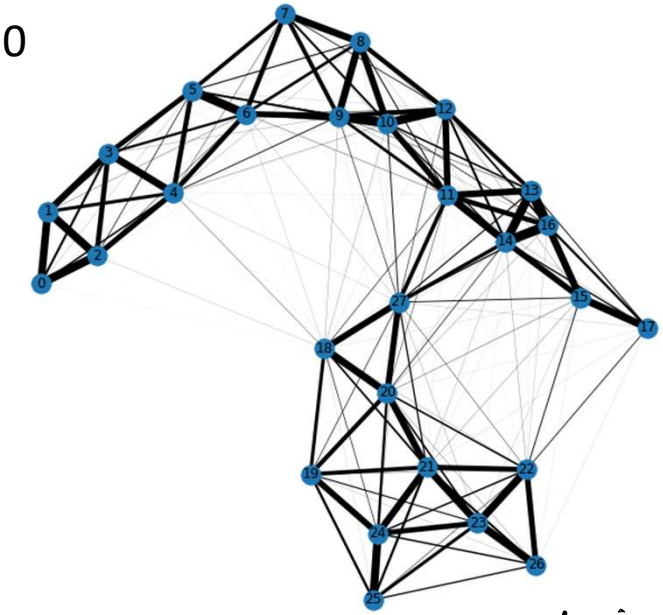
σ=3.2



K=10



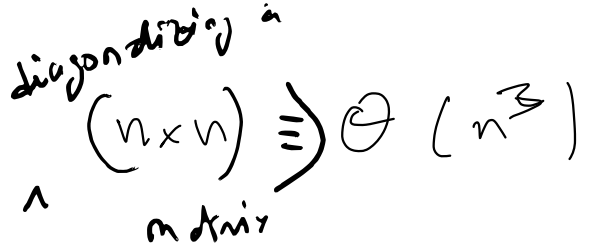
σ=1.0

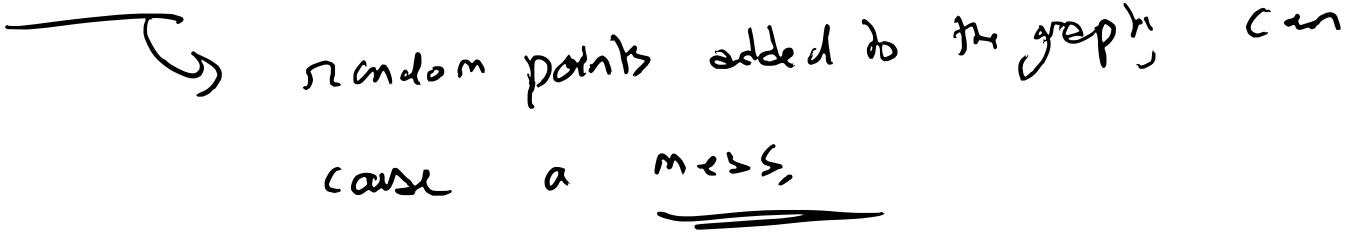


(graphs are completely diff., depending on choice of params etc.)

Spectral Clustering problems

- How to construct the similarity matrix?
- How to decide the number of clusters in absence of gap?
- Computational Scaling
- Noise sensitivity





Expectation-Maximization clustering

Modern clustering algorithms (II)

Model based clustering

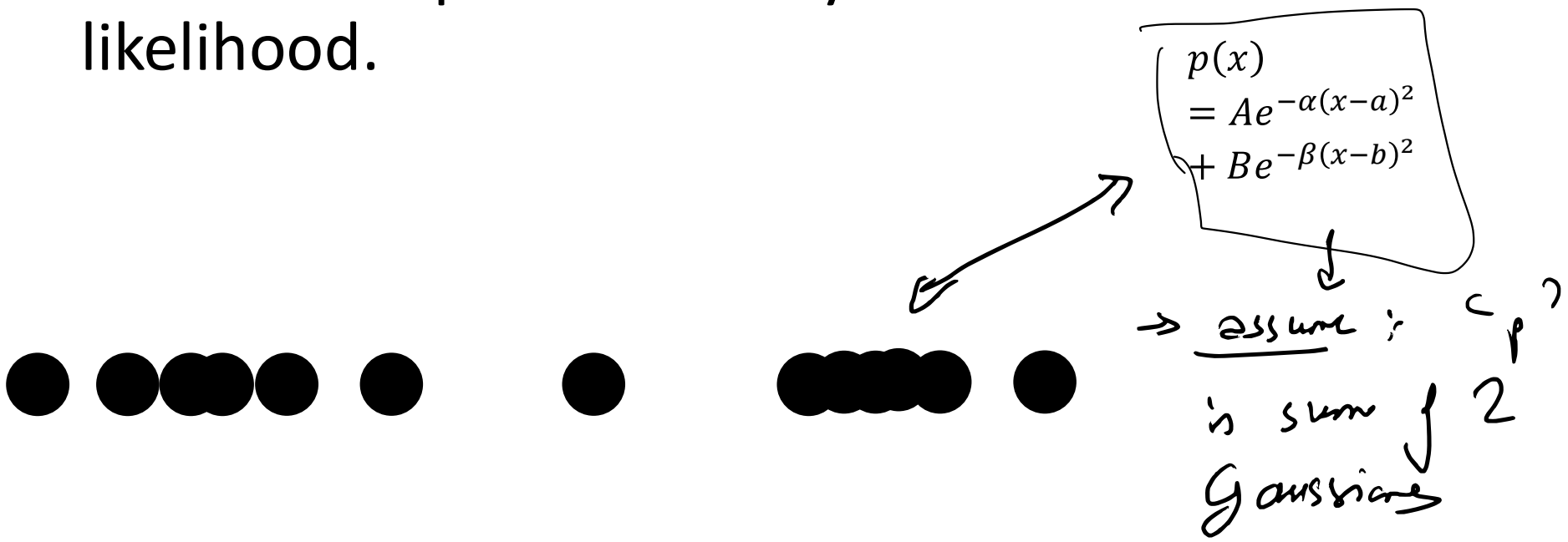
- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.

Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.

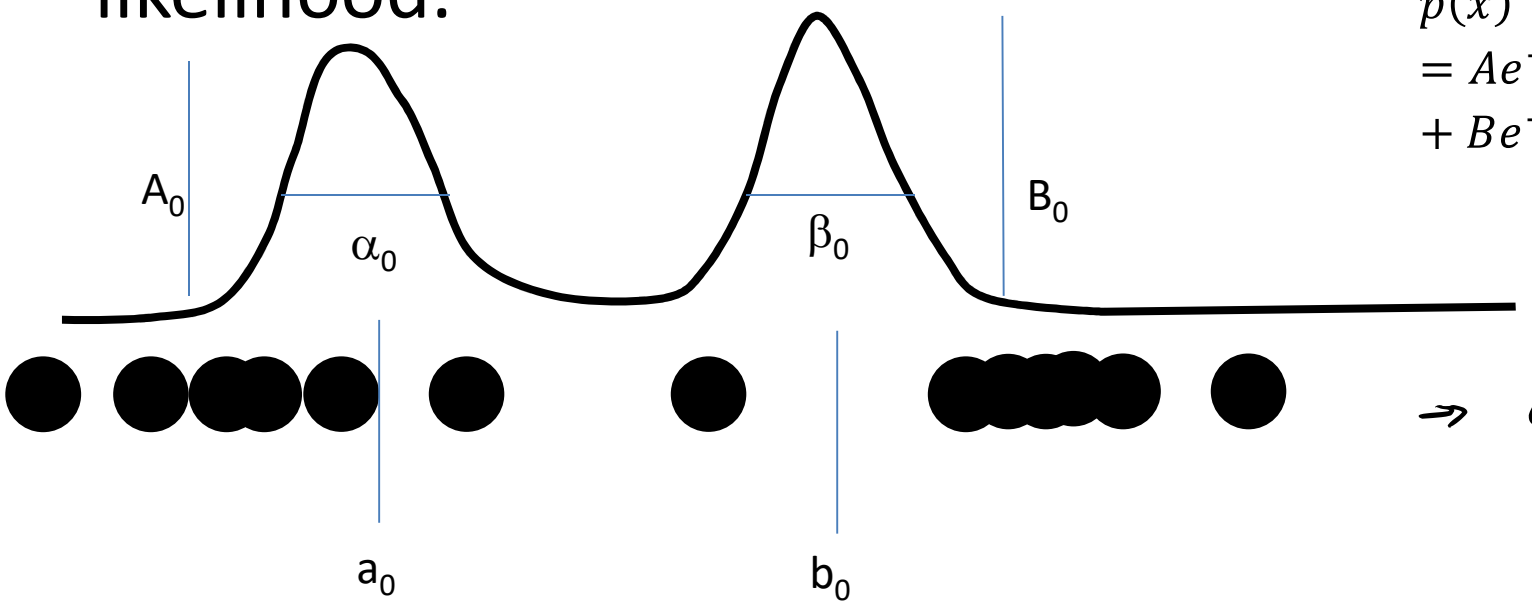
Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.



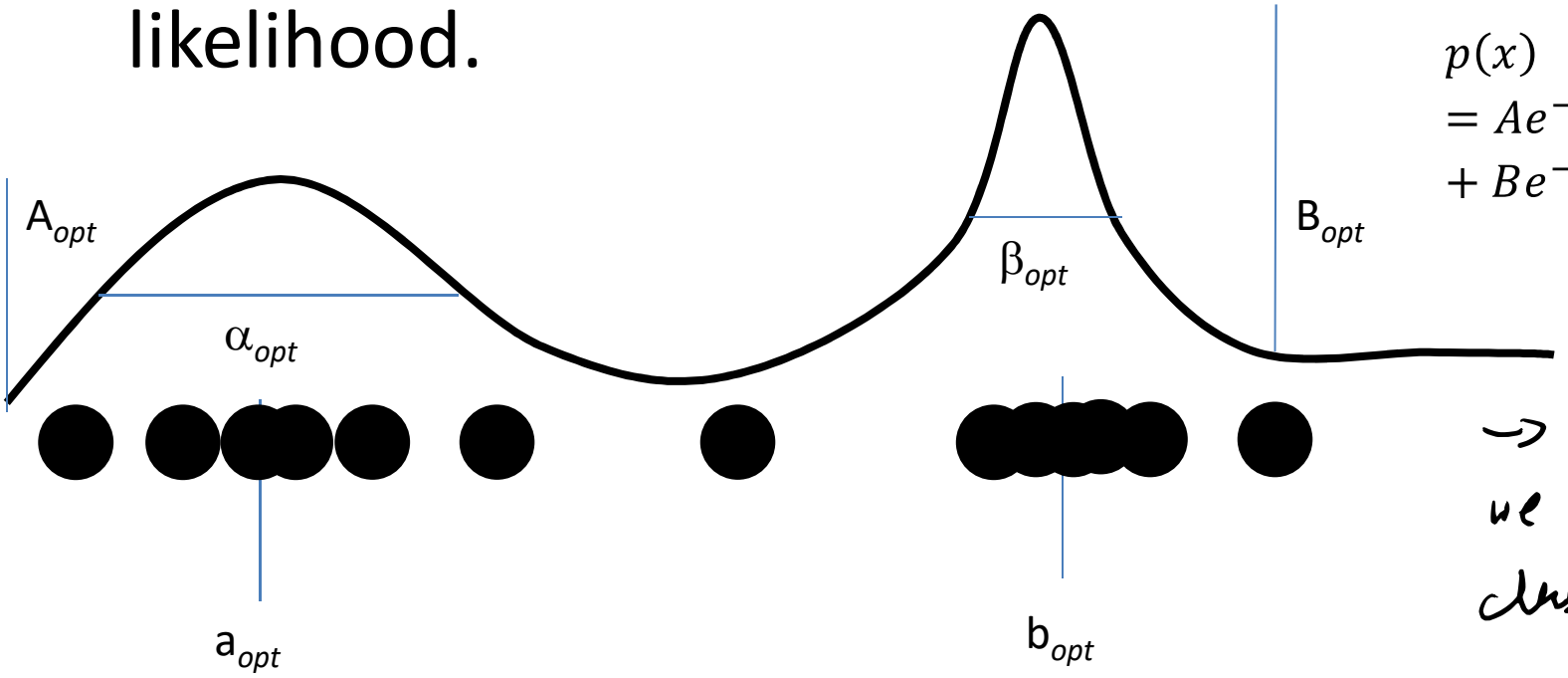
Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.



Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.

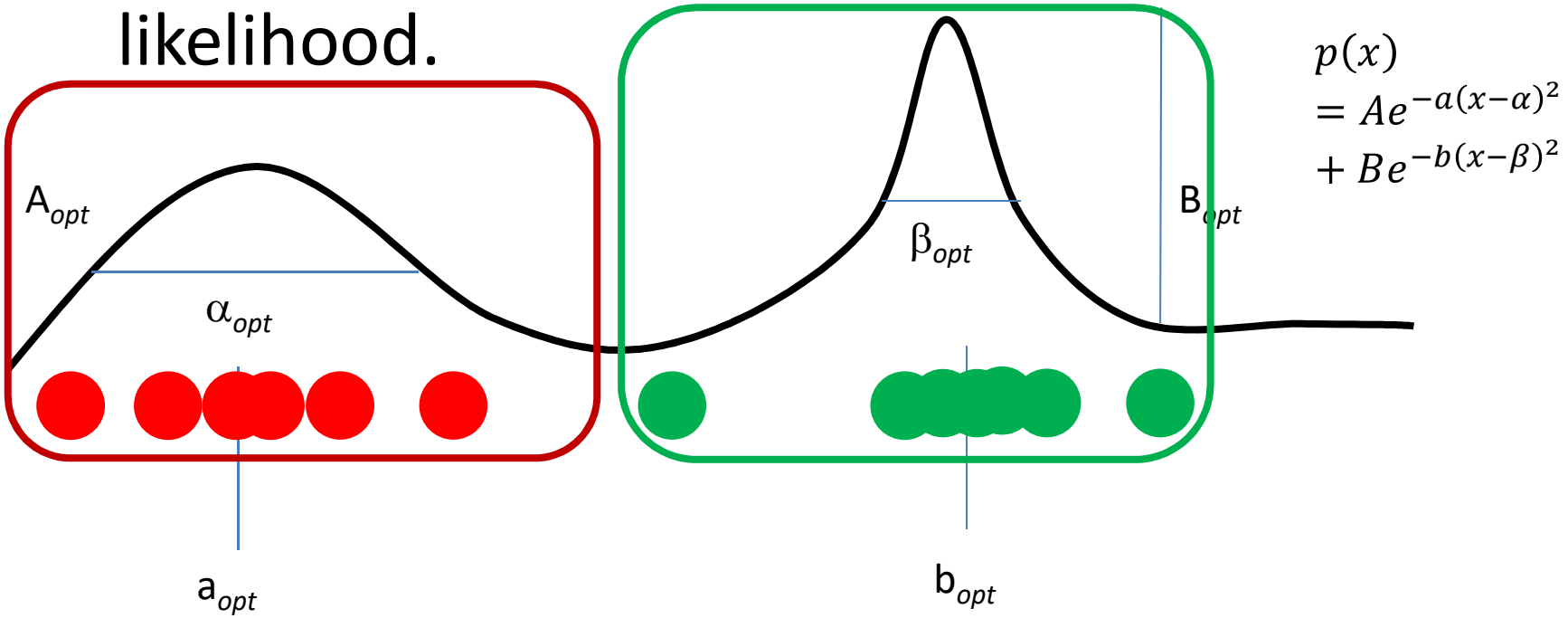


→ after optimization we perform a cluster assignment

∴ EM allows us to estimate the params when they can't be estimated analytically.

Model based clustering

- Consider your data as a set of realizations of an underlying probability function $p(x)$.
- Assume the functional form of $p(x)$ and estimate its parameters by maximum likelihood.



Expectation-maximization algorithm

- Iterative procedure to compute the ***Maximum Likelihood (ML)*** estimate – even in the presence of missing or hidden data
- **EM** consists of two steps:
  - **Expectation step:** the (missing) data are estimated given the observed data and current estimates of model parameters
  - **Maximization step:** The likelihood function is maximized under the assumption that the (missing) data are known , (i.e., we recompute the params given the prob. distrib.)

→ Explained in next slides

EM algorithm for mixture of Gaussians

- What is a mixture of **K** Gaussians?

$$p(x) = \sum_{k=1}^{K} \pi_k F(x \mid \Theta_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1$$

and **F(x|Θ)** is the Gaussian distribution with parameters **Θ = {μ,Σ}**

EM algorithm for mixture of Gaussians

- If all points **x∈X** are mixtures of **K** Gaussians then

$$\mathcal{L}(X) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k F(x_i \mid \Theta_k)$$

- **Goal:** Find $\pi_1, \ldots, \pi_k$ and $\Theta_1, \ldots, \Theta_k$ such that $\mathcal{L}(X)$ is maximized
- Or, $\ln(\mathcal{L}(X))$ is maximized: (equivalent)

$$L(\Theta) = \sum_{i=1}^{n} \ln\left\{\sum_{k=1}^{K} \pi_k F(x_i \mid \Theta_k)\right\}$$

likelihood is prob. of prob. points = product of the Gaussians

$\mathcal{L}(x)$

$\mathcal{L}(x)$ = likelihood

loss func^2

Mixtures of Gaussians -- notes

- Every point x_i is ***probabilistically*** assigned (generated) to (by) the **k**-th Gaussian

- Probability that point x_i is generated by the **k**-th Gaussian is

$$w_{ik} = \frac{\pi_k F(x_i \mid \Theta_k)}{\sum_{j=1}^{K} \pi_j F(x_i \mid \Theta_j)}$$

Mixtures of Gaussians -- notes

- Every Gaussian (cluster) C_k has an effective number of points assigned to it N_k

$$N_k = \sum_{i=1}^{n} w_{ik}$$

- With mean

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} x_i$$

- And variance

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} (x_i - \mu_k) x_i (x_i - \mu_k)^T$$

EM for Gaussian Mixtures

- Initialize the means μ_k, variances Σ_k ($\Theta_k=(\mu_k,\Sigma_k)$) and mixing coefficients π_k, and evaluate the initial value of the loglikelihood

(can be random or from k means)

- **Expectation step**: Evaluate weights

$$w_{ik} = \frac{\pi_k F(x_i \mid \Theta_k)}{\sum_{j=1}^{K} \pi_j F(x_i \mid \Theta_j)}$$

→ how likely is it that each point belongs to a given cluster.

EM for Gaussian Mixtures

- **Maximization step**: Re-evaluate parameters

(Compute new params for the Gaussians)

$$\mu_k^{new} = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} x_i$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{i=1}^{n} w_{ik} \left(x_i - \mu_k^{new}\right) x_i \left(x_i - \mu_k^{new}\right)^T$$

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

- Evaluate $L(\Theta^{new})$ and stop if converged

EM characteristics

- Notice the similarity between EM for Normal mixtures and K-means: The expectation step is the assignment, while the maximization step is the update of the centers.
- Contrary to what happens in the case of fuzzy clustering, the weights have a real probabilistic interpretation.
- If the model (k, functional form) is not realistic, neither will the results.
- It is not guaranteed to reach the global optimum.
- K can be inferred with (bayesian) model selection (Dirichlet process).

diff. forms can be used → gaussian exp etc.

"prior" is imposed to after "K".

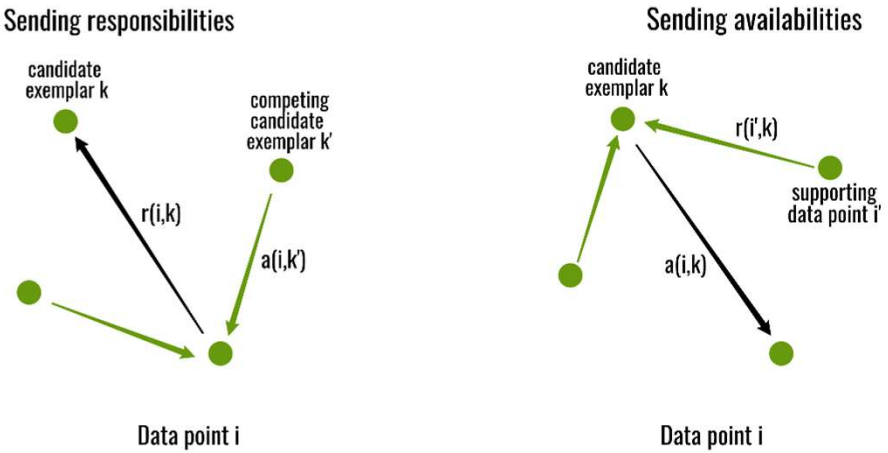
so we need to repeat many times like in kMeans.

Affinity propagation

Modern clustering algorithms (III)

Affinity Propagation. Concepts

- AffinityPropagation creates clusters by sending messages between pairs of samples until convergence.
- Two categories of messages:
  - **Responsibility**
  - **Availability**



Affinity Propagation. Responsibility

Accumulated evidence that sample k should be the exemplar for sample i.

$$r(i,k) \leftarrow S_{ik} - \max_{k' \neq k}[S_{ik'} + a(i,k')]$$

data points that are representative of the clusters

→ compute evidence that a point is exemplar for another.

(This is the formula)

Affinity Propagation. Availability

Accumulated evidence that sample i should choose sample k to be its exemplar

$$a(i,k) \leftarrow \min\left[0, r(k,k) + \sum_{i' \neq \{k,i\}} \max\{0, r(i',k)\}\right]$$

$$a(k,k) \leftarrow \sum_{i' \neq \{k\}} \max\{0, r(i',k)\}$$

(for itself)

Note that it considers the values for all other samples

Affinity Propagation. The algorithm

- S_{ik} is the similarity, usually is taken $S_{ik} = -\|x_i - x_k\|^2$
- S_{kk}, known as preference, plays an important role. A value close to the minimum possible similarity produces fewer classes, while a value close to or larger than the maximum possible similarity produces many classes. It is typically initialized to the median similarity of all pairs of inputs.

Affinity Propagation. The algorithm

- The algorithm starts by setting all the values of r and a to zero. And iterates by first updating responsibilities and then availabilities until a number of iterations is reached.
- So, after the first iteration:
  - $r(i,k) \leftarrow S_{ik} - \max_{k' \neq k}[S_{ik'}]$
  - $a(i,k) \leftarrow \min\left[0, r(k,k) + \sum_{i' \neq \{k,i\}} \max\{0, r(i',k)\}\right]$
- $r(i,k)_t \leftarrow S_{ik} - \max_{k' \neq k}[S_{ik'} + a(i,k')]$ decreases when the availability for other points increases!

Affinity Propagation. The algorithm

- The exemplars are extracted from the final matrices as those whose 'responsibility + availability' for themselves is positive ($r(k,k) + a(k,k) > 0$).
- The element i is assigned to the same cluster as the exemplar l for which $r(i,l)$ is maximum.

Affinity Propagation. The algorithm

- The algorithm described in this way is usually not stable numerically.
- A dumping factor is introduced to smooth the transitions:

$$\begin{cases} r(i,k)_{t+1} = \lambda r(i,k)_t + (1-\lambda)\, r(i,k)_{t+1} \\ a(i,k)_{t+1} = \lambda a(i,k)_t + (1-\lambda)\, a(i,k)_{t+1} \end{cases}$$

→ once we re-compute (r) & (a), we "dump" the change by a factor so that the transitions are smoother.

Affinity Propagation. Prices and caveats

- Two parameters: Preference and dumping factor.
- The preference parameter strongly affects the results and setting it is not so clear.
- The method is quite weight computationally speaking, with a complexity that is $\mathcal{O}(TN^2)$.
- It trends to generate convex clusters.

Next lecture…

Metrics for validation

- K-means
- Fuzzy c-means
- Agglomerative hierarchical
- Spectral Clustering
- Gaussian Mixture (Expectation - Maximization)
- Affinity Propagation
- Density Based Algorithms:
  - DBSCAN
  - Density Peaks
  - Mean-Shift clustering

(last bc)