HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

UNSUPERVISED LEARNING – CLUSTERING

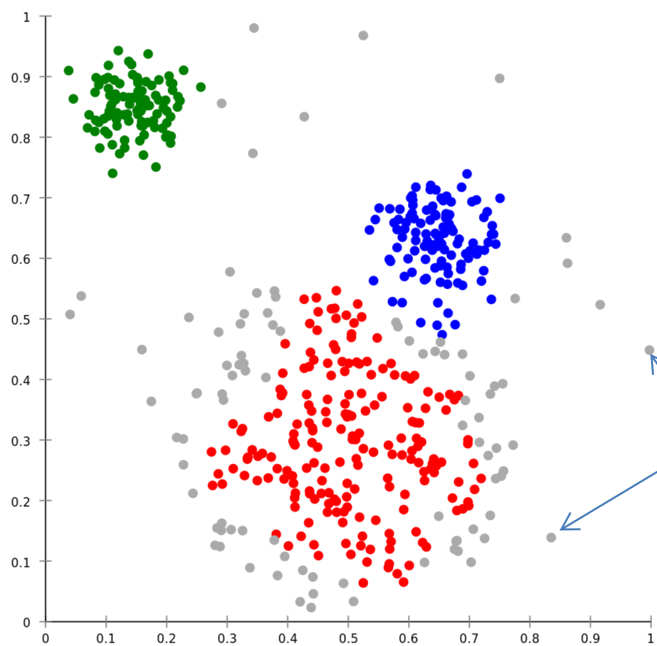
Outline

- Clustering overview
- Internal and external clustering criteria
- Impossibility Theorem for clustering
- Hierarchical clustering
- Single-link, complete-link heuristics
- Partitional/flat clustering algorithms

Clustering overview

- Why clustering?
  - … no labels available ➔ group by similarity (**unsupervised learning** scenario)
  - … to hopefully detect **“intrinsic” structure** in the data (“natural clusters”)
  - … to hopefully better understand/analyze the data through **reduction to important patterns**
  - … to detect **outliers**

Source: Wikipedia



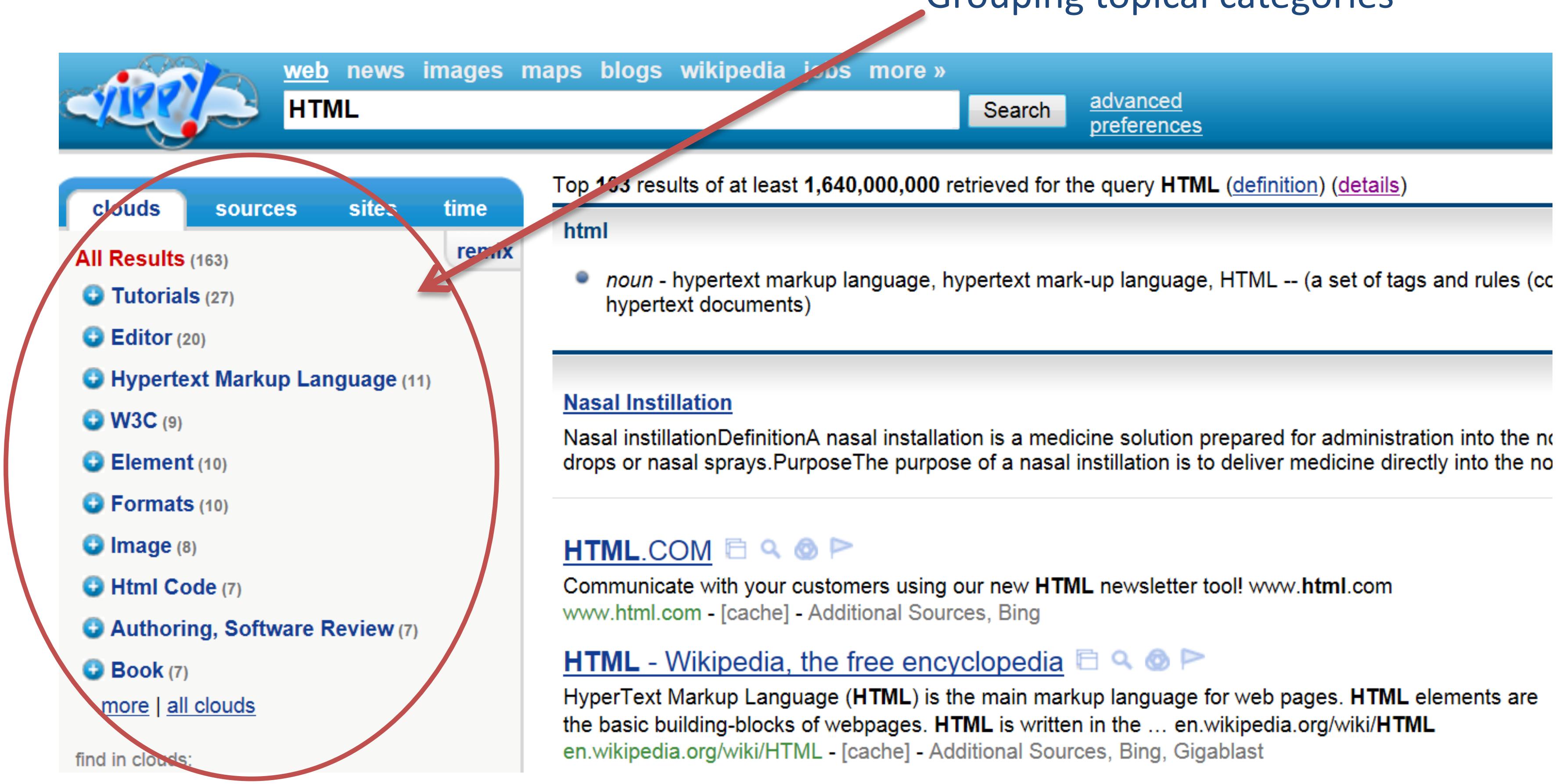
Clustering by density

outliers

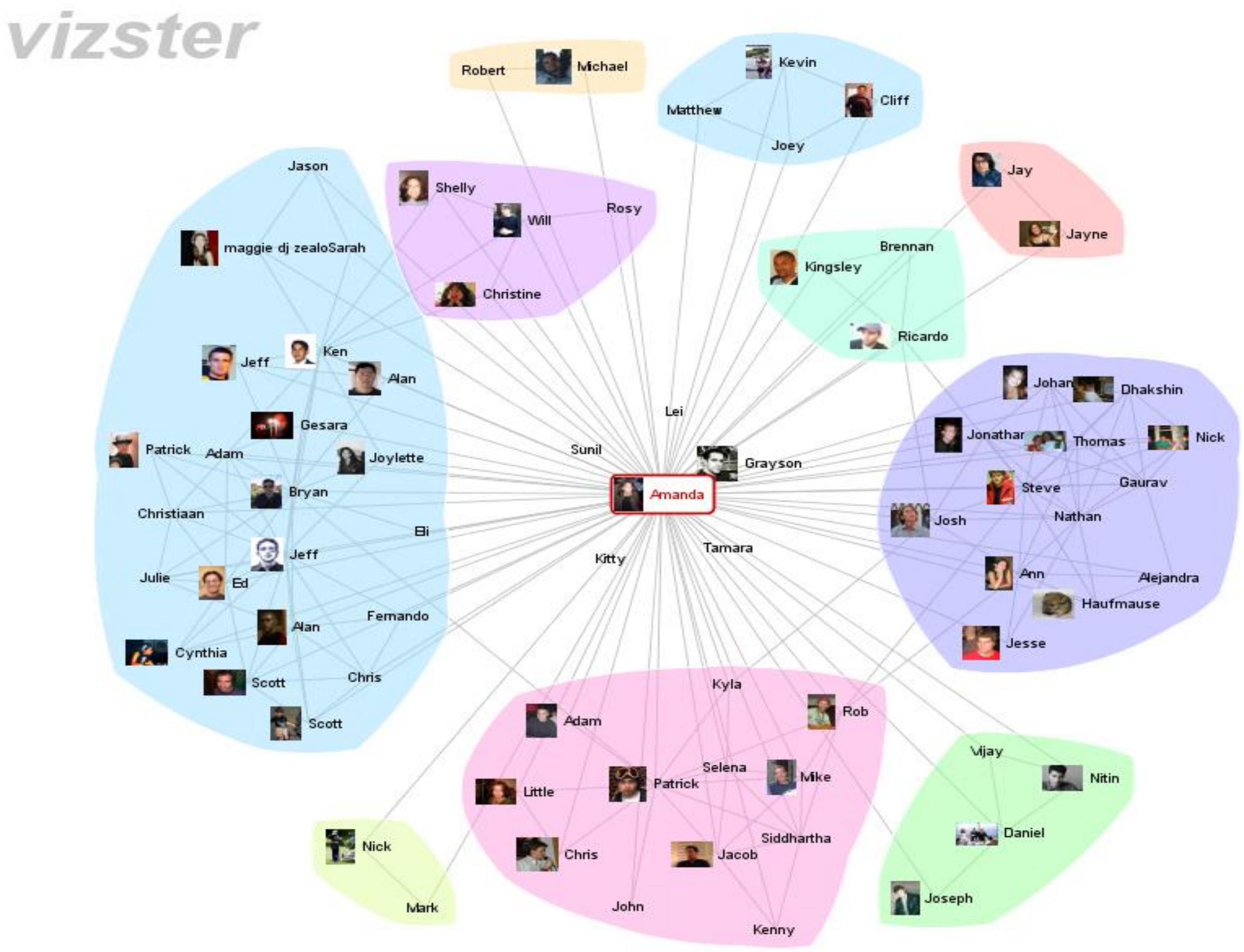
Clustering search results

Source: http://yippy.com/

Grouping topical categories



Finding communities in social networks



http://hci.stanford.edu/jheer/projects/vizster/

Hierarchical vs. partitional/flat clustering

- Hierarchical
  - Detailed, insightful hierarchies/dendrograms
  - Simple but expensive algorithms
    - Top-down (divisive)
    - Bottom-up (agglomerative)
- Partitional/flat
  - Coarse data overview
  - Level of detail depends on number of clusters
  - Relatively efficient algorithms
    - K-means
    - EM on mixture models
    - …

From metric distances to similarities

- Similarity is typically based on a **metric distance**:
  A data space M with distance function $d: M \times M \rightarrow \mathbb{R}$ is called a **metric space** if for any $x, y, z \in M$:
  1. $d(x, y) = 0$ iff $x = y$
  2. $d(x, y) = d(y, x)$ (symmetry)
  3. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

In a metric space M with distance function d the similarity between any $x, y \in M$ can be defined as $sim(x, y) := \frac{1}{1+d(x,y)}$ or $sim(x, y) := \frac{1}{e^{d(x,y)}}$

Metric distance	Definition
Euclidean	$\|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_i (x_i - y_i)^2}$
Manhattan	$\|\mathbf{x} - \mathbf{y}\|_1 = \sum_i \lvert x_i - y_i \rvert$
Maximum	$\|\mathbf{x} - \mathbf{y}\|_\infty = \max_i \lvert x_i - y_i \rvert$
Mahalanobis	$d_{maha}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i \left(\frac{x_i - y_i}{\sigma_i}\right)^2}$ (for normally distributed data)

Other popular similarity and distance measures

- Pearson correlation

$$\rho(\mathbf{x},\mathbf{y}) = \frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_i (x_i-\bar{x})^2}\sqrt{\sum_i (y_i-\bar{y})^2}}$$ (similarity measure)

$$d_\rho(\mathbf{x},\mathbf{y}) = \frac{1-\rho(\mathbf{x},\mathbf{y})}{2}$$ (distance metric)

- Cosine similarity

$$csim(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$

$$d_{csim}(\mathbf{x},\mathbf{y}) = 1 - csim(\mathbf{x},\mathbf{y})$$ (distance measure)

- Jaccard similarity

$$J(\mathbf{c},\mathbf{c}') = \frac{|\mathbf{c}\cap\mathbf{c}'|}{|\mathbf{c}\cup\mathbf{c}'|}$$ (similarity measure)

$$d_J(\mathbf{c},\mathbf{c}') = 1 - J(\mathbf{c},\mathbf{c}')$$ (distance metric)

Internal clustering criteria

- General goal: For objects $\mathbf{x}_1, \dots, \mathbf{x}_n$ with pair-wise similarities, construct $k \leq n$ clusters $\mathbf{c}_1, \dots, \mathbf{c}_k$ such that

  - **Intra-cluster similarity** is high

$$\frac{1}{k}\sum_i \left(\frac{1}{|\mathbf{c}_i|(|\mathbf{c}_i|-1)}\sum_{\mathbf{x},\mathbf{x}'\in\mathbf{c}_i} sim(\mathbf{x},\mathbf{x}')\right) \text{ or } \frac{1}{k}\sum_i \left(\frac{1}{|\mathbf{c}_i|}\sum_{\mathbf{x}\in\mathbf{c}_i} sim(\mathbf{x},\mathbf{c}_i{}^*)\right)$$

  - **Inter-cluster similarity** is low

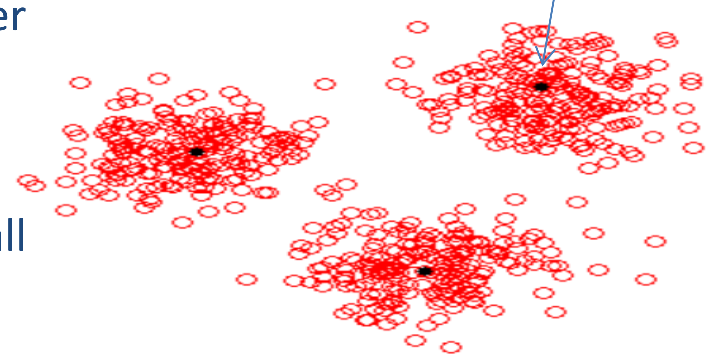
$$\frac{1}{\sum_{\mathbf{c}_i,\mathbf{c}_j}|\mathbf{c}_i||\mathbf{c}_j|}\sum_{\mathbf{x}\in\mathbf{c}_i,\mathbf{x}'\in\mathbf{c}_j} sim(\mathbf{x},\mathbf{x}') \text{ or } \frac{1}{k(k-1)}\sum_{\mathbf{c}_i{}^*,\mathbf{c}_j{}^*} sim(\mathbf{c}_i{}^*,\mathbf{c}_j{}^*)$$

Cluster centroids or clustroids

Centroid: element representing the center of the cluster, e.g. in vector space:

$$\mathbf{c}_i{}^* = \frac{1}{|\mathbf{c}_i|}\sum_{\mathbf{x}\in\mathbf{c}_i}\mathbf{x}$$

Clustroid: cluster point that is closest to all cluster points



External clustering criteria (1)

- How well does the clustering of N elements $\boldsymbol{C} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k\}$ represent the ground truth classes $\boldsymbol{G} = \{\boldsymbol{c}'_1, \dots, \boldsymbol{c}'_l\}$

  - Purity (each cluster should possibly contain only elements from one class)

    $$Purity(\boldsymbol{C}, \boldsymbol{G}) = \frac{1}{N} \sum_{i=1}^{k} \max_j \{|\boldsymbol{c}_i \cap \boldsymbol{c}'_j|\}$$

    Note: purity is 1 if each element is in its own cluster

  - Normalized mutual information (each cluster should possibly contain only elements from one class and possibly all the elements from that class)

$$NMI(\boldsymbol{C}, \boldsymbol{G}) = \frac{\sum_i \sum_j \frac{|\boldsymbol{c}_i \cap \boldsymbol{c}'_j|}{N} \log \frac{N|\boldsymbol{c}_i \cap \boldsymbol{c}'_j|}{|\boldsymbol{c}_i||\boldsymbol{c}'_j|}}{\frac{1}{2}\left(\sum_i \frac{|\boldsymbol{c}_i|}{N} \log \frac{N}{|\boldsymbol{c}_i|} + \sum_i \frac{|\boldsymbol{c}'_i|}{N} \log \frac{N}{|\boldsymbol{c}'_i|}\right)}$$

External clustering criteria (2)

- How well does the clustering of N elements $\boldsymbol{C} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k\}$ represent the ground truth classes $\boldsymbol{G} = \{\boldsymbol{c}'_1, \dots, \boldsymbol{c}'_l\}$

  - **Rand index** (accuracy, i.e., percentage of agreements with ground truth)

$$Rand(\boldsymbol{C}, \boldsymbol{G}) = \frac{TP + TN}{TP + TN + FP + FN}$$

    where

    TP: # pairs in same group in C and in G

    TN: # pairs in different groups in C and in G

    FP: # pairs in same group in C but in different groups in G

    FN: # pairs in same group in G but in different groups in C

  - **Precision, Recall, F-measure** can be defined analogously

Impossibility Theorem

- Let $f_d: D \mapsto 2^D$ be a partitioning function on the dataset D based on a (metric or non-metric) distance function $d: D \times D \mapsto \mathbb{R}_0$ that satisfies $d(x, y) = 0 \Leftrightarrow x = y$

The following axioms cannot be satisfied simultaneously:

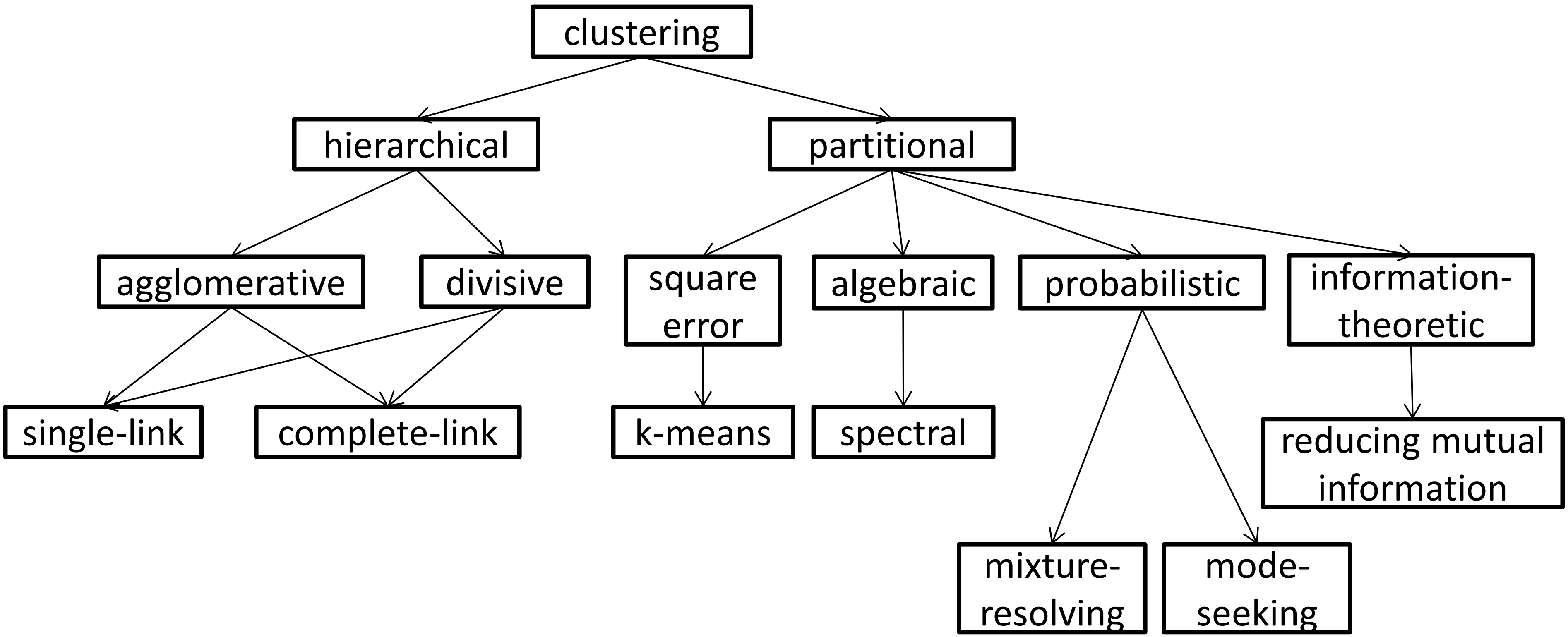
- Scale-invariance:
  for any d and any $\alpha > 0$: $f_d = f_{\alpha d}$

- Expressiveness (control over the data):
  for any partitioning $\Pi \subseteq 2^D$ there exists a d, such that f_d produces Π

- Consistency:
  for any d, let d' be such that $d'(x, y) < d(x, y)$ if x, y are in the same cluster created by f_d and $d'(x, y) > d(x, y)$ otherwise, then $f_{d'} = f_d$

Source: J. Kleinberg, NIPS 2002

Simple taxonomy of generic clustering approaches



Hierarchical clustering

- **Divisive/top-down**
  - Start with a single cluster containing the whole dataset
  - In each iteration:

    identify the cluster $\mathbf{c}$ with lowest intra-cluster similarity

    divide it into two clusters $\mathbf{c}_1, \mathbf{c}_2$ with minimal $sim(\mathbf{c}_1, \mathbf{c}_2)$

    stop when each cluster has only one element

With exhaustive search $O(2^n)$

- **Agglomerative/bottom-up**
  - Start with a cluster for each element in the dataset
  - In each iteration:

    identify the two clusters $\mathbf{c}_1, \mathbf{c}_2$ with maximal $sim(\mathbf{c}_1, \mathbf{c}_2)$

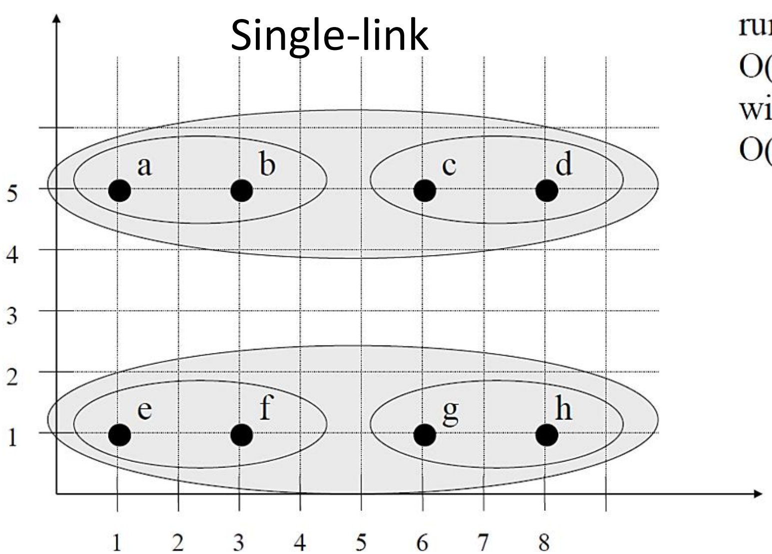
    merge $\mathbf{c}_1, \mathbf{c}_2$ into $\mathbf{c} = \mathbf{c}_1 \cup \mathbf{c}_2$

    stop when there is single cluster

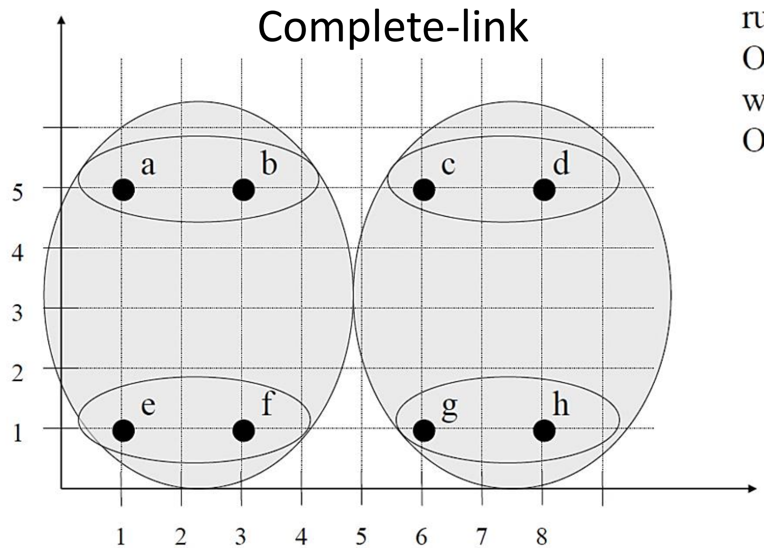
$O(n^3)$ (at best $O(n^2)$ for special cases)
Best known methods: **SLINK, CLINK**

Hierarchical clustering: Single-link vs. complete-link

- Let $\max\{d(\mathbf{c},\mathbf{c}'), d(\mathbf{c},\mathbf{c}'')\} \geq d(\mathbf{c},\mathbf{c}' \cup \mathbf{c}'')$ for all partitions $\mathbf{c},\mathbf{c}',\mathbf{c}''$
  - **Single-link method:** $d(\mathbf{c},\mathbf{c}') = d(\mathbf{x},\mathbf{x}')$, such that $\mathbf{x} \in \mathbf{c}$ and $\mathbf{x}' \in \mathbf{c}'$ have the **minimum distance** of all elements from $\mathbf{c},\mathbf{c}'$
  - **Complete-link method**: $d(\mathbf{c},\mathbf{c}') = d(\mathbf{x},\mathbf{x}')$, such that $\mathbf{x} \in \mathbf{c}$ and $\mathbf{x}' \in \mathbf{c}'$ have the **maximum distance** of all elements from $\mathbf{c},\mathbf{c}'$ (merge the two clusters with **smallest maximum pairwise distance**)

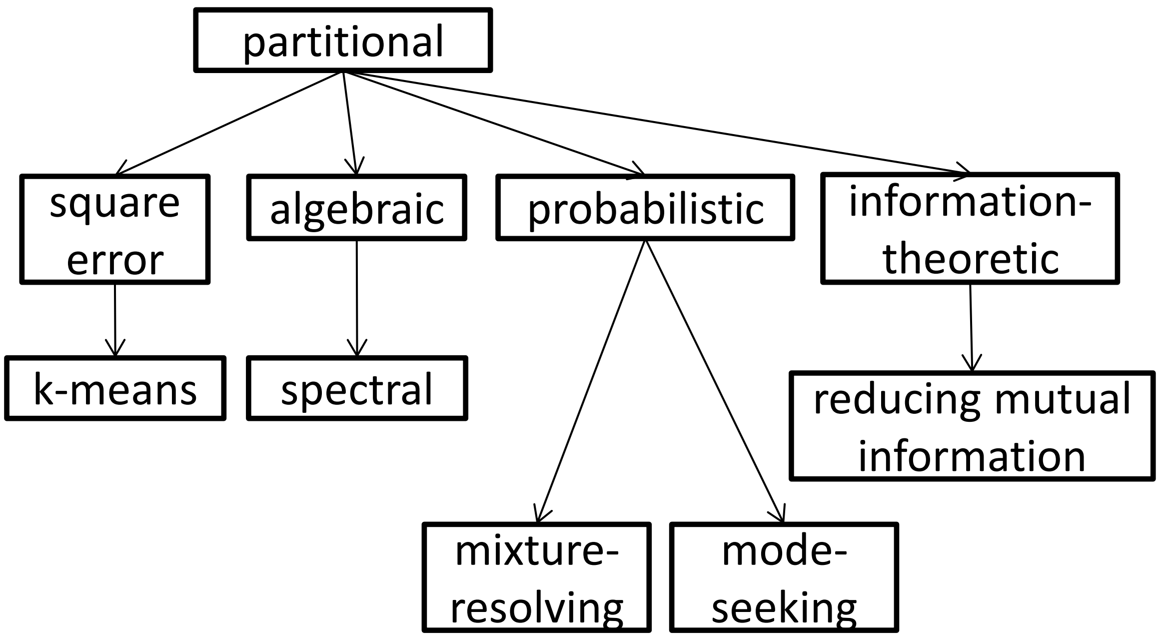


run-time: $O(n^2)$ with space $O(n^2)$



run-time: $O(n^2 \log n)$ with space $O(n^2)$

Partitional clustering approaches



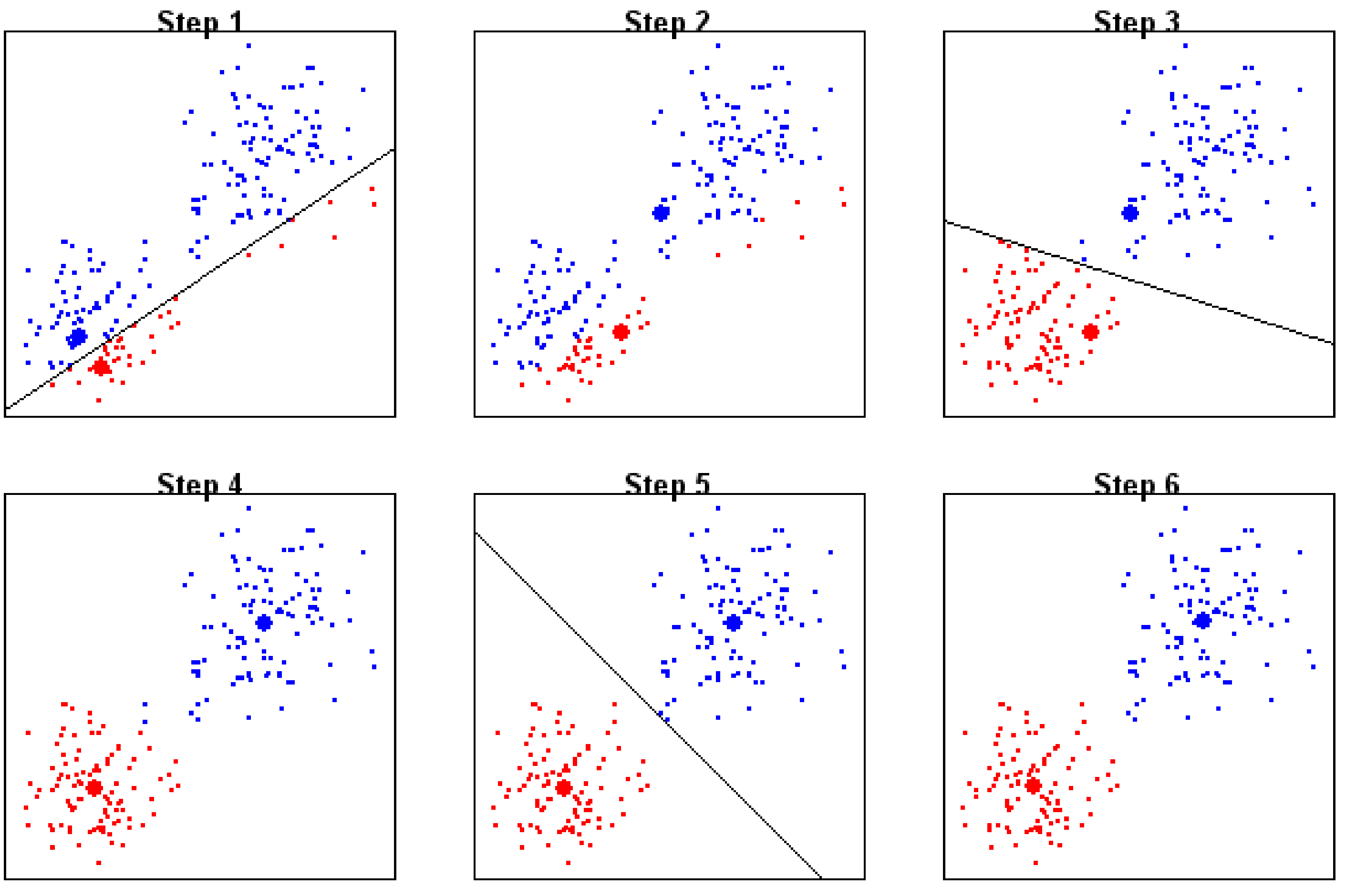
K-means (1)

- For given data records $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^m$, find $k \leq n$ clusters $\mathbf{c}_1, \dots, \mathbf{c}_k$ according to some similarity measure sim and a cluster stability threshold t

```
Randomly choose prototype clusters c_1,…,c_k, by choosing
random centroids and assigning a point to its closest
centroid
While there exists c_i with Σ_{x∈c_i} ||x − c_i*||² > t
      For j := 1 to n do
            Assign x_j to c_l with the highest sim(c_l*, x_j)
      For j := 1 to k do
            Recompute c_j* //where c_j* = (1/|c_j|) Σ_{x∈c_j} x
```

K-means (2)

- Example



From http://astrostatistics.psu.edu/su09/lecturenotes/clus2.html

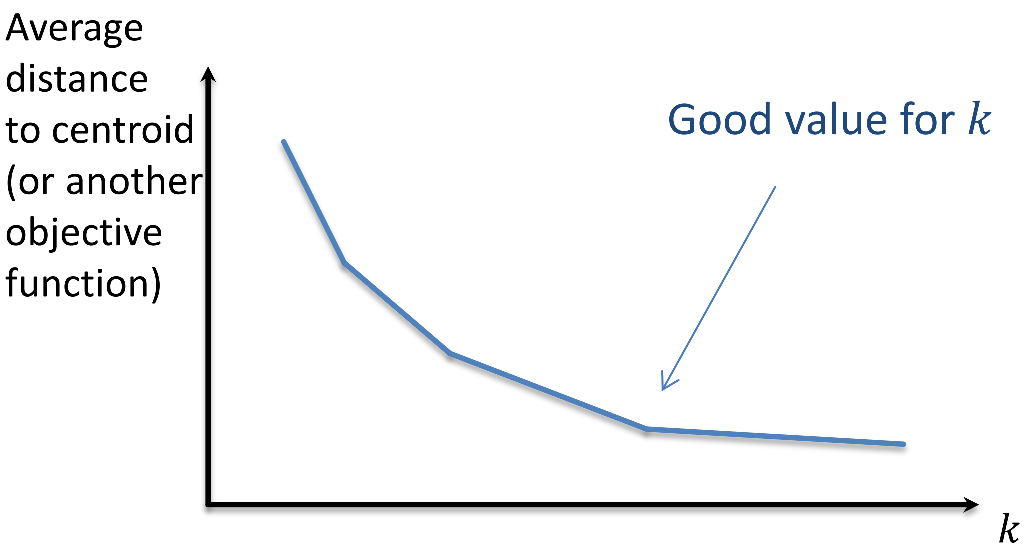
K-means (3)

- In practice, runtime is polynomial
- Theoretical complexity is exponential ($2^{\Omega(n)}$)
- k can be determined experimentally or based on the minimum-description-length (MDL) principle
- Choice of initial prototype vectors influences the result; often k-means is re-run multiple times with random choices
- Initial prototype vectors could be chosen by using another – very efficient – clustering method (on random sample of the data records)
- Any arbitrary metric can be used

Getting k right

1) For increasing values of k estimate the change of the average distance to the centroid

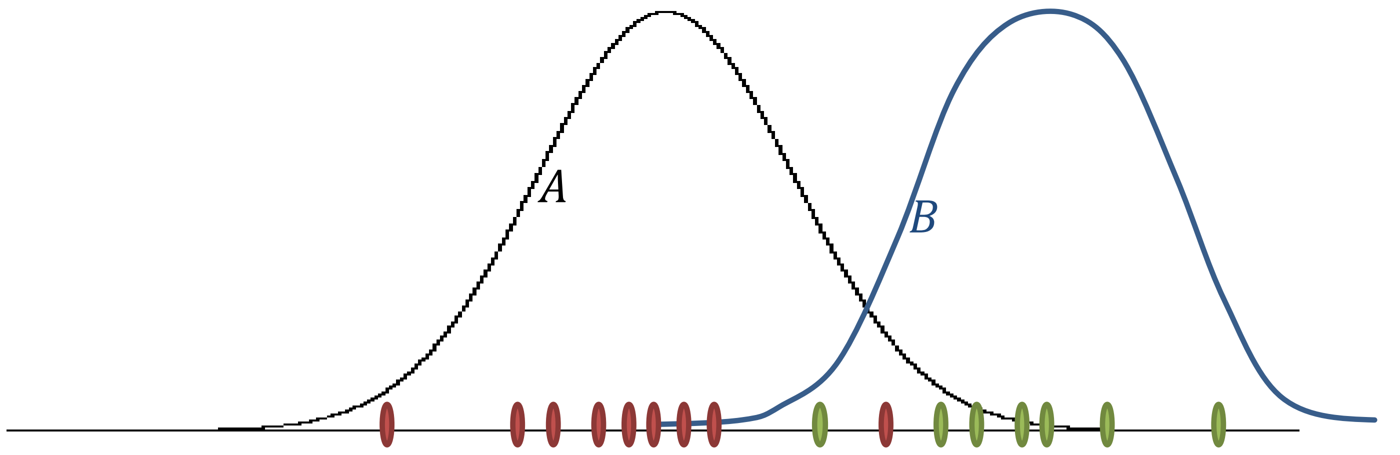
Choose k for which average distance changes very little



2) MDL criterion: Check whether cost of encoding the information of the current cluster configuration exceeds the cost of the previous configuration

Mixture models for clustering (1)

- Resolving mixtures through expectation maximization (EM) for clustering



$$L(\mu_A, \sigma_A, \mu_B, \sigma_B, p_A; x_1, \dots, x_n) = \prod_i (p_A \, P(x_i|A) + p_B \, P(x_i|B))$$

1. **Expectation step:** Estimate the expected membership value of each point x_i given the current estimations of $\mu_A, \sigma_A, \mu_B, \sigma_B, p_A, p_B$
2. **Maximization step:** Maximize the likelihood of $\mu_A, \sigma_A, \mu_B, \sigma_B, p_A, p_B$ in light of the observations (i.e., use the expected membership values to re-estimate the parameters)

Mixture models for clustering (2)

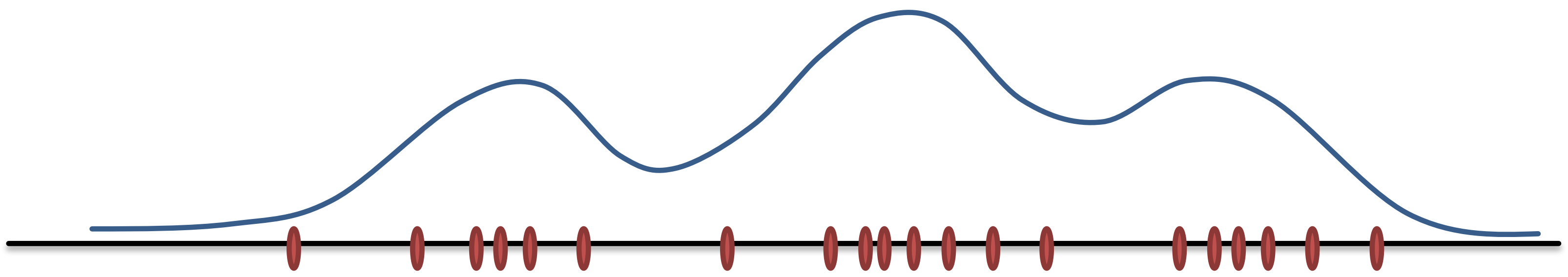
- Resolving mixtures through expectation maximization (EM)

$$L(\mu_A, \sigma_A, \mu_B, \sigma_B, p_A; x_1, \dots, x_n) = \prod_i (p_A\, P(x_i|A) + p_B\, P(x_i|B))$$

- EM in practice
  - Initialize the parameters $\mu_A, \sigma_A, \mu_B, \sigma_B, p_A, p_B$ to some random values (note: $p_A + p_B = 1$)
  - E-step: Compute expected membership values $P(A|x_i), P(B|x_i)$
  - M-step: Re-estimate the parameters
  - Iterate steps 2 and 3 until convergence (i.e., until changes of log likelihood are negligible)

Mode-seeking clustering with Mean-Shift

- Method to locate the maxima of a density function

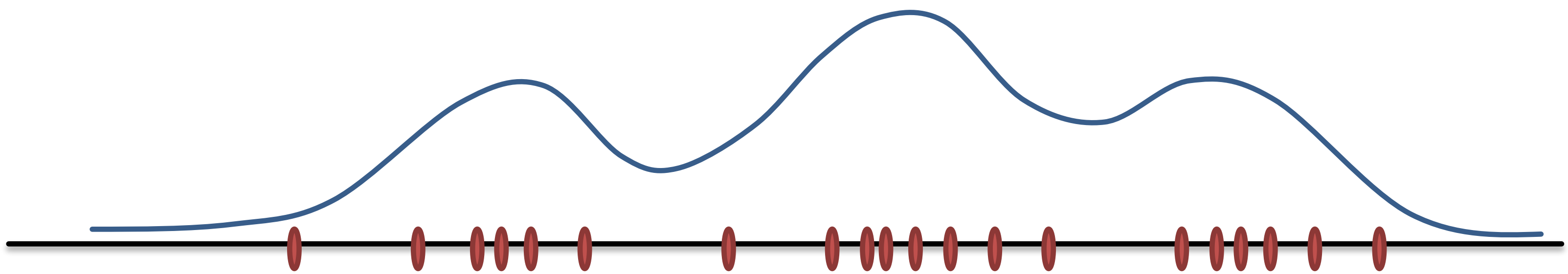


```
Select x_1, ..., x_k of the n sample points (at random) as modes
While x_1, ..., x_k not converged
    For each x_i
```

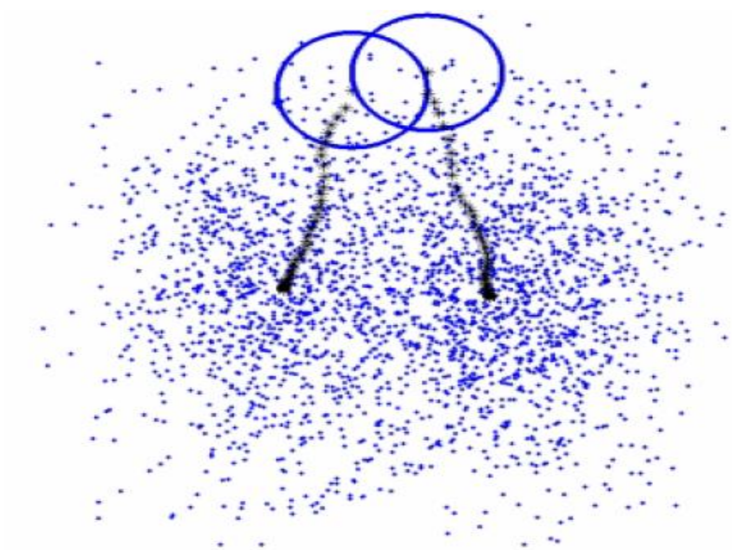
$$m(\mathbf{x}_i) \coloneqq \frac{\sum_{\mathbf{y}\in Nb(\mathbf{x}_i)} K(\mathbf{y},\mathbf{x}_i)\,\mathbf{y}}{\sum_{\mathbf{y}\in Nb(\mathbf{x}_i)} K(\mathbf{y},\mathbf{x}_i)}, \quad \text{with: } K(\mathbf{y},\mathbf{x}) = e^{c\|\mathbf{y}-\mathbf{x}\|}$$

$$\mathbf{x}_i \coloneqq m(\mathbf{x}_i)$$

Mean-Shift visualization



- Mean shift vector always points toward the direction of maximum increase in density



Efficient mode-seeking clustering with DBSCAN

- DBSCAN: density-based clustering for applications with noise

```
For each data point x do
   Insert x into (spatial) index //(e.g. R-tree)
For each data point x do
   Locate all points with distance less than d_max to x
   If these points form a single cluster then
        Add x to this cluster
   Else
        If there are at least min_pts data points (that
        do not yet belong to a cluster) such that for all
        point pairs the distance is less than d_max then
            Construct a new cluster with these points
```

- Mode-seeking algorithm with average run-time: $O(n \ \log n)$
- Data points that are added later can be easily assigned to a cluster
- Points that do not belong to any cluster are considered "noise"

Spectral clustering techniques

- Typically used derive a lower-dimensional representation of the data

- Variant 1
  - Map each data point into k-dimensional space
  - Assign each point to its highest-value dimension (strongest spectral component)

- Variant 2
  - Compute k clusters for the data points (using any clustering algorithm)
  - Project data points onto k centroid vectors (“axes” of k-dim. space)

Spectral clustring

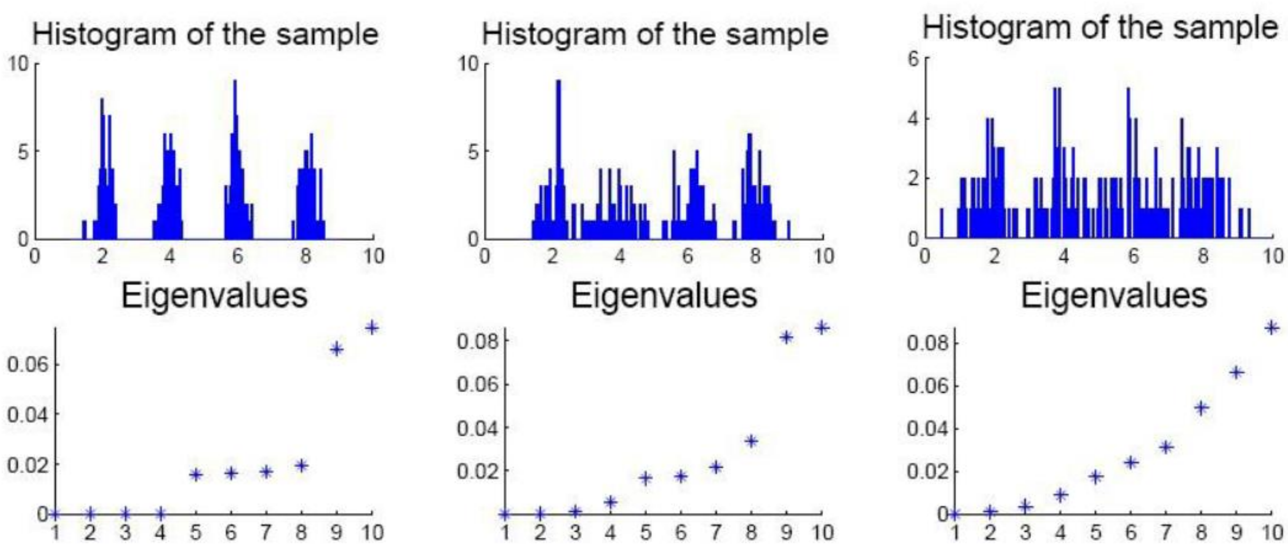
➤ Spectral clustering algorithm for variant 1

```
Construct similarity graph of n data points
Construct graph Laplacian L = D − W // D: diagonal with
                                    // D_ii =degree of i'th node
                                    //W weighted adjacency matrix
Compute smallest k Eigenvalues and Eigenvectors
                                    // Lx = λDx
                                    // λ: Eigenvalue
Let M be the n×k matrix with these Eigenvectors
as columns
Treat the n rows of M as k-dim. data points
Run k-means with these points
```

Runtime: $\Theta(|L|^2)$

Spectral clustering – choosing k

- **Theorem**
  - All Eigenvalues of a graph Laplacian are non-negative reals.
  - The multiplicity k of the smallest Eigenvalue 0 is the number of connected components of the graph.
  - The corresponding Eigenvectors $\mathbf{x}_1, \dots, \mathbf{x}_k$ are indicator vectors of the components $\mathbf{x}_i(j) = 1$ if node j is in the i'th component, and 0 otherwise.



Source: U. von Luxburg, A Tutorial on Spectral Clustering

Summary

- Clustering goals
  - Internal criteria
  - External criteria
  - Impossibility theorem

- Hierarchical clustering
  - Divisive
  - Agglomerative
  - Merging based on single-link, complete-link heuristics

- Flat clustering
  - K-means (getting k right)
  - Mean-shift
  - DBSCAN
  - Spectral clustering