SOCIAL SEARCH – CLUSTERING

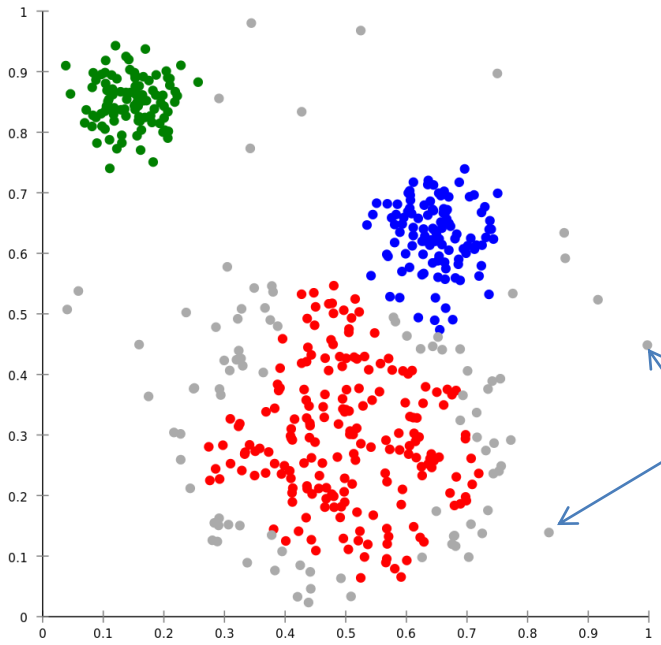
Outline

- Intro
- Basics of probability and information theory
- Retrieval models
- Retrieval evaluation
- Link analysis
- From queries to top-k results
- Social search
  - Overview & applications
  - Clustering & recommendation

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

Internal clustering criteria

- General goal:

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

  - Intra-cluster similarity is high

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

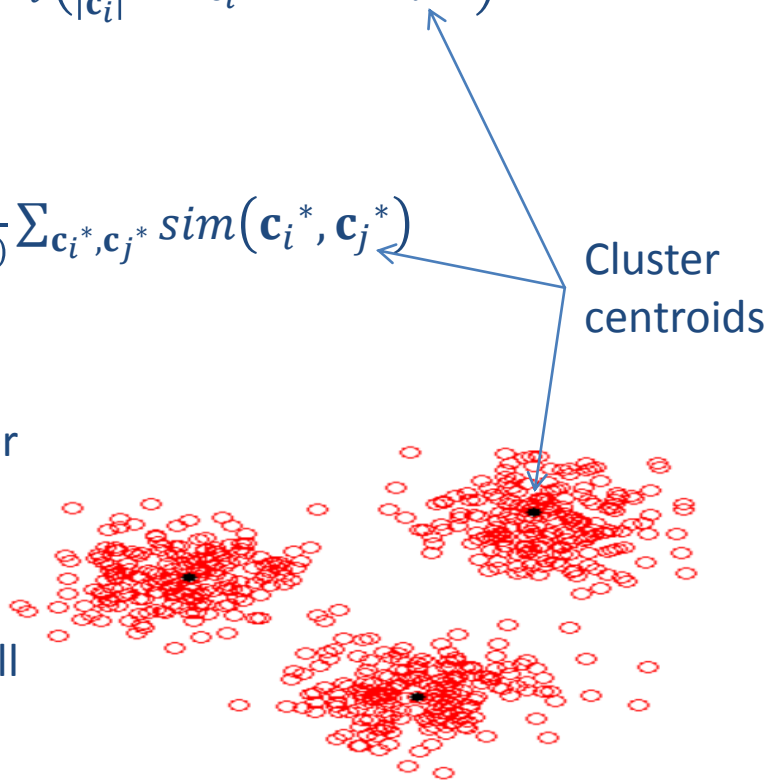
  - Inter-cluster similarity is low

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

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



From metric distances to similarities

- Similarity is typically based on a **metric distance**:

  A space M with distance d 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 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	$\|\mathbf{x} - \mathbf{y}\|_1 = \sum_i \|x_i - y_i\|$
Maximum	$\|\mathbf{x} - \mathbf{y}\|_\infty = \max_i \|x_i - y_i\|$
Mahalanobis	$d_{maha}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i \left(\frac{x_i - y_i}{\sigma_i}\right)^2}$ (for normally distributed data)

Popular similarity and distance measures

- Pearson correlation

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

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

- Cosine similarity

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

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

- Jaccard similarity

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

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

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

External clustering criteria (1)

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

  - **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 only elements from one class and possibly all the elements from that class)

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

External clustering criteria (2)

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

  - **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.

K-means (1)

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

Randomly choose prototype clusters $\mathbf{c}_1, \ldots, \mathbf{c}_k$, by choosing random centroids and assigning a point to its closest centroid

While there exists $\mathbf{c}_i$ with $\sum_{\mathbf{x} \in \mathbf{c}_i} \|\mathbf{x} - \mathbf{c}_i^*\|^2 > t$

For $j := 1$ to n do

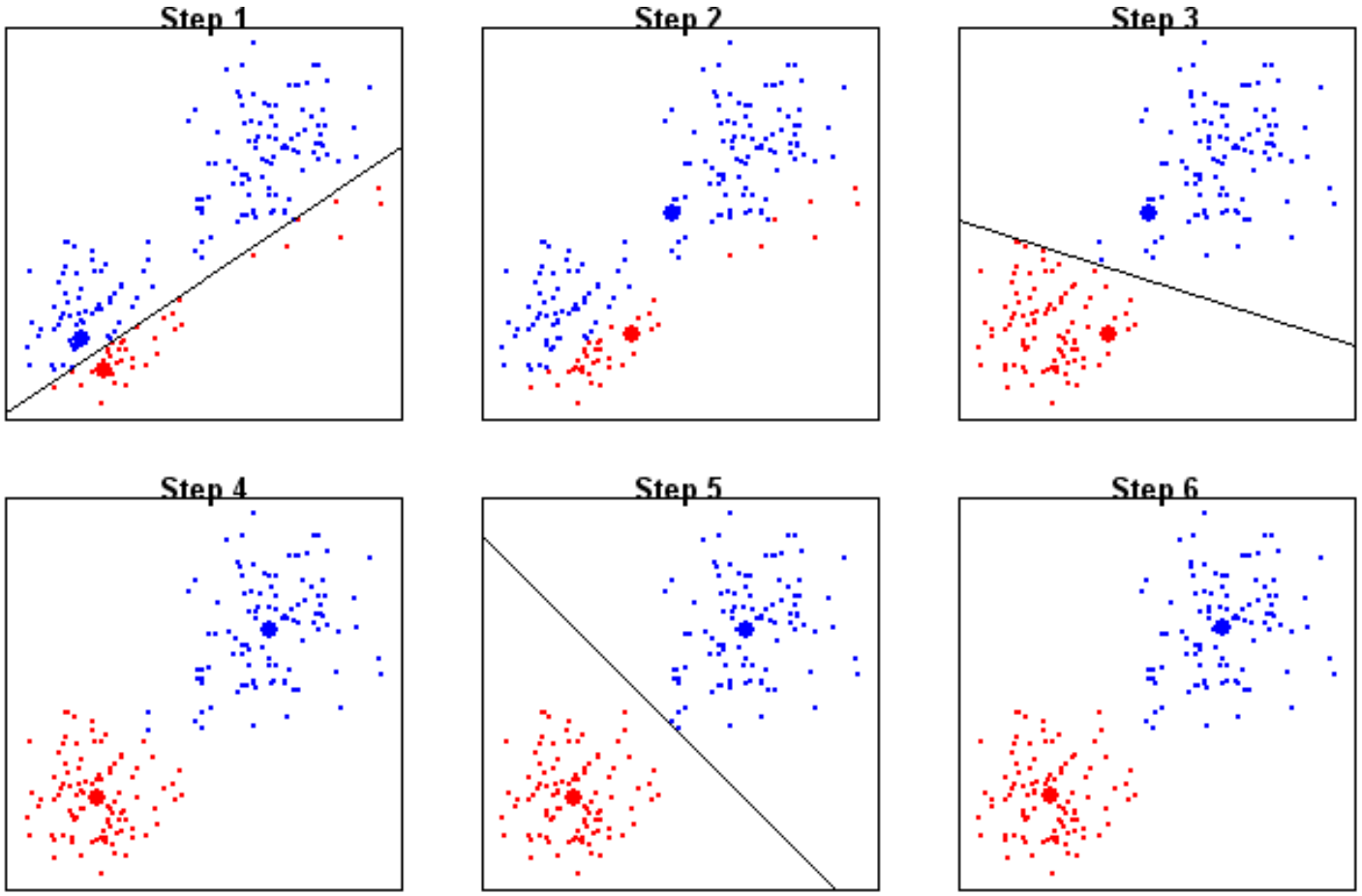
Assign $\mathbf{x}_j$ to $\mathbf{c}_l$ with the largest $sim(\mathbf{c}_l^*, \mathbf{x}_j)$

For $j := 1$ to k do

Recompute $\mathbf{c}_j^*$ //where $\mathbf{c}_j^* = \frac{1}{|\mathbf{c}_j|} \sum_{\mathbf{x} \in \mathbf{c}_j} \mathbf{x}$

K-means (2)

- Example



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

K-means (3)

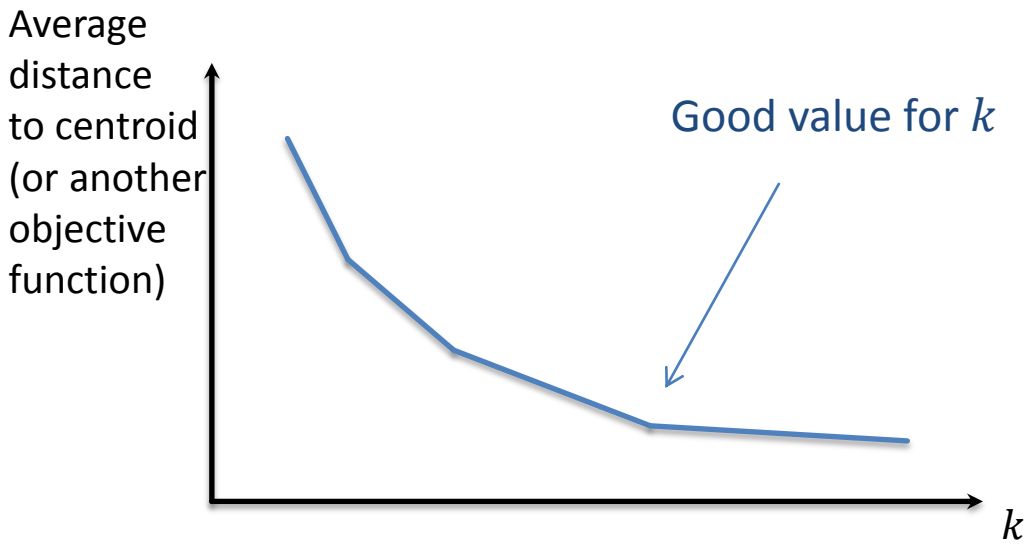
- In practice, runtime is polynomial
- Theoretical complexity is exponential ($2^{\Omega(n)}$)
- k can be determined experimentally or based on the minimum-description-length (MDL) principle
- Choice of initial prototype vectors influences the result; often k-means is re-run multiple times with random 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

Typical clustering problem

- Getting k right

  1) For different (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.

DBSCAN

- Mode-seeking clustering method
  - 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”

Co-clustering (1)

- Given a feature-item matrix (e.g., containing relative co-occurrence frequencies)
- Is it possible to group features and items simultaneously, so that latent groups (e.g., topics) are revealed?
  - Idea: related features occur in related items and related items have related features

	D_1	D_2	D_3	D_4	D_5	D_6
soccer	.05	.05	.05	0	0	0
goal	.05	.05	.05	0	0	0
basketball	0	0	0	.05	.05	.05
NBA	0	0	0	.05	.05	.05
team	.04	.04	0	.04	.04	.04
player	.04	.04	.04	0	.04	.04

	$\hat{y}_1$	$\hat{y}_2$
$\hat{x}_1$	.3	0
$\hat{x}_2$	0	.3
$\hat{x}_3$	.2	.2

Example from Dhillon et al., KDD 2003

Co-clustering (2)

- Formally, we are given the joint distribution of features and items, e.g.:

$$p(X,Y) = \begin{bmatrix} .05 & .05 & .05 & 0 & 0 & 0 \\ .05 & .05 & .05 & 0 & 0 & 0 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ 0 & 0 & 0 & .05 & .05 & .05 \\ .04 & .04 & 0 & .04 & .04 & .04 \\ .04 & .04 & .04 & 0 & .04 & .04 \end{bmatrix}$$

- Goal: derive a clustering of rows (denoted by $\hat{X}$) and columns (denoted by $\hat{Y}$) that minimizes the loss in mutual information

$$I(X,Y) - I(\hat{X},\hat{Y}) = KL\big(p(X,Y) \big\| p(\hat{X},\hat{Y})p(X|\hat{X})p(Y|\hat{Y})\big)$$

e.g.: $$p(\hat{X},\hat{Y}) = \begin{bmatrix} .3 & 0 \\ 0 & .3 \\ .2 & .2 \end{bmatrix}$$

)-: This problem is NP-hard!

Co-clustering (3)

- Greedy algorithm
  - Input: joint probability distr. $p(X, Y)$, the desired number k of row clusters, the desired number l of column clusters
  - Output: partitions $\hat{X}, \hat{Y}$ such that $I(X, Y) - I(\hat{X}, \hat{Y})$ is minimized

Start with initial $\hat{X}^{(0)}, \hat{Y}^{(0)}$

Repeat until no improvement is possible concerning $I(X, Y) - I(\hat{X}^{(t)}, \hat{Y}^{(t)})$

Recompute p_{ij} for all blocks b_{ij} according to $\hat{X}^{(t)}, \hat{Y}^{(t)}$

For each row $x \in X$

Assign x to row block i that minimizes local loss in mutual information, i.e., $I(x, Y) - I\left(\hat{x}_i^{(t)}, \hat{Y}^{(t)}\right)$

Recompute p_{ij} for all blocks b_{ij} according to $\hat{X}^{(t+1)}, \hat{Y}^{(t+1)}$

For each column $y \in Y$

Assign y to column block j that minimizes local loss in mutual information, i.e., $I(X, y) - I\left(\hat{X}^{(t+1)}, \hat{y}_j^{(t+1)}\right)$

- Algorithm converges to (local) minimum in $O(\#iterations \cdot (k + l))$

Spectral clustering techniques

- Typically used for graph-based clustering
- 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 // $L\mathbf{x} = \lambda D\mathbf{x}$
// λ: Eigenvalue

Let M be the $n \times k$ matrix with these Eigenvectors as columns

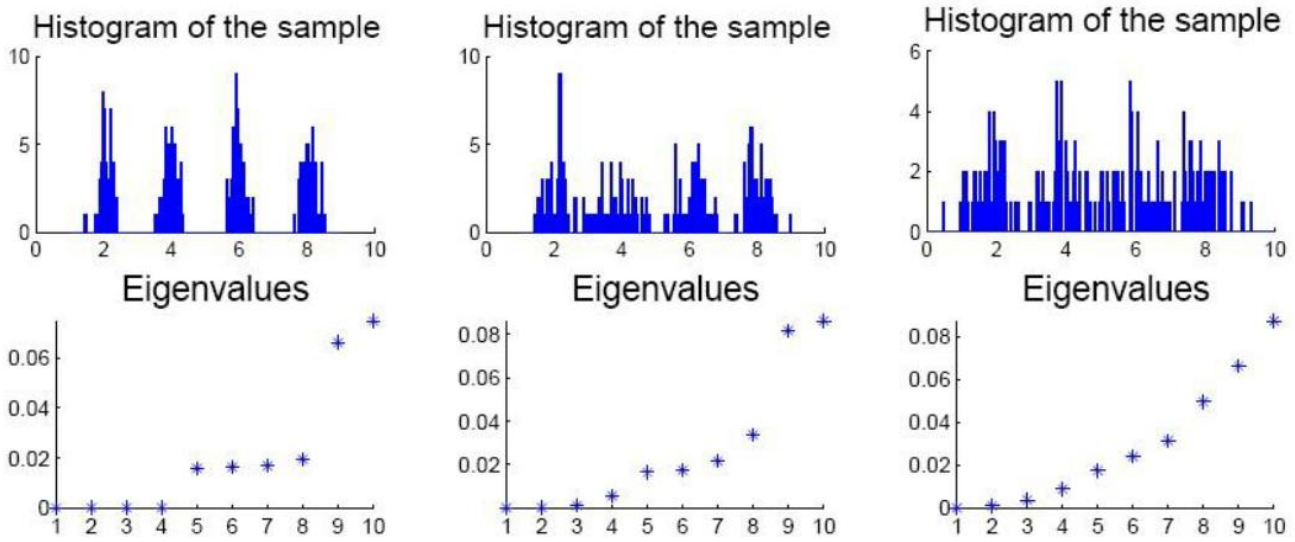
Treat the n rows of M as k-dim. data points

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 $\mathbf{x}_1, \ldots, \mathbf{x}_k$ 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
  - Impossibility theorem
  - External criteria
- Clustering techniques
  - K-means (getting k right)
  - DBSCAN
  - Co-clustering
  - Spectral clustering