CS 512 Data Mining Principles

Cluster Analysis

Hanghang Tong, Computer Science, Univ. Illinois at Urbana-Champaign, 2021

Suggested studying time: 2/22/2021-3/4/2021

Cluster Analysis

- Cluster Analysis Overview 
- K-Means
- Mixture Models and E-M algorithm
- Spectral Methods
- Summary

Cluster Analysis: An Introduction

- What Is Cluster Analysis?
- Applications of Cluster Analysis
- Cluster Analysis: Requirements and Challenges
- Cluster Analysis: A Multi-Dimensional Categorization
- An Overview of Typical Clustering Methodologies
- An Overview of Clustering Different Types of Data
- An Overview of User Insights and Clustering

What Is Cluster Analysis?

- **What is a cluster?**
  - A cluster is a collection of data objects which are
    - Similar (or related) to one another within the same group (i.e., cluster)
    - Dissimilar (or unrelated) to the objects in other groups (i.e., clusters)
- **Cluster analysis** (or ***clustering***, ***data segmentation***, …)
  - Given a set of data points, partition them into a set of groups (i.e., clusters) which are as similar as possible
- Cluster analysis is **unsupervised learning** (i.e., no predefined classes)
  - This contrasts with *classification* (i.e., *supervised learning*)
- Typical ways to use/apply cluster analysis
  - As a stand-alone tool to get insight into data distribution, or
  - As a preprocessing (or intermediate) step for other algorithms

What Is Good Clustering?

- A good clustering method will produce high quality clusters which should have
  - **High intra-class similarity:** Cohesive within clusters
  - **Low inter-class similarity:** Distinctive between clusters
- **Quality function**
  - There is usually a separate "quality" function that measures the "goodness" of a cluster
  - It is hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective
- There exist many similarity measures and/or functions for different applications
- Similarity measure is critical for cluster analysis

Cluster Analysis: Applications

- A key intermediate step for other data mining tasks
  - Generating a compact summary of data for classification, pattern discovery, hypothesis generation and testing, etc.
  - Outlier detection: Outliers—those “far away” from any cluster
- Data summarization, compression, and reduction
  - Ex. Image processing: Vector quantization
- Collaborative filtering, recommendation systems, or customer segmentation
  - Find like-minded users or similar products
- Dynamic trend detection
  - Clustering stream data and detecting trends and patterns
- Multimedia data analysis, biological data analysis and social network analysis
  - Ex. Clustering images or video/audio clips, gene/protein sequences, etc.

Considerations for Cluster Analysis

- **Partitioning criteria**
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable, e.g., grouping topical terms)
- **Separation of clusters**
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- **Similarity measure**
  - Distance-based (e.g., Euclidean, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- **Clustering space**
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- **Quality**
  - Ability to deal with different types of attributes: Numerical, categorical, text, multimedia, networks, and mixture of multiple types
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
- **Scalability**
  - Clustering all the data instead of only on samples
  - High dimensionality
  - Incremental or stream clustering and insensitivity to input order
- **Constraint-based clustering**
  - User-given preferences or constraints; domain knowledge; user queries
- **Interpretability and usability**
  - The final generated clusters should be semantically meaningful and useful

Cluster Analysis

- Cluster Analysis Overview
- K-Means
- Mixture Models and E-M algorithm
- Spectral Methods
- Summary

Partitioning Algorithms: Basic Concepts

- Partitioning method: Discovering the groupings in the data by optimizing a specific objective function and iteratively improving the quality of partitions
- *K*-partitioning method: Partitioning a dataset ***D*** of ***n*** objects into a set of ***K*** clusters so that an objective function is optimized (e.g., the sum of squared distances is minimized, where c_k is the centroid or medoid of cluster C_k)
  - A typical objective function: **Sum of Squared Errors** (**SSE**)

$$SSE(C) = \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - C_j\|^2$$

- Problem definition: Given *K*, find a partition of *K clusters* that optimizes the chosen partitioning criterion
  - Global optimal: Needs to exhaustively enumerate all partitions
  - Heuristic methods (i.e., greedy algorithms): *K-Means*, *K-Medians*, *K-Medoids*, etc.

The *K-Means* Clustering Method

- *K-Means* (MacQueen'67, Lloyd'57/'82)
  - Each cluster is represented by the center of the cluster
- Given K, the number of clusters, the *K-Means* clustering algorithm is outlined as follows
  - Select *K* points as initial centroids
  - **Repeat**
    - Form *K* clusters by assigning each point to its closest centroid
    - Re-compute the centroids (i.e., *mean point*) of each cluster
  - **Until** convergence criterion is satisfied
- Different kinds of measures can be used
  - Manhattan distance (L_1 norm), Euclidean distance (L_2 norm), Cosine similarity

How to Efficiently Clustering Data?

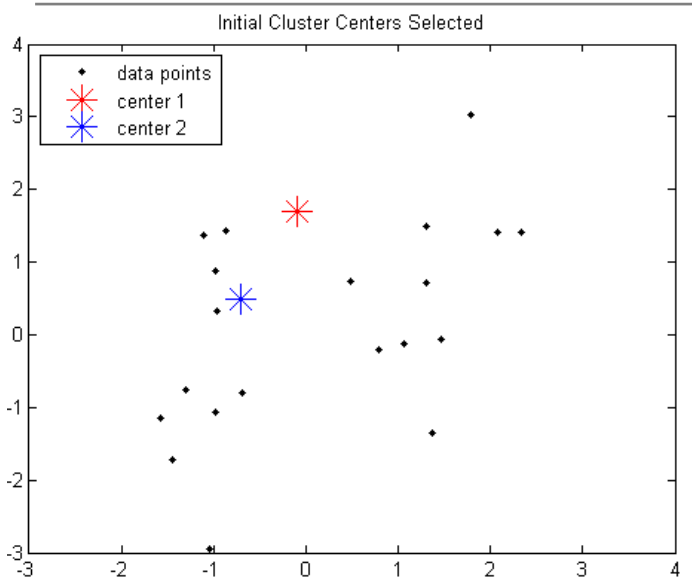
$$SSE(C) = \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - C_j\|^2$$

Memberships $\{m_{i,j}\}$ and centers $\{C_j\}$ are correlated.

Given centers $\{C_j\}$, $m_{i,j} = \begin{cases} 1 & j = \arg\min_k (x_i - C_j)^2 \\ 0 & \text{otherwise} \end{cases}$

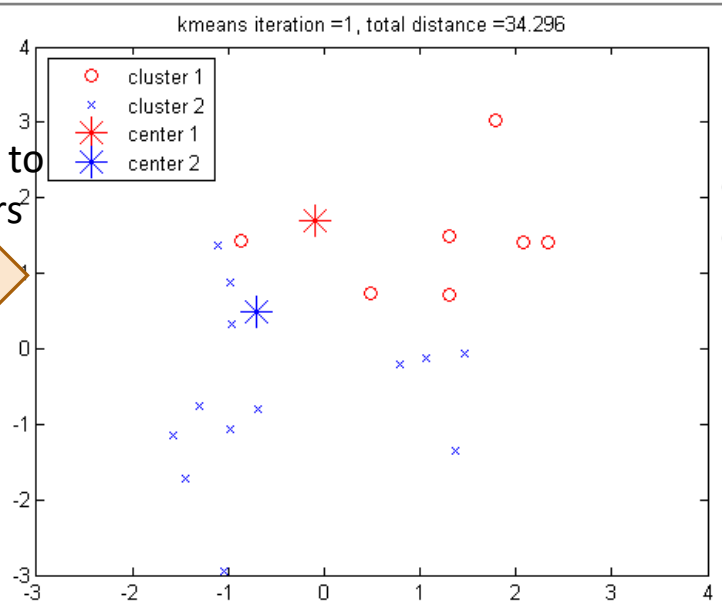
Given memberships $\{m_{i,j}\}$, $C_j = \frac{\sum_{i=1}^{n} m_{i,j} x_i}{\sum_{i=1}^{n} m_{i,j}}$

Example: *K-Means* Clustering

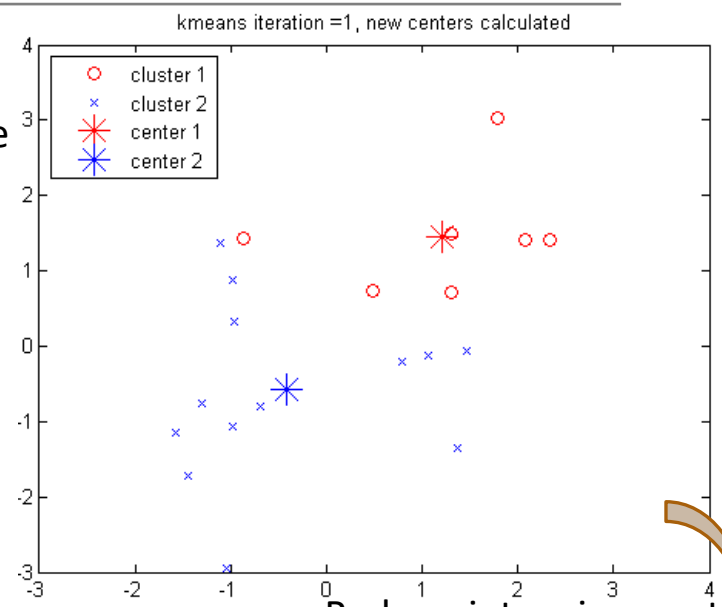


The original data points & randomly select $K = 2$ centroids

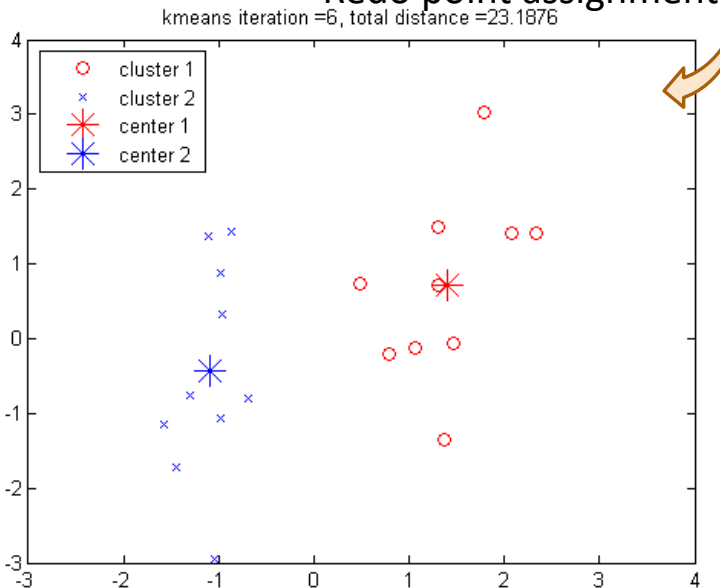
Assign points to clusters



Recompute cluster centers



Redo point assignment



Execution of the K-Means Clustering Algorithm

Select K points as initial centroids

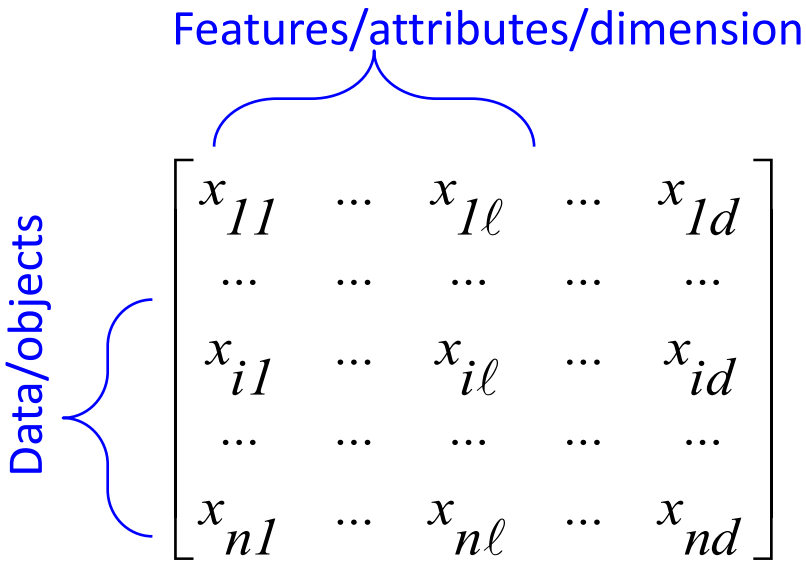
Repeat

- Form K clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

Until convergence criterion is satisfied

K-means as Matrix Factorization

$$SSE(C) = \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - C_j\|^2$$



1. X ~= F x G.
    1. X: data matrix
    2. What is F? what is G?
    3. Which norm to approximate?

2. F: n x k cluster membership matrix (0 or 1)
3. G: k x d cluster-description matrix

Computation of K-means: Find Similar Things

- d is relatively small, many mature indexing techniques
  - B-tree
  - Quad-tree
  - K-d-tree

- What if *d* if very high (e.g. 10K+)?
- A: LSH (can be viewed as special dimensionality reduction technique that preserves the 'locality')

K-means Summary

- Goal
  - Find the cluster membership
  - Find the cluster center (or more generally, the cluster description)

- Optimization
  - Objective (NP-hard if d>=2)
  - Algorithms: alternating (~= block coordinate descent)

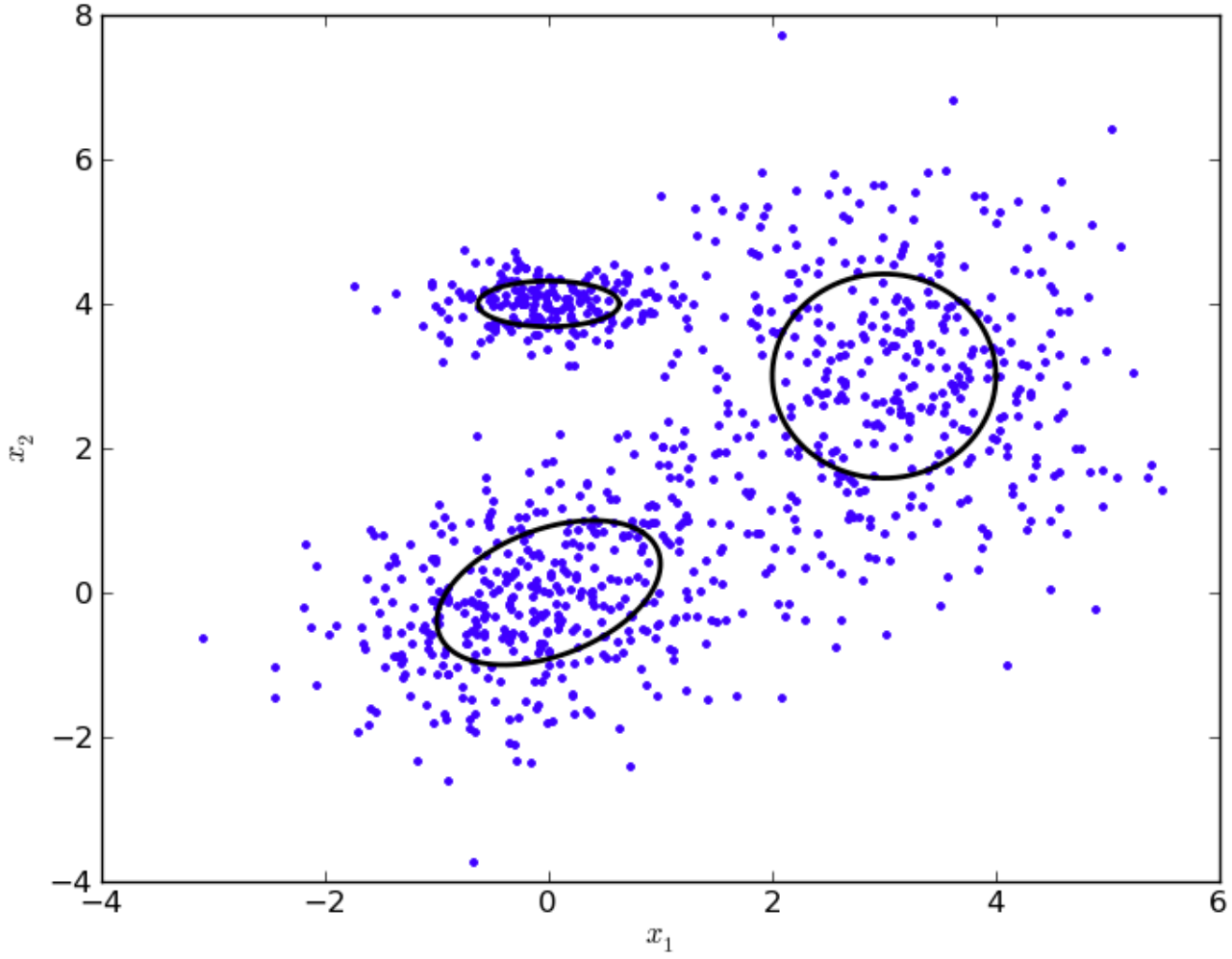
- Connections to
  - KNN
  - Matrix Factorization

- Many Variants (e.g., K-means++, K-medians, K-centers, etc.)

Cluster Analysis

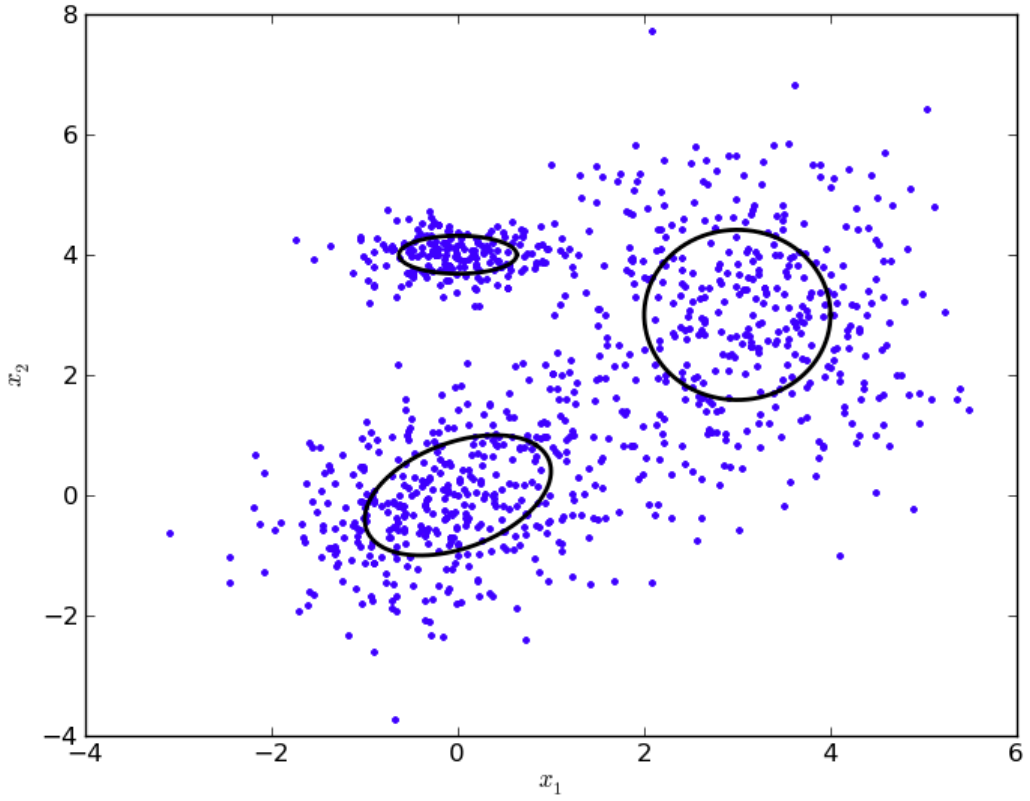
- Cluster Analysis Overview
- K-Means
- Mixture Models and E-M algorithm 
- Spectral Methods
- Summary

Hard Clustering Can Be Difficult

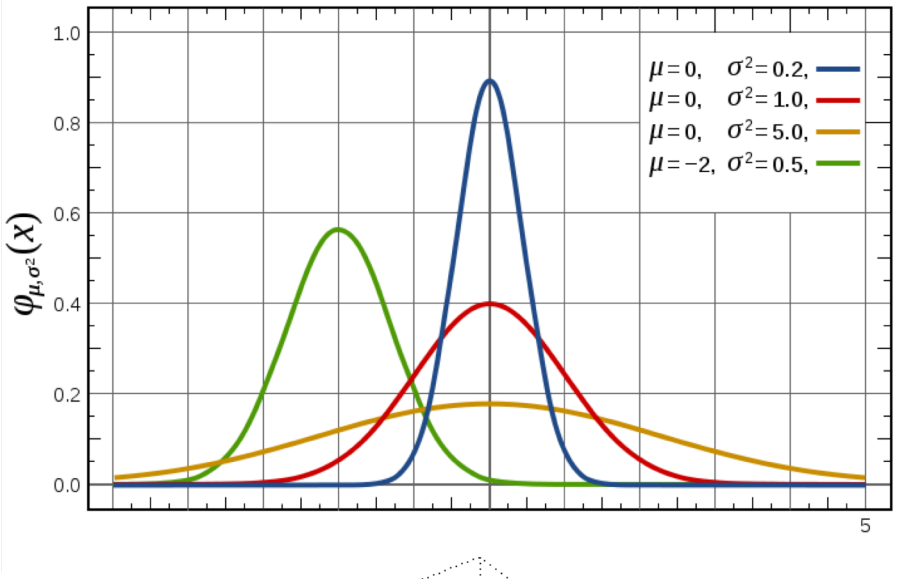


Soft Clustering

- Every object i is assigned to one cluster j with a probability
  - $P(z_i = j) \in [0,1]$ and $\Sigma_j P(z_i = j) = 1$
  - Where z_i is a hidden variable of which cluster x_i belongs to.



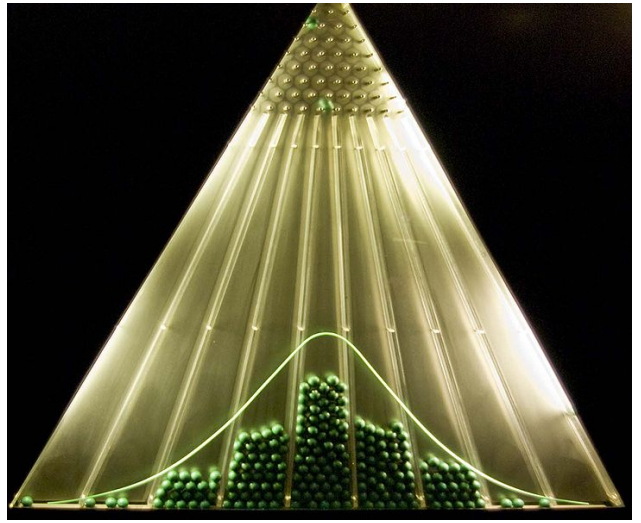
Gaussian Distribution

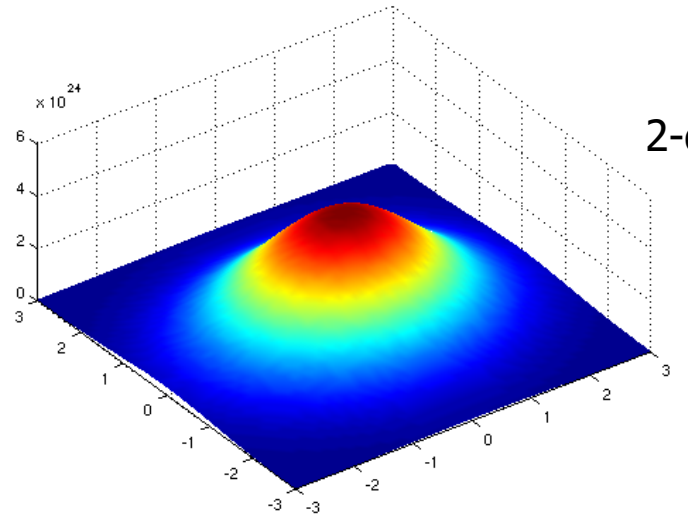


1-d Gaussian

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Bean machine: drop ball with pins





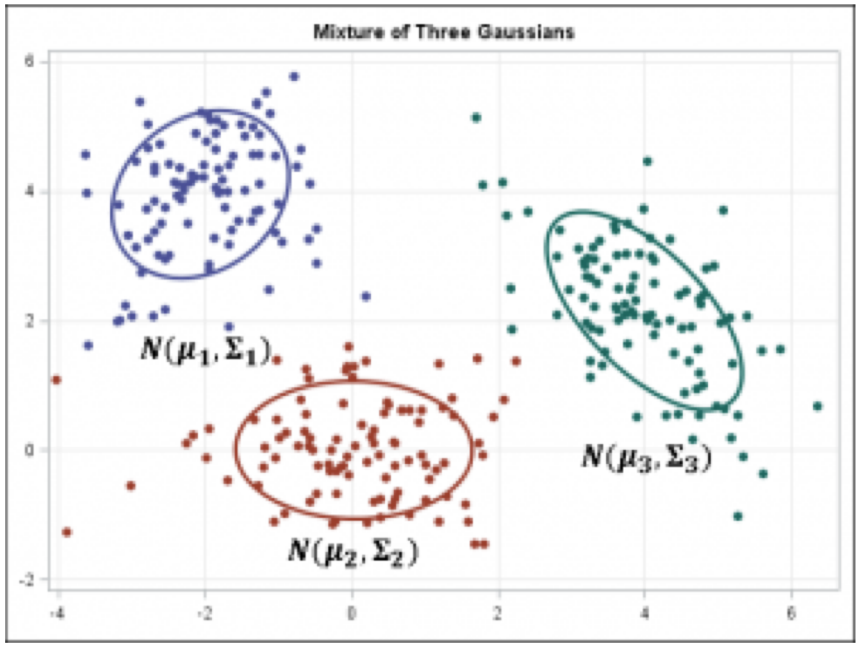
2-d Gaussian

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

From wikipedia and http://home.dei.polimi.it

Gaussian Mixture Model

- Assumptions
  - Each data point comes from one of K classes.
  - The cluster prior distribution w_j is *unknown*.
  - Each class c_j follows a Gaussian



distribution: $P(x|c_j, \theta_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}$

  - The parameters for each class μ_j, σ_j are *unknown*(need to be learned).
  - The probability of x_i is the sum over all classes, $P(x_i|\theta) = \Sigma_{j=1}^{K} P(x_i|c_j, \theta_j) P(c_j)$

Soft Clustering with Gaussian Mixture Model

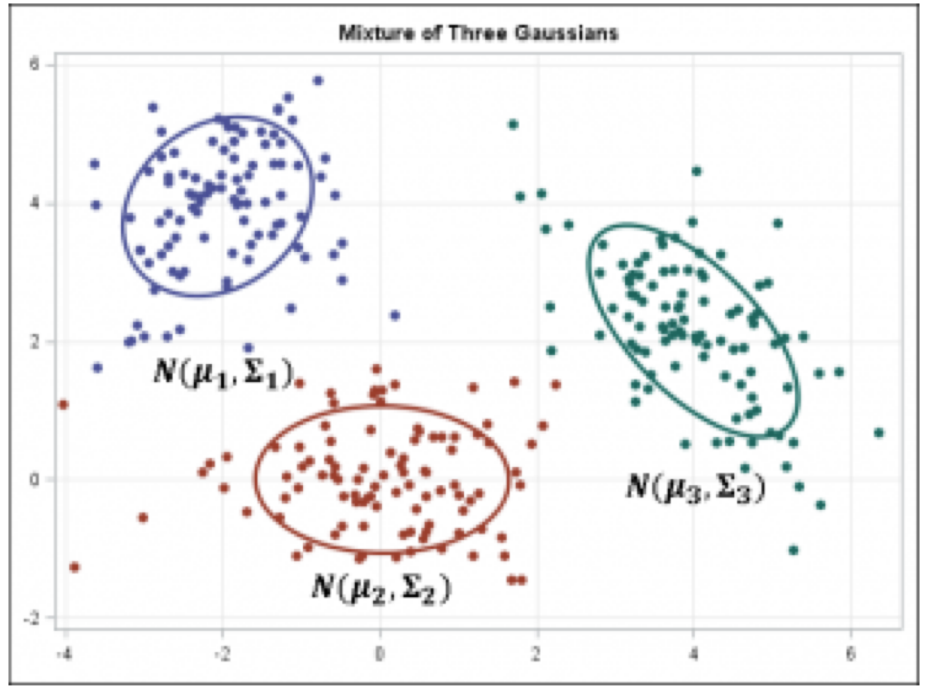
- Every object i is assigned to one cluster j with a probability
  - $P(z_i = j) \in [0,1]$ and $\Sigma_j P(z_i = j) = 1$
  - Where z_i is a hidden variable of which cluster x_i belongs to.

Assume the parameters of the GMM have been learned

- The probability of x_i belonging to cluster c_j :
  - $P(z_i = c_j|x_i) \propto P(x_i, z_i = c_j)$
    $= w_j P(x_i|z_i = c_j)$

Cluster prior probabilities

Probability density function of each cluster



The E-M(Expectation Maximization) Algorithm

- A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models.
- Expectation Step:
  - Assigns objects to clusters according to the current soft clustering or parameters of probabilistic clusters
  - $w_{ij}^{t+1} = P\left(z_i = j \middle| x_i, \theta_j^t\right) \propto w_j P\left(x_i \middle| z_i = j, \theta_j^t\right)$

Joint probability of x_i and its cluster c_j

- Maximization Step:
  - finds the new parameters of each cluster that maximize the expected likelihood
  - $\theta_{t+1} = argmax_\theta \Sigma_i \Sigma_j w_{ij}^{t+1} logL(x_i, z_j|\theta)$

Example: Applying E-M algorithm to 1-D GMM

- Iteratively do the following two steps
  - E-Step: Evaluate the soft clustering probability according to $\mu_j^t, \sigma_j^t, w_j^t$
    - $w_{ij}^{t+1} = \frac{w_j^t P\left(x_i \middle| \mu_j^t, \sigma_j^t\right)}{\Sigma_k w_k^t P\left(x_i \middle| \mu_k^t, \sigma_k^t\right)}$

In K-Means: Given centers $\{C_j\}$, $m_{i,j} = \begin{cases} 1 & j = \arg\min_k (x_i - C_j)^2 \\ 0 & \text{otherwise} \end{cases}$

  - M-Step: Find the new parameters μ_j^t, σ_j^t that maximize log likelihood. In Gaussian distribution, this is equivalent to do parameter estimation when each data point has a weight.
    - $\mu_j^{t+1} = \frac{\Sigma_i w_{ij}^{t+1} x_i}{\Sigma_i w_{ij}^{t+1}}, \left(\sigma_j^2\right)^{t+1} = \frac{\Sigma_i w_{ij}^{t+1} (x_i - \mu_j^{t+1})^2}{\Sigma_i w_{ij}^{t+1}}$

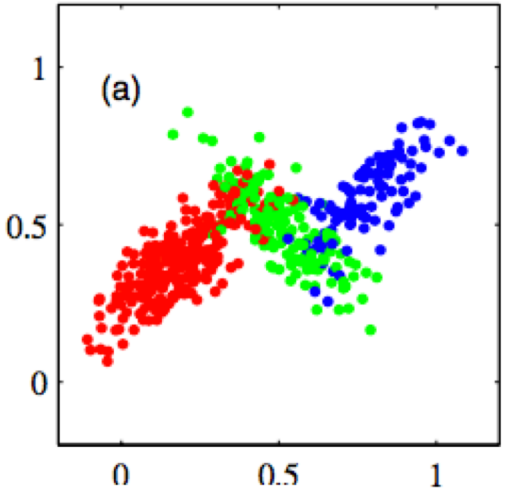
Weighted average means and variance

    - $w_j^{t+1} = \frac{\Sigma_i w_{ij}^{t+1}}{n}$

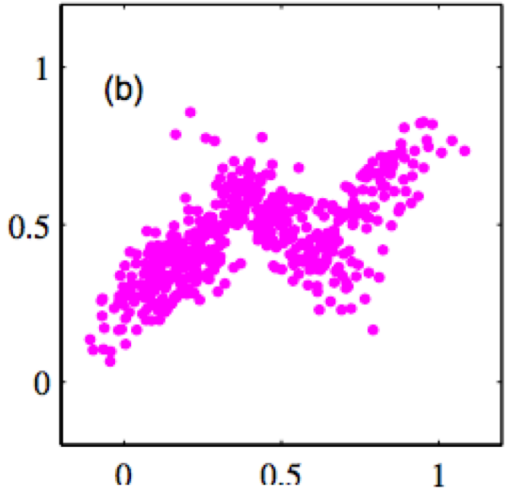
In K-Means: Given memberships $\{m_{i,j}\}$, $C_j = \frac{\sum_{i=1}^{n} m_{i,j} x_i}{\sum_{i=1}^{n} m_{i,j}}$

Gaussian Mixture Model

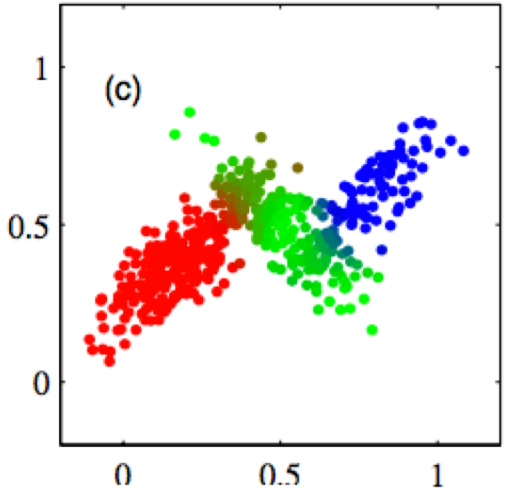
- Example of applying Gaussian Mixture Model



The data points belong to three classes. Each class follows a Gaussian distribution.



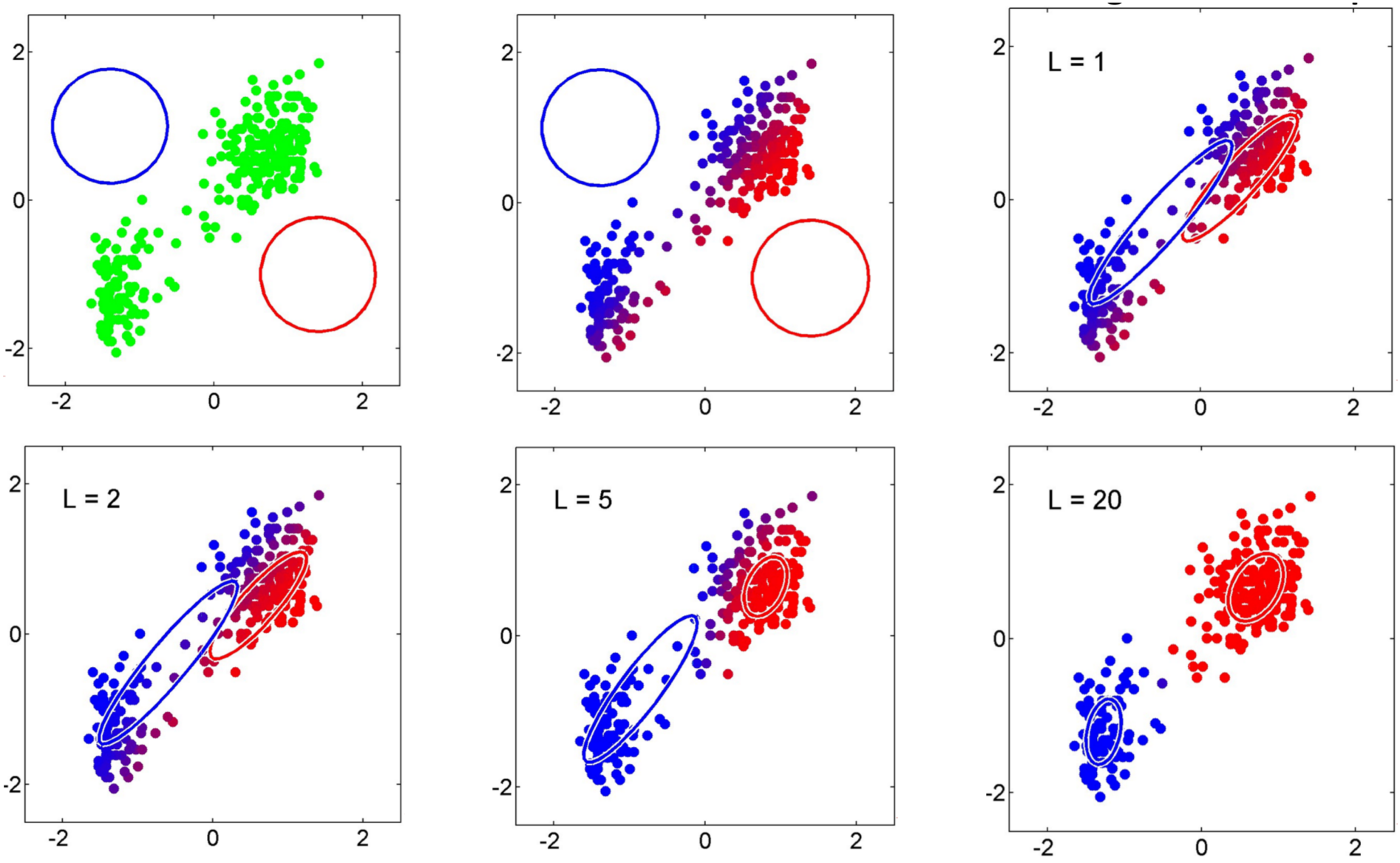
We hide the class information.



Cluster information inferred by GMM.

- We can use E-M algorithm to learn the parameters.

Example: Applying E-M algorithm to GMM



Gaussian Mixture Model – Strength and Weakness

- Advantages
  - Mixture models are more general than partitioning: different densities and sizes of clusters
  - Clusters can be characterized by a small number of parameters
  - The results satisfy the statistical assumptions of generative models
- Disadvantages
  - Converge to local optimal ⇐ Overcome it by running multi-times w. random initialization
  - Computationally more expensive
  - Hard to estimate the number of clusters
  - Can only deal with spherical clusters

K-means vs. GMM

Remember in GMM, $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$, where $\pi_k = p(z_k = 1)$ is the prior for the k^{th} component; and μ_k,Σ_k are the mean and covariance matrix for k^{th} component respectively. In the E-step, we will update $p(z_k = 1|x_n) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}$

Now suppose that

(1) $\Sigma_k = \epsilon \mathbf{I}$ where ϵ is some *given* number;

(2) $\pi_k \neq 0$ $(k = 1, ..., K)$;

(3) $\|x_n - \mu_i\| \neq \|x_n - \mu_j\|$ for any $i \neq j$.

Under the above assumptions, prove that when $\epsilon \to 0$, $p(z_k = 1|x_n) = r_{n,k}$, where $r_{n,k}$ is the cluster assignment used in Kmeans.

Mixture Model for Doc Clustering

- A set of language models $\Theta = \{\theta_1, \theta_2, ..., \theta_K\}$
  - $\theta_i = \{p(w_1 \mid \theta_i), p(w_2 \mid \theta_i), ..., p(w_V \mid \theta_i)\}$

Mixture Model for Doc Clustering

- A set of language models $\Theta = \{\theta_1, \theta_2, ..., \theta_K\}$
  - $\theta_i = \{p(w_1 \mid \theta_i), p(w_2 \mid \theta_i), ..., p(w_V \mid \theta_i)\}$
- Probability $p(d = d_i)$

$$p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j)$$

Mixture Model for Doc Clustering

- A set of language models $\Theta = \{\theta_1, \theta_2, ..., \theta_K\}$
  - $\theta_i = \{p(w_1 \mid \theta_i), p(w_2 \mid \theta_i), ..., p(w_V \mid \theta_i)\}$
- Probability $p(d = d_i)$

$$p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j)$$

$$= \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i \mid \theta = \theta_j)$$

Mixture Model for Doc Clustering

- A set of language models
  - $\theta_i = \{p(w_1 \mid \theta_i), p(w_2 \mid \theta_i$
- Probability $p(d = d_i)$

Introduce hidden variable z_{ij}
z_{ij}: document d_i is generated by the j-th language model θ_j.

$$p(d = d_i) = \sum_{\theta_j} p(d = d_i, \theta = \theta_j)$$

$$= \sum_{\theta_j} p(\theta = \theta_j) p(d = d_i \mid \theta = \theta_j)$$

$$\propto \sum_{\theta_j} p(\theta = \theta_j) \prod_{k=1}^{V} \left[p(w_k \mid \theta_j) \right]^{tf(w_k, d_i)}$$

Learning a Mixture Model for Doc Clustering

E-Step

$$E[z_{ij}] = p(\theta = \theta_j \mid d = d_i)$$

$$= \frac{p(d = d_i \mid \theta = \theta_j)\, p(\theta = \theta_j)}{\sum_{n=1}^{K} p(d = d_i \mid \theta = \theta_n)\, p(\theta = \theta_n)}$$

K: # of clusters
V: # of terms
N: # of docs

$$= \frac{\prod_{m=1}^{V} \left[p(w_m \mid \theta_j) \right]^{tf(w_k, d_i)} p(\theta = \theta_j)}{\sum_{n=1}^{K} \prod_{m=1}^{V} \left[p(w_m \mid \theta_n) \right]^{tf(w_k, d_i)} p(\theta = \theta_n)}$$

In GMM

$$w_{ij}^{t+1} = \frac{w_j^t P\left(x_i \middle| \mu_j^t, \sigma_j^t\right)}{\Sigma_k w_k^t P\left(x_i \middle| \mu_k^t, \sigma_k^t\right)}$$

Learning a Mixture Model for Doc Clustering

M-Step

$$p(w_i \mid \theta_j) \leftarrow \frac{\sum_{k=1}^{N} E[z_{kj}]\, tf(w_i, d_k)}{\sum_{k=1}^{N} E[z_{kj}]\, |d_k|}$$

$$p(\theta = \theta_j) \leftarrow \frac{1}{N} \sum_{i=1}^{N} E[z_{ij}]$$

N: number of documents

In GMM

$$\mu_j^{t+1} = \frac{\Sigma_i w_{ij}^{t+1} x_i}{\Sigma_i w_{ij}^{t+1}}, \left(\sigma_j^2\right)^{t+1} = \frac{\Sigma_i w_{ij}^{t+1} (x_i - \mu_j^{t+1})^2}{\Sigma_i w_{ij}^{t+1}}$$

$$w_j^{t+1} = \frac{\Sigma_i w_{ij}^{t+1}}{n}$$

A Numerical Example

	data	info	retrieval	brain	lung	and
D1	1	1	1	0	0	0
D2	2	2	2	0	0	0
D3	1	1	1	0	0	2
D4	4	4	4	0	0	2
D5	0	0	0	2	2	2
D6	0	0	0	3	3	2
D7	0	0	0	1	1	0

- Given a collection of 7 documents (D1-D7), assume our vocabulary consists of 6 words {'data', 'info', 'retrieval', 'brain', 'lung', 'and'}
- The entry in the left table indicates the frequency of the corresponding word in the corresponding document

Suppose that at the previous iteration, we have the following estimation for the two language models: $P(\theta_1) = P(\theta_2) = 0.5$, $\theta_1 = [0.3,\ 0.3,\ 0.3,\ 0,\ 0,\ 0.1]$ and $\theta_2 = [0,\ 0,\ 0,\ 0.4,\ 0.4,\ 0.2]$. We also have the following estimation for the cluster membership $E[z_{ij}]$ $(i = 1, ..., 7;\ j = 1, 2$ (i.e., $E[z_{ij}]$ is the probability that the $i-th$ document belongs to $j-th$ cluster):

1	0
1	0
0.8	0.2
0.8	0.2
0.1	0.9
0.1	0.9
0	1

Table 1: $E[z_{ij}]$

- What is $P(d = d_1)$? (You can use the following fact: $0^0 = 1$, $1^0 = 1$ and $0^1 = 0$).

  We run EM algorithm one iteration. What is the updated $E[z_{ij}]$ $(i = 1, ..., 7;\ j = 1, 2$ after the E-step?

- What is the updated $P(\theta_1)$, $P(\theta_2)$, θ_1, and θ_2, respectively, after the M-step?

Discussions: K-means, GMM, and Mixture Model for Doc Clustering

- What are the commonalities among them?
- What are the differences among them?

Cluster Analysis

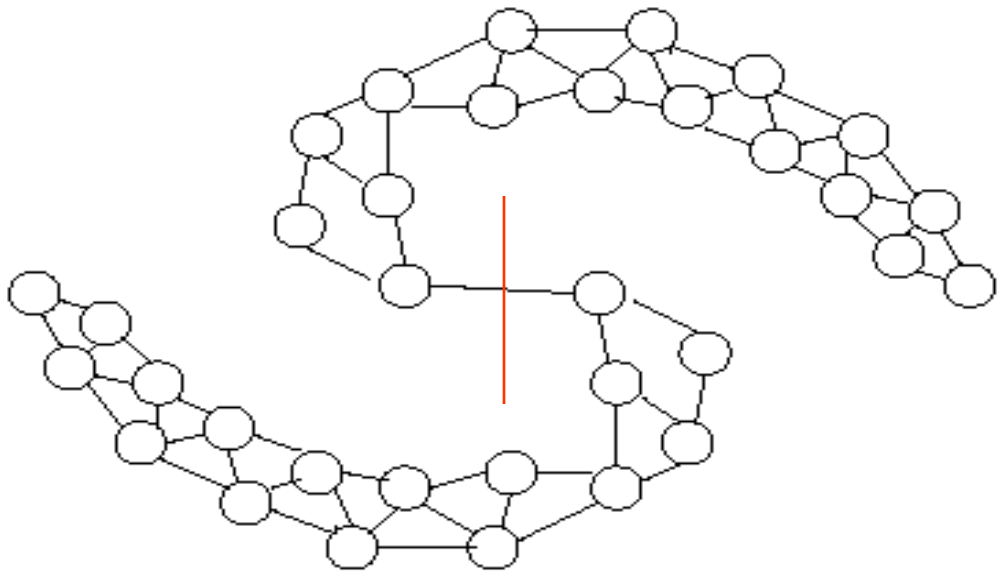
- Cluster Analysis Overview
- K-Means
- Mixture Models and E-M algorithm
- Spectral Methods
- Summary

Problems (I)

- Both k-means and mixture models take the feature/vector representation of the data as input (think of the data matrix X)
- What if the input data is NOT represented in feature/vector, format?
  - E.g., graph, or only pair-wised distance.

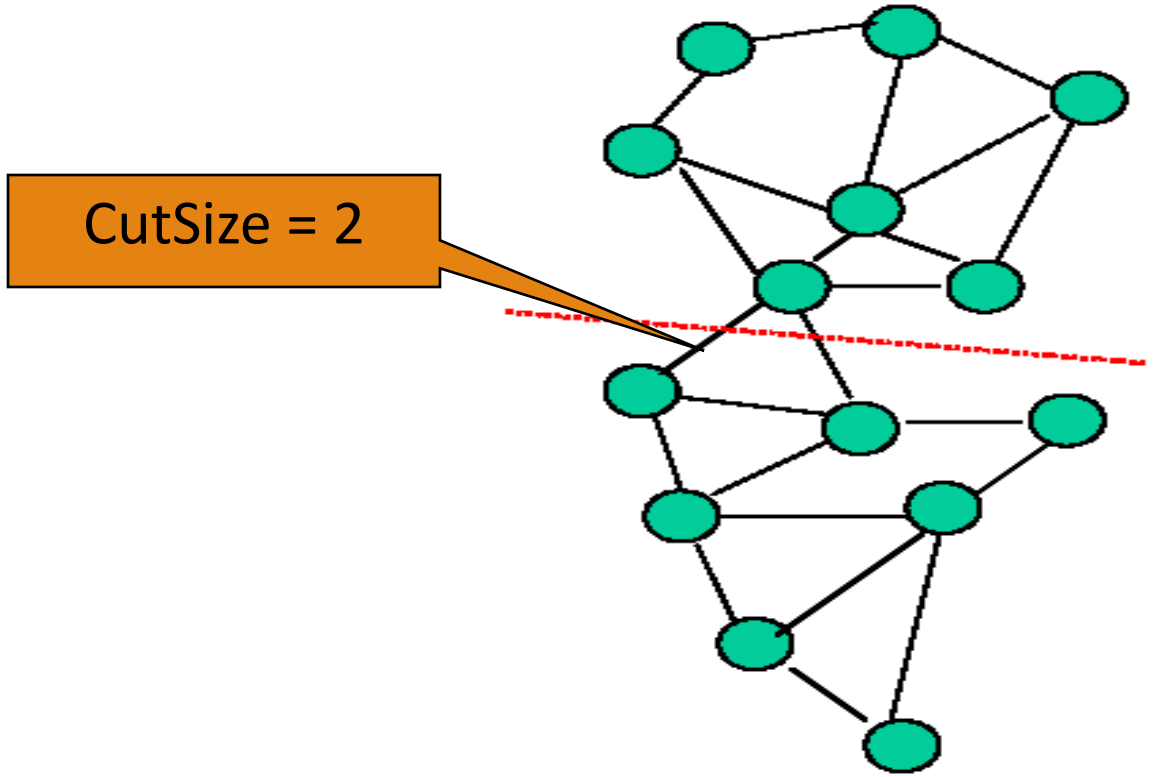
Problems (II)

- Both k-means and mixture models look for compact clustering structures
  - In some cases, connected clustering structures are more desirable



Graph Partition

- MinCut: bipartite graphs with minimal number of cut edges



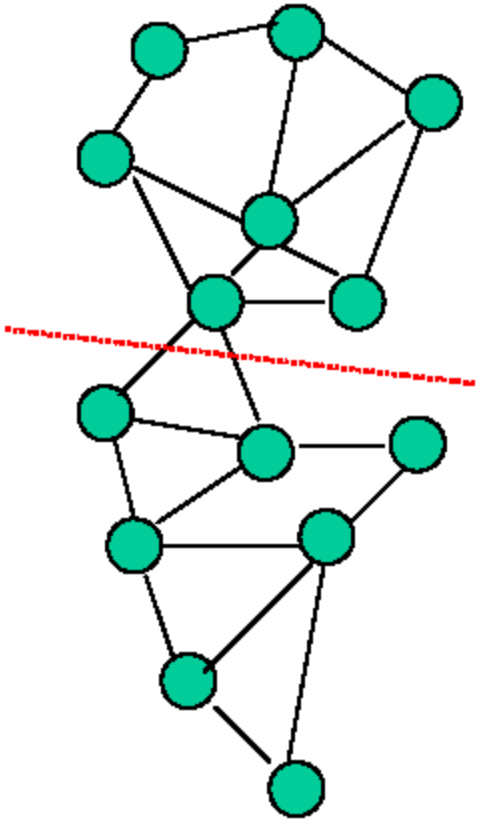
2-way Spectral Graph Partitioning

- Weight matrix **W**
  - $w_{i,j}$: the weight between two vertices i and j
- (Cluster) Membership vector **q**

$$q_i = \begin{cases} 1 & i \in \text{Cluster } A \\ -1 & i \in \text{Cluster } B \end{cases}$$

$$\mathbf{q} = \arg\min_{\mathbf{q} \in [-1,1]^n} CutSize$$

$$CutSize = J = \frac{1}{4} \sum_{i,j} \left(q_i - q_j \right)^2 w_{i,j}$$



Solving the Optimization Problem

$$\mathbf{q} = \underset{\mathbf{q}\in[-1,1]^n}{\arg\min} \frac{1}{4} \sum_{i,j} \left(q_i - q_j \right)^2 w_{i,j}$$

- Directly solving the above problem requires combinatorial search → exponential complexity
- How to reduce the computational complexity?

Relaxation Approach

- Key difficulty: q_i has to be either –1, 1
  - Relax q_i to be any real number
  - Impose constraint

$$\sum_{i=1}^{n} q_i^2 = n$$

$$J = \frac{1}{4}\sum_{i,j}\left(q_i - q_j\right)^2 w_{i,j} = \frac{1}{4}\sum_{i,j}\left(q_i^2 + q_j^2 - 2q_i q_j\right)w_{i,j}$$

$$= \frac{1}{4}\sum_i 2q_i^2\left(\sum_j w_{i,j}\right) - \frac{1}{4}\sum_{i,j} 2q_i q_j w_{i,j}$$

$$= \frac{1}{2}\sum_i q_i^2 d_i - \frac{1}{2}\sum_{i,j} q_i q_j w_{i,j} = \frac{1}{2}\sum_i q_i\left(d_i\delta_{i,j} - w_{i,j}\right)q_j$$

$$d_i \equiv \sum_j w_{i,j}$$

$$D \equiv \left[d_i\delta_{i,j}\right]$$

$$J = 1/2\,\mathbf{q}^T(\mathbf{D} - \mathbf{W})\mathbf{q}$$

Relaxation Approach

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} J = \arg\min_{\mathbf{q}} \mathbf{q}^T (\mathbf{D} - \mathbf{W})\mathbf{q}$$

$$\text{subject to } \sum_k q_k^2 = n$$

Relaxation Approach

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} J = \arg\min_{\mathbf{q}} \mathbf{q}^T (\mathbf{D} - \mathbf{W})\mathbf{q}$$

$$\text{subject to } \sum_k q_k^2 = n$$

- Solution: the second minimum eigenvector for **D-W**

$$(D - W)\mathbf{q} = \lambda_2 \mathbf{q}$$

Graph Laplacian

$$\mathbf{L} = \mathbf{D} - \mathbf{W}: \ \mathbf{W} = \left[w_{i,j} \right], \ \mathbf{D} = \left[\delta_{i,j} \left(\sum_j w_{i,j} \right) \right]$$

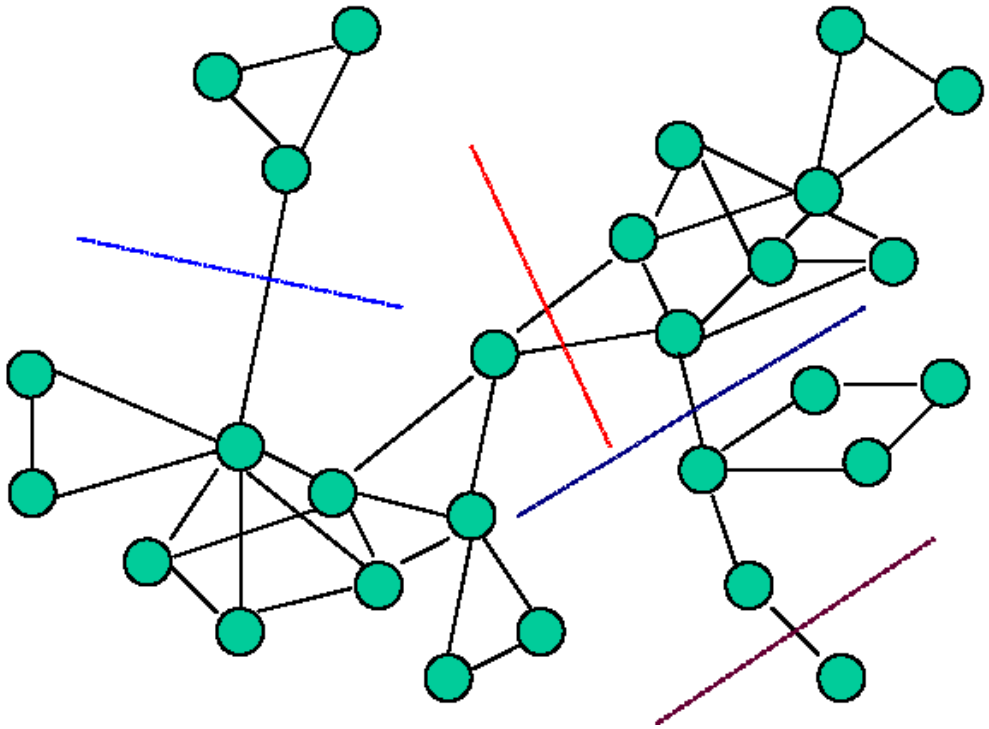
- **L** is semi-positive definitive matrix
  - For Any **x,** we have $\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0$, why?
  - Minimum eigenvalue $\lambda_1 = 0$ (what is the eigenvector?)
  - $0 = \lambda_1 < \lambda_2 < \lambda_3 \ldots < \lambda_k$
  - The eigenvector that corresponds to the second minimum eigenvalue λ_2 gives the best bipartite graph partition

Recovering Partitions

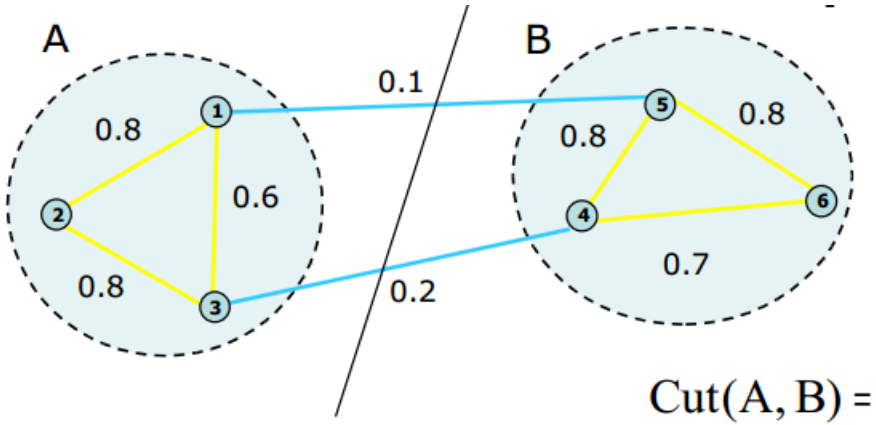
- Due to the relaxation, **q** can be any number (not just –1 and 1)
- How to construct partition based on the eigenvector?
  - Simple strategy: $A = \{i \mid q_i < 0\},\ B = \{i \mid q_i \geq 0\}$

Spectral Clustering

- Minimum cut does not balance the size of bipartite graphs



Basic Concepts



$$\text{Cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \Rightarrow \text{Cut}(A, B) = 0.3$$

$$\text{Cut}(A, A) = \sum_{i \in A, j \in A} w_{ij} \Rightarrow \text{Cut}(A, A) = 2.2$$

$$\text{Cut}(B, B) = \sum_{i \in B, j \in B} w_{ij} \Rightarrow \text{Cut}(B, B) = 2.3$$

$$\text{Vol}(A) = \sum_{i \in A} \sum_{j=1}^{n} w_{ij} \Rightarrow \text{Vol}(A) = 4.7$$

$$\text{Vol}(B) = \sum_{i \in B} \sum_{j=1}^{n} w_{ij} \Rightarrow \text{Vol}(B) = 4.9$$

$|A| = |B| = 3$

From MinCut to NCut

- Objective #1: minimize inter-cluster connections
  - Min cut (A, B)
- Objective #2: maximize intra-cluster connections
  - Max vol (A, A) and vol (B, B)
- Overall objective, to minimize

$$J_{NCut}(A,B) = Cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{Vol(B)}\right)$$

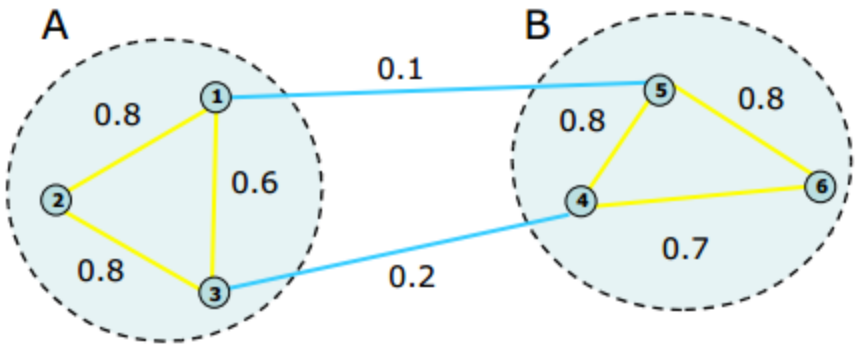
- Solution: the 2nd smallest eigenvector of

$$(D - W)y = \lambda Dy$$

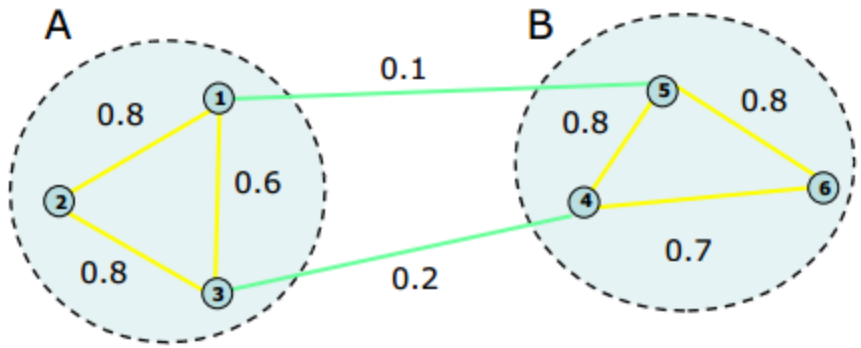
$$L_{NCut} = D^{-1/2}(D - W)D^{-1/2}$$

- Variants
  - RatioCut $J_{RatioCut}(A,B) = Cut(A,B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$
  - MinMaxCut $J_{MinMaxCut}(A,B) = Cut(A,B)\left(\frac{1}{Cut(A,A)} + \frac{1}{Cut(B,B)}\right)$

An Illustrative, Numerical Example



Graph and similarity matrix



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0.8	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

An Illustrative, Numerical Example

Pre-processing

Build Laplacian matrix L of the graph

x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	-0.8	0	-0.2	2.5	-0.8	-0.7
x_5	-0.1	0	0	0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

Decomposition : Find

- eigenvalues Λ and
- eigenvectors X of matrix L

$\Lambda =$

0.0
0.3
2.2
2.3
2.5
3.0

$X =$

0.4	0.2	0.1	0.4	-0.2	-0.9
0.4	0.2	0.1	-0.	0.4	0.3
0.4	0.2	-0.2	0.0	-0.2	0.6
0.4	-0.4	0.9	0.2	-0.4	-0.6
0.4	-0.7	-0.4	-0.8	-0.6	-0.2
0.4	-0.7	-0.2	0.5	0.8	0.9

- Map vertices to the corresponding components of 2nd eigenvector

x_1	0.2
x_2	0.2
x_3	0.2
x_4	-0.4
x_5	-0.7
x_6	-0.7

How do we find the clusters?

Spectral Clustering Algorithms
(continued)

x_1	0.2
x_2	0.2
x_3	0.2
x_4	-0.4
x_5	-0.7
x_6	-0.7

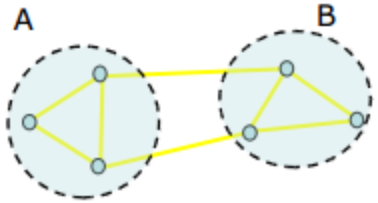
Split at value 0

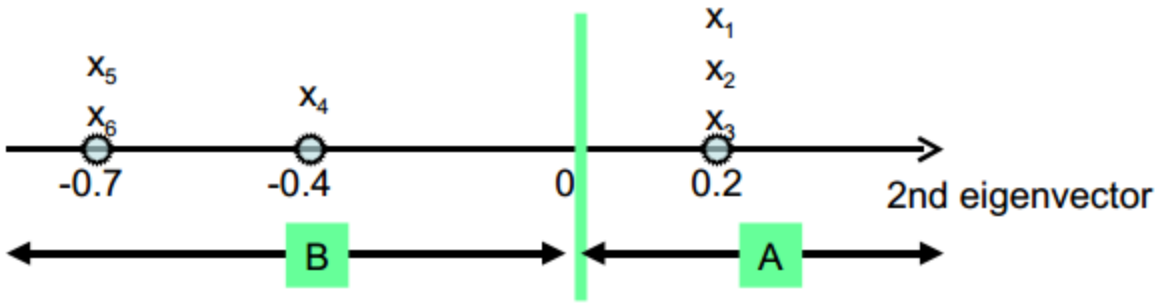
Cluster A: Positive points

Cluster B: Negative points

x_1	0.2
x_2	0.2
x_3	0.2

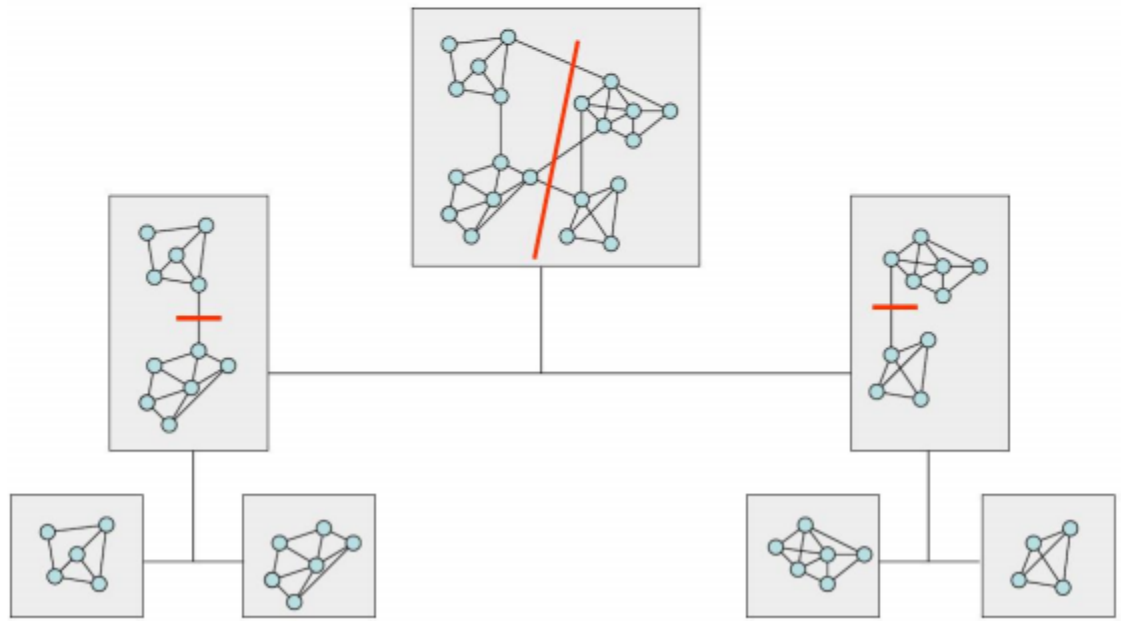
x_4	-0.4
x_5	-0.7
x_6	-0.7





Recursive bi-partitioning

- Recursively apply bi-partitioning algorithm in a hierarchical divisive manner.
- Disadvantages: Inefficient, unstable



k-way graph cuts

In order to partition a dataset or graph into k classes, two basic approaches can be used:

- Recursive bi-partitioning: The basic idea is to recursively apply bi-partitioning algorithm in a hierarchical way: after partitioning the graph into two, reapply the same procedure to the subgraphs. The number of groups is supposed to be given or directly controlled by the threshold allowed to the objective function.
- k-way partitioning: The 2-way objective functions can be generalized to take into consideration more than two clusters:

$$J_{RatioCut}(A_1,..,A_k) = \sum_{i=1}^{k} \frac{Cut(A_i,\overline{A}_i)}{|A_i|}$$

$$J_{NCut}(A_1,..,A_k) = \sum_{i=1}^{k} \frac{Cut(A_i,\overline{A}_i)}{Vol(A_i)}$$

$$J_{MinMaxCut}(A_1,..,A_k) = \sum_{i=1}^{k} \frac{Cut(A_i,\overline{A}_i)}{Cut(A_i,A_i)}$$

Spectral Clustering Summary

In general, the spectral clustering methods can be divided to three main varieties since the basic spectral algorithm is itself divided to three steps.

2. Preprocessing: Spectral clustering methods can be best interpreted as tools for analysis of the block structure of the similarity matrix. So, building such matrices may certainly ameliorate the results.
   - Calculation of the similarity matrix is not evident.
   - Choosing the similarity function can highly affect the results of the following steps. In most cases, the Gaussian kernel is chosen, while other similarities like cosine similarity are used for specific applications.
3. Graph and similarity matrix construction: Laplacian matrices are generally chosen to be positive and semi-definite thus their eigenvalues will be non-negatives. The most used Laplacian matrices are summarized in the following.

Unnormalized	$L = D - W$
Symmetric	$L_{Sy} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$
Asymmetric	$L_{As} = D^{-1} L = I - D^{-1} W$

9. Clustering: simple algorithms other than k-means can be used in the last stage such as simple linkage, k-lines, elongated k-means, mixture model, etc.

Spectral Clustering – Scale-up

- Q: how to compute the top-k eigen-pairs of a given matrix *n* x *n* *A*?
- A: Nyström method

$$\int_a^b h(x)\,\mathrm{d}x \approx \sum_{k=1}^{n} w_k h(x_k)$$

  - Think of matrix-vector multiplication as “integration”
  - Use sampling for speed-up

Nyström Method (1924)

- Goal: find approximate top-k eigen-pairs of matrix A: n x n matrix
- Method (sketch)
  - Step 1: Subsample it to form an n' x n' matrix A'
  - Step 2: Do spectral clustering on A'
  - Step 3: Extend clustering on A' to A

Nyström Method: Details

- Step 1: Subsample it to form an n' x n' matrix A'
  - Sample a row/column every r rows/columns

$\mathbf{A}$ is a permutation Π of $\begin{pmatrix} \mathbf{A}' & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}$

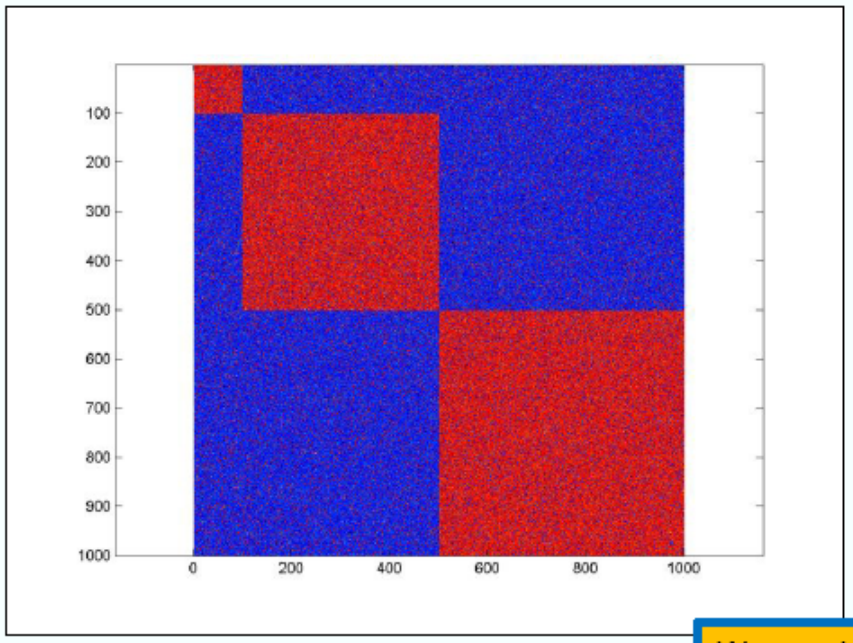
- Step 2: Do spectral clustering on A'
  - U', Λ': top-k eigenvectors and eigenvalues of A'
- Step 3: Extend clustering on A' to A
  - U, Λ: approximated top-k eigenvectors and eigenvalues of A'
  - Equivalent to approximate

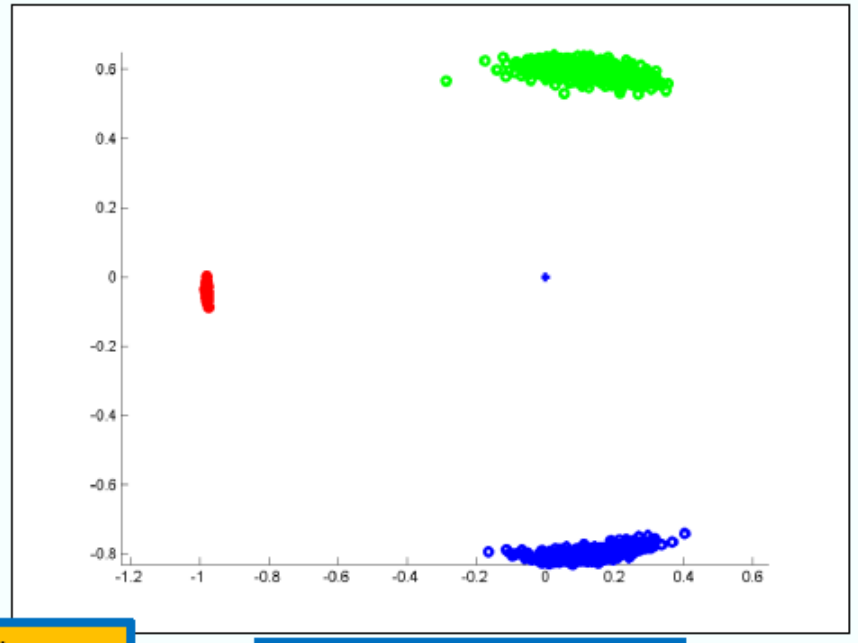
$$C \approx B^T A'^{-1} B$$

$$U = \begin{pmatrix} A' \\ B^T \end{pmatrix} U'(\Lambda')^{-1}$$

$$\Lambda = \Lambda'$$

Nyström Method – An Example



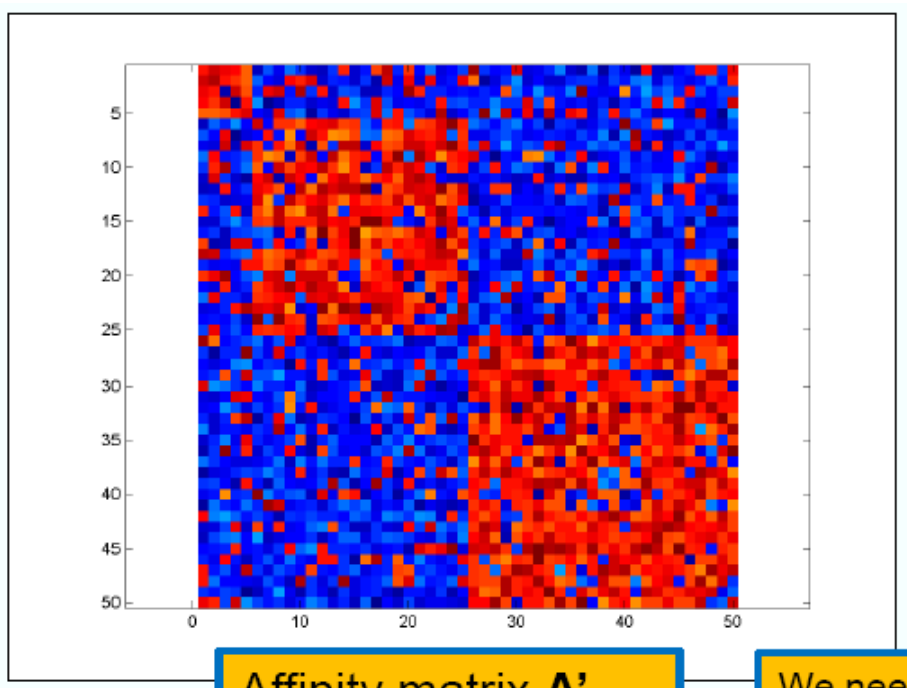


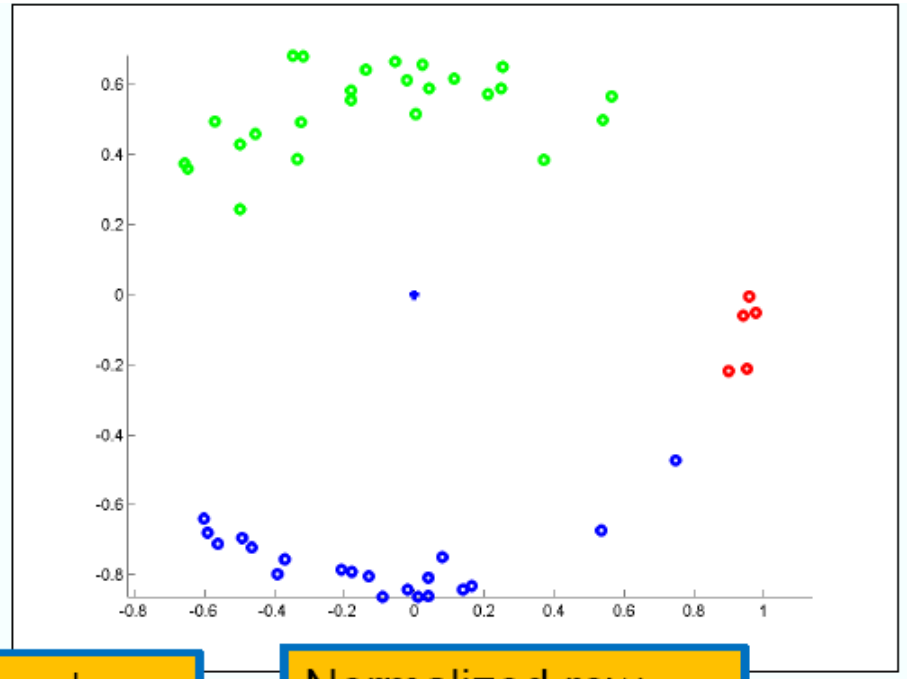
Affinity matrix **A**
1000 points

We need to solve a 1000×1000 eigenvalue problem to get here

Normalized row space of **U**

Nyström Method – An Example





Affinity matrix **A'**
50 points

We need to solve a 50×50 eigenvalue problem to get here

Normalized row space of **U'**

Cluster Analysis

- Cluster Analysis Overview
- K-Means
- Mixture Models and E-M algorithm
- Spectral Methods
- Summary

Cluster Analysis

- Cluster Analysis Overview
- K-Means
- Mixture Models and E-M algorithm
- Spectral Methods
- More on Clustering (not covered)
  - Hierarchical Methods
  - Density- and Grid-Based Methods
  - Matrix Factorization-Based Methods
  - Co-clustering
  - Compression-Based Methods
  - …

References: (I) Cluster Analysis: An Introduction

- Jiawei Han, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd ed. , 2011 (Chapters 10 & 11)
- Charu Aggarwal and Chandran K. Reddy (eds.). Data Clustering: Algorithms and Applications. CRC Press, 2014
- Mohammed J. Zaki and Wagner Meira, Jr.. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, 2014
- L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 1990
- Charu Aggarwal. An Introduction to Clustering Analysis. *in* Aggarwal and Reddy (eds.). Data Clustering: Algorithms and Applications (Chapter 1). CRC Press, 2014

References: (II) Partitioning Methods

- J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, 1967
- S. Lloyd. Least Squares Quantization in PCM. *IEEE Trans. on Information Theory*, 28(2), 1982
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988
- R. Ng and J. Han. Efficient and Effective Clustering Method for Spatial Data Mining. VLDB'94
- B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural computation*, 10(5):1299–1319, 1998
- I. S. Dhillon, Y. Guan, and B. Kulis. Kernel K-Means: Spectral Clustering and Normalized Cuts. *KDD*'04
- D. Arthur and S. Vassilvitskii. K-means++: The Advantages of Careful Seeding. SODA'07
- C. K. Reddy and B. Vinzamuri. A Survey of Partitional and Hierarchical Clustering Algorithms, in (Chap. 4) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014

References: (III) Hierarchical Methods

- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. SIGMOD'96
- S. Guha, R. Rastogi, and K. Shim. Cure: An Efficient Clustering Algorithm for Large Databases. SIGMOD'98
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *COMPUTER*, 32(8): 68-75, 1999.
- C. K. Reddy and B. Vinzamuri. A Survey of Partitional and Hierarchical Clustering Algorithms, in (Chap. 4) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014

References: (IV) Density- and Grid-Based Methods

- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases. KDD'96
- W. Wang, J. Yang, R. Muntz, STING: A Statistical Information Grid Approach to Spatial Data Mining, VLDB'97
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. SIGMOD'98
- A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98
- M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering Points to Identify the Clustering Structure. SIGMOD'99
- M. Ester. Density-Based Clustering. In (Chapter 5) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications . CRC Press. 2014
- W. Cheng, W. Wang, and S. Batista. Grid-based Clustering. In (Chapter 6) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press. 2014

References: (IV) Evaluation of Clustering

- M. J. Zaki and W. Meira, Jr.. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, 2014
- L. Hubert and P. Arabie. Comparing Partitions. *Journal of Classification*, 2:193–218, 1985
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988
- M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Info. Systems*, 17(2-3):107–145, 2001
- J. Han, M. Kamber, and J. Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd ed. , 2011
- H. Xiong and Z. Li. Clustering Validation Measures. in (Chapter 23) C. Aggarwal and C. K. Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press, 2014