Inspire…Educate…Transform.

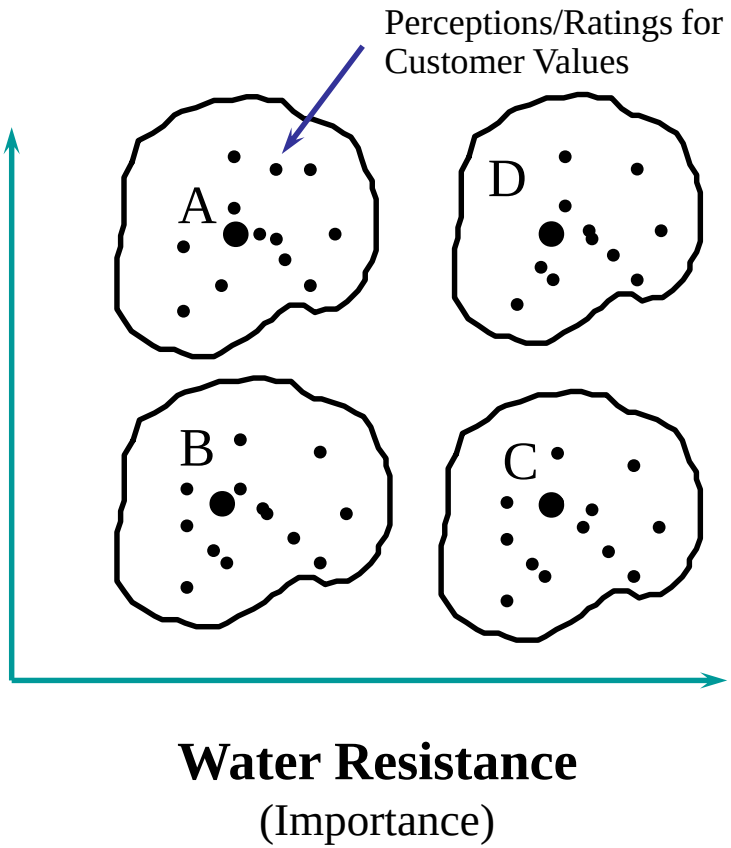
Clustering

Dr. Manoj Chinnakotla
Senior Applied Scientist, Microsoft
Adjunct Professor, IIIT Hyderabad

Today's Agenda

- Unsupervised Learning
- Hard Clustering
  - K-Means
- Hierarchical Clustering
- Soft Clustering
  - Expectation Maximization (EM)
- Spectral Clustering
- Conclusion

Market Segmentation Problem – Carpet Fiber



A,B,C,D:
Location of segment centers.

Typical Members:

A: Schools
B: Light commercial
C: Indoor/outdoor Carpeting
D: Health clubs

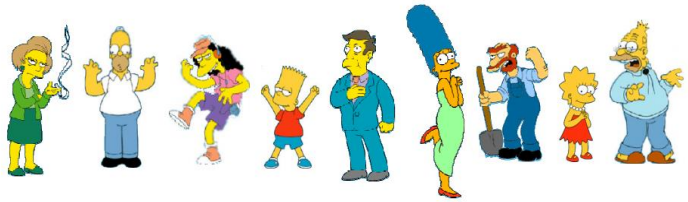
The Task
Automatically Discover the underlying structure in the data

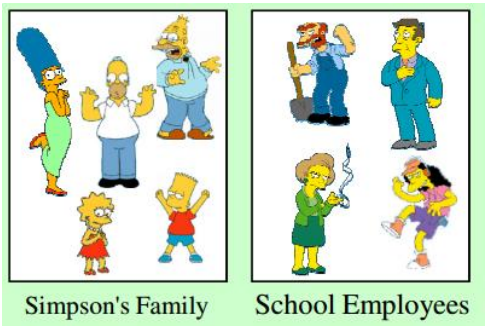
The Challenge
No Training Data available

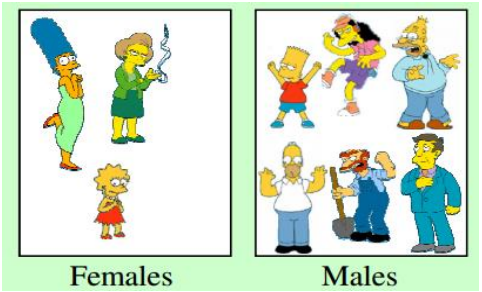
What is Clustering?

- Clustering – Grouping objects together based on similarity
- Unsupervised Learning – No predefined classes

What is a natural grouping among these objects?







Clustering is Subjective

CSE 7306c

Classification vs. Clustering

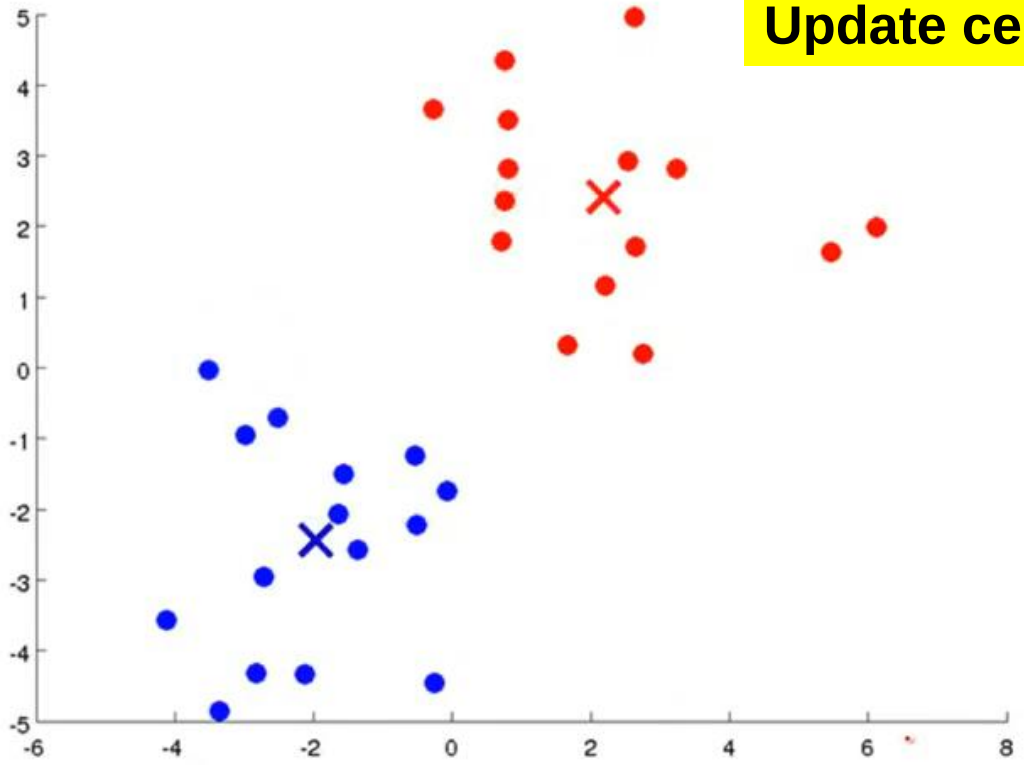
	Classification	Clustering
Cost (or Loss) $\mathcal{L}$	Expectd error	many! (probabilistic or not)
	Supervised	Unsupervised
Generalization	Performance on new data is what matters	Performance on current data is what matters
K	Known	Unknown
"Goal"	Prediction	Exploration Lots of data to explore!

K-Means

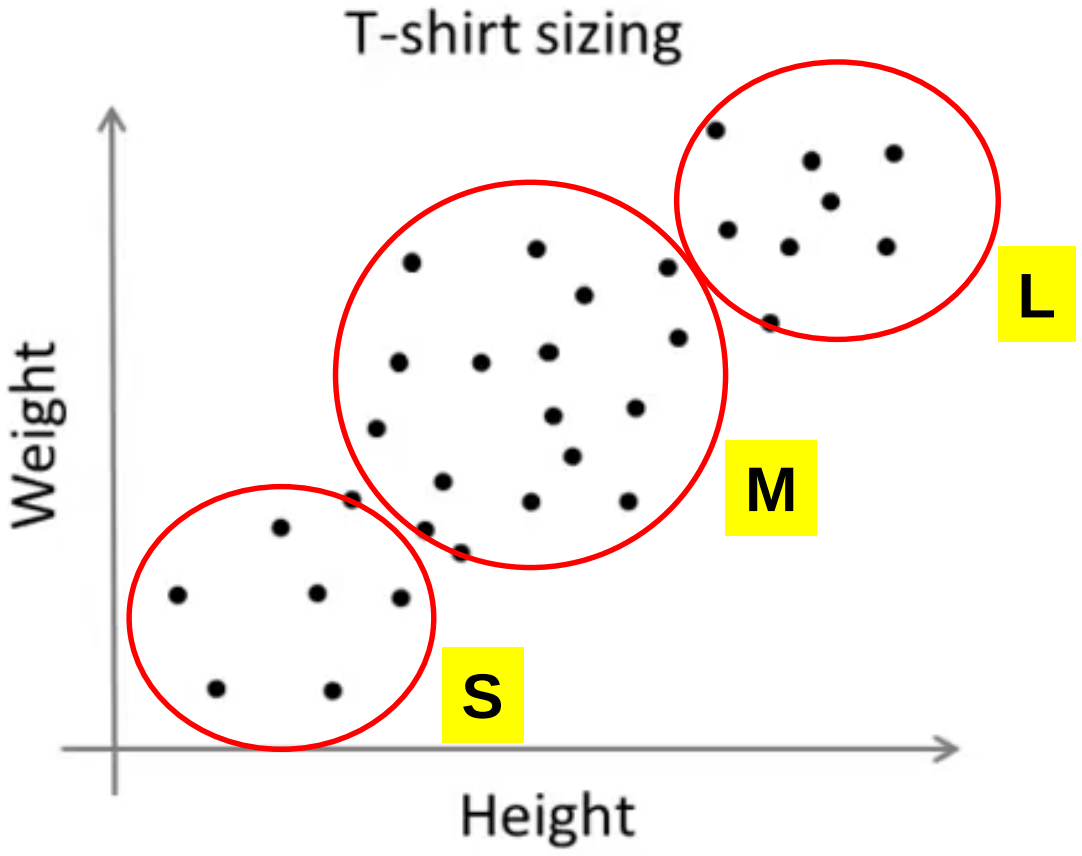
Randomly initialize cluster centroids

Cluster Assignment
Assign each point to a cluster with least distance to centroid

Update centroid of cluster



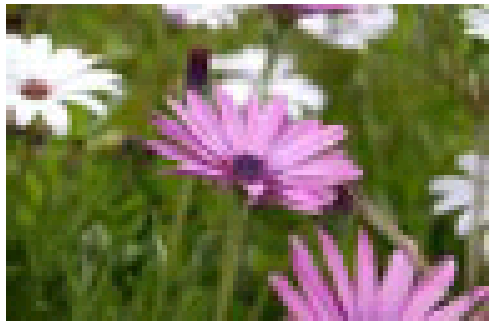
What if Clusters are Not Clearly Separable?



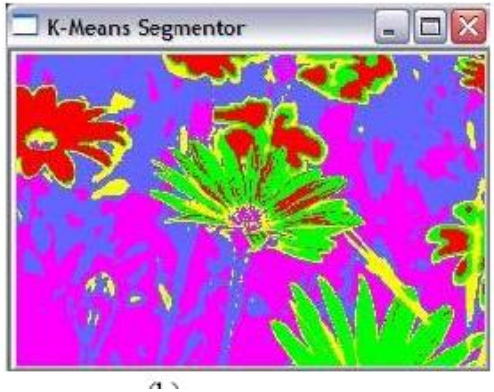
Sounds too simple?, it really works!

Image Segmentation

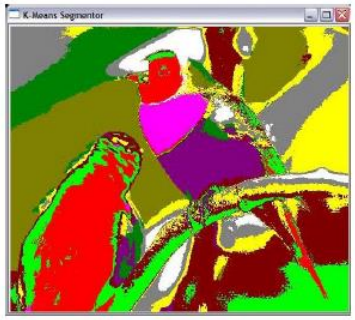
K=5, RGB spac



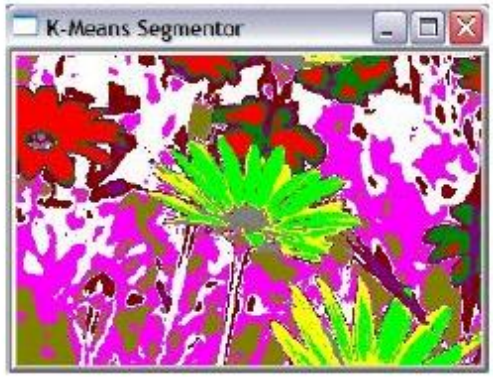
K=5, RGB space



K=10, RGB space



K=10, RGB space



K-Means Optimization Objective

$c^{(i)}$ = index of cluster (1,2,...,K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1,\ldots,\mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

Optimizing J is NP-Hard

Distortion Function

K-Means Optimization - Steps

- Cluster Assignment Step
  - Assuming cluster centroids are fixed, assign points to clusters
  - Choose $c^{(i)}$s, assuming $\mu_{c^{(i)}}$ constant
  - Assigning points to nearest centroids helps in ***minimizing J***

$$\frac{\partial J}{\partial c^{(i)}} \alpha \left\| x^{(i)} - \mu_{c^{(i)}} \right\|$$

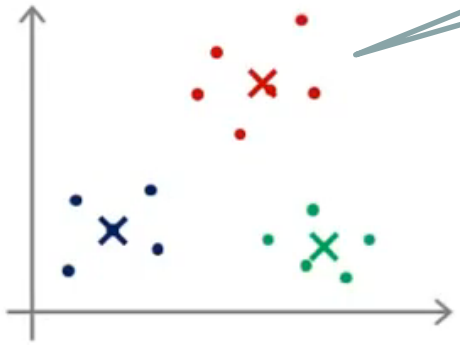
$$\min_{c^{(i)}} \frac{\partial J}{\partial c^{(i)}} = \min_{c^{(i)}} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|$$

- Centroid Update Step
  - Assuming points are assigned to clusters, choose a representative for cluster
  - Choose $\mu_{c^{(i)}}$ assuming $c^{(i)}$s constant
  - Choosing $\mu_{c^{(i)}}$ as the centroid of $x^{(i)}$s ***minimizes J***
    - Since "Mean" minimizes the sum of squares with all the points

$$\min_{\mu_{c^{(i)}}} \frac{\partial J}{\partial \mu_{c^{(i)}}}$$

Not guaranteed to reach global optimum
Keeps moving towards gradient

Preventing Local Optima

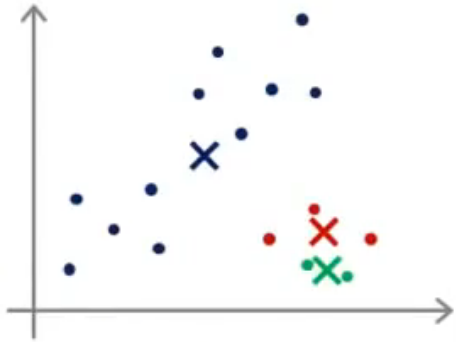


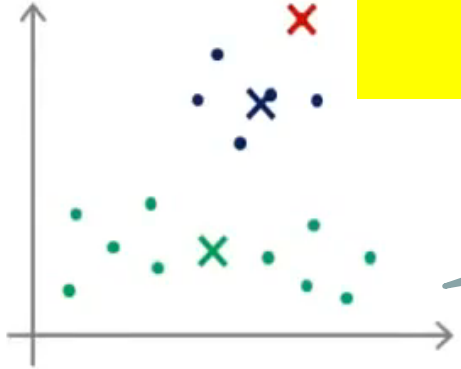
Good cluster

Randomly initialize K-means.
Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.
Compute cost function (distortion)
$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

Repeat above multiple times (usually 50-1000 times)

Choose the clustering assignment with $\mathbf{min}\, J$



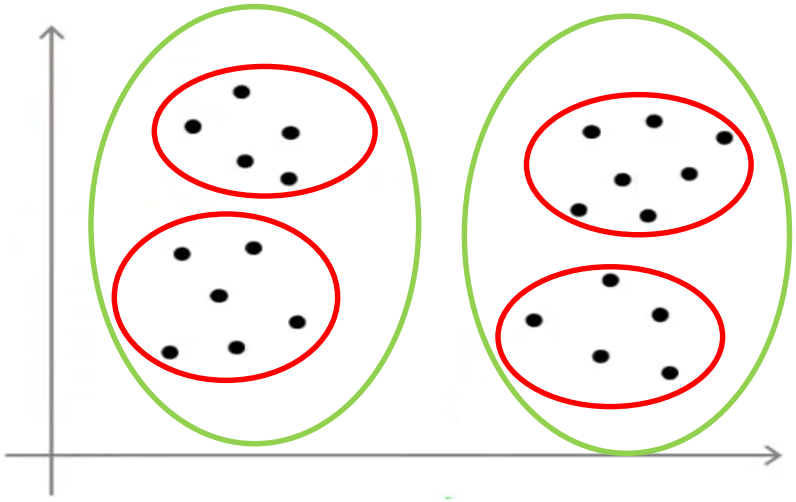


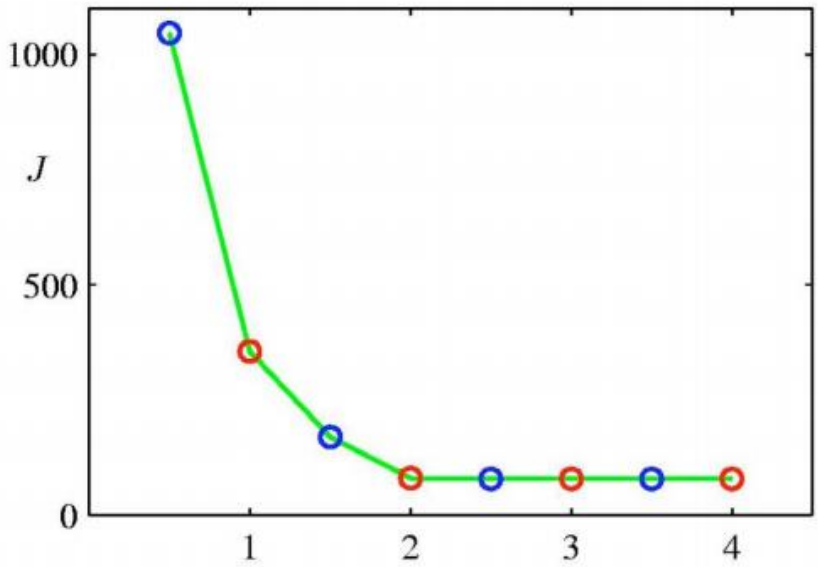
Sub-optimal clusters

CSE 7306c

How to choose K?

- Sometimes difficult due to genuine ambiguities regarding clusters
- Tradeoff – Distortion vs. Accuracy (if ground truth is known)
- In practice, chosen based on domain knowledge and constraints
- Some theories, rules of thumb exist in the absence of any information:
  - Elbow Method
  - Information Criterion Approach
  - Analyzing the Kernel Matrix





A Few Limitations of K-Means

- Only applicable to data where "mean" is well-defined
  - Restricts applicability to only Euclidean spaces
  - For example: if categorical attributes present (Example: Marital Status, Gender), "mean" not meaningful

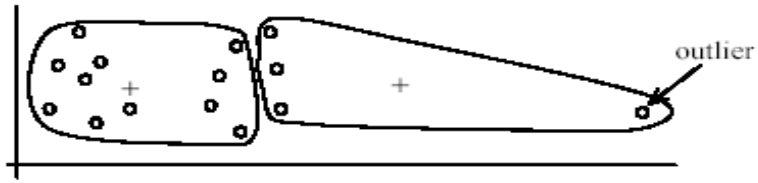
- K-Medoids – a minor variant
  - Choose most centrally located point within the cluster as representative

$$\underset{\{i:C(i)=k\}}{\operatorname{argmin}} \sum_{C(i')=k} D(x_i, x_{i'})$$

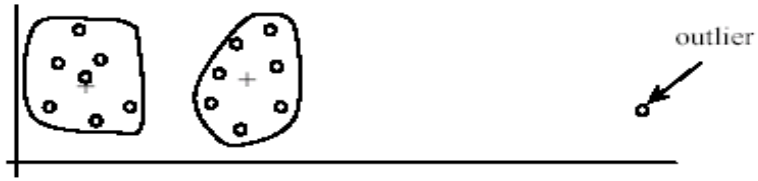
  - This step is more computationally intensive - $O\left(N_k^2\right)$
  - N_k is the number of points assigned to a cluster
  - For K-Means: $O(N_k)$

A Few Limitations of K-Means (Contd..)

- Sensitivity to outliers in data
  - Detect and remove outliers before clustering
  - K-Medians is relatively more robust to outliers

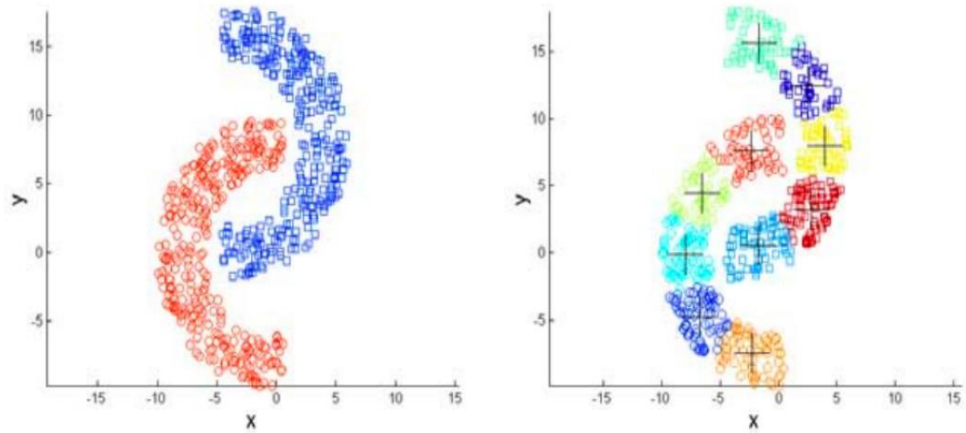


(A): Undesirable clusters



(B): Ideal clusters

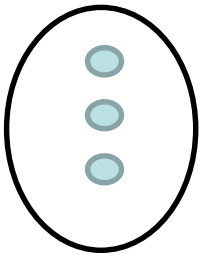
- Cannot find arbitrary shaped clusters
  - May occur sometimes in nature and data



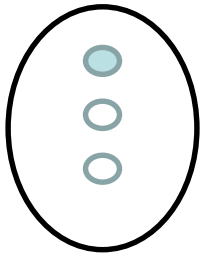
Evaluation of Cluster Quality

Extrinsic Evaluation – Given Ground Truth Data

Reference

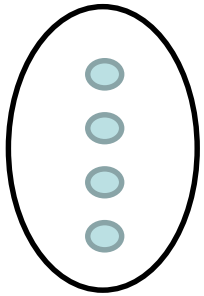


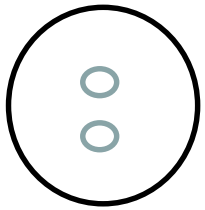
Precision = 3/3
Recall = 3/4



Precision = 2/3
Recall = 2/2

Truth





Proposed by Meila, 2003

$$AlgoPrecision = \sum_i \frac{|C_i|}{n} \max_j Precision(C_i, L_j)$$

$$Precision\ (C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$$

$$AlgoRecall = \sum_i \frac{|L_i|}{n} \max_j Recall\ (C_j, L_i)$$

$$Recall\ (C_j, L_i) = \frac{|C_j \cap L_i|}{|L_i|}$$

$$F = \sum_i \frac{|L_i|}{n} \max_j \{F(C_j, L_i)\}$$

F-Measure summarizes the performance

$$\frac{C_j, L_i) \times Precision(C_i, L_j)}{, L_i) + Precision(C_i, L_j)}$$

CSE 7306c

Evaluation of Cluster Quality (Contd..)

- Intrinsic Evaluation
  - When no gold standard data is available
  - Develop measures for some general goodness criterion
  - For example: Good clusters should ***high intra-cluster similarity*** and ***low inter-cluster similarity***
  - Davies-Bouldin (DB) Index

Avg. distance of all elements with centroid

Number of Clusters

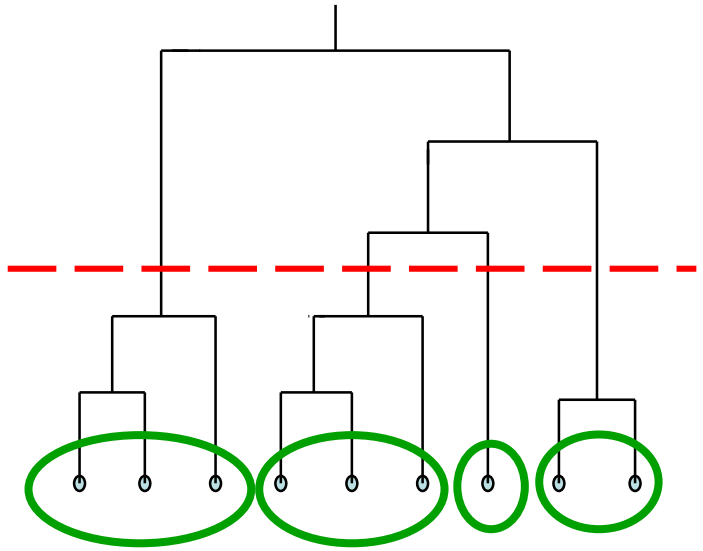
$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

Distance between centroids

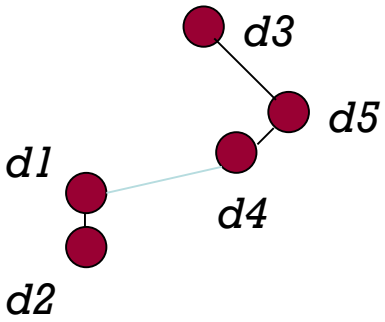
The lesser the DB index is – the better the quality of clusters

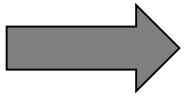
Hierarchical Clustering

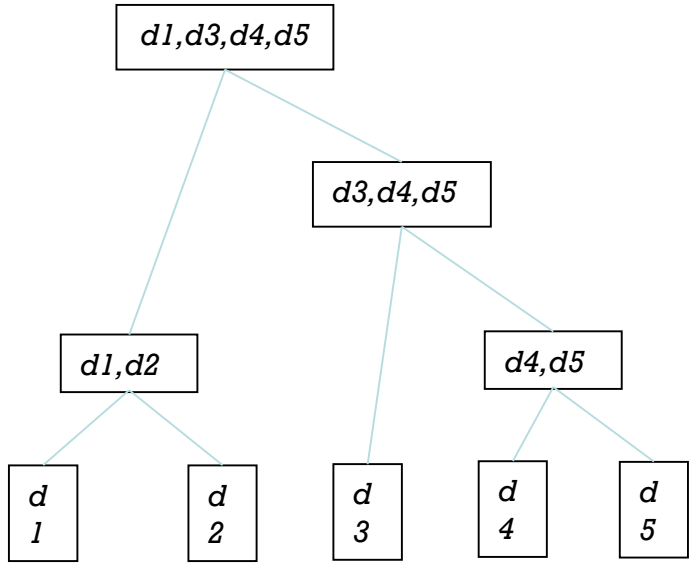
- Many applications require hierarchical clustering of data
  - Clustering web documents into topical hierarchy (Yahoo!, Dmoz directories)
- The hierarchy obtained during the clustering process is called *"Dendogram"*
- A specific clustering is obtained by cutting-off the dendogram at desired level
  - The connected components form the clusters
- No need to know the number of clusters a-priori



Hierarchical Agglomerative Clustering (HAC)





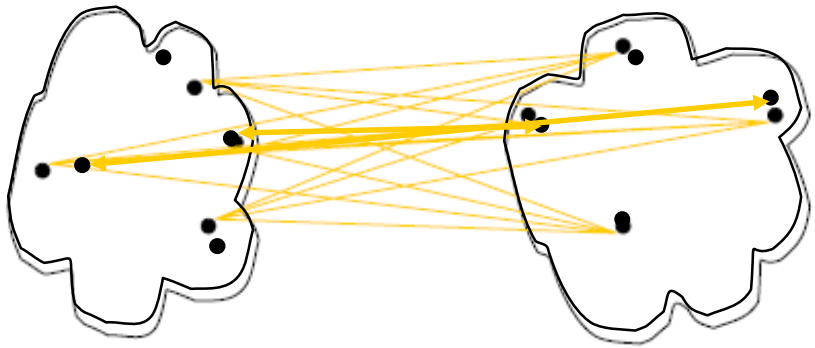


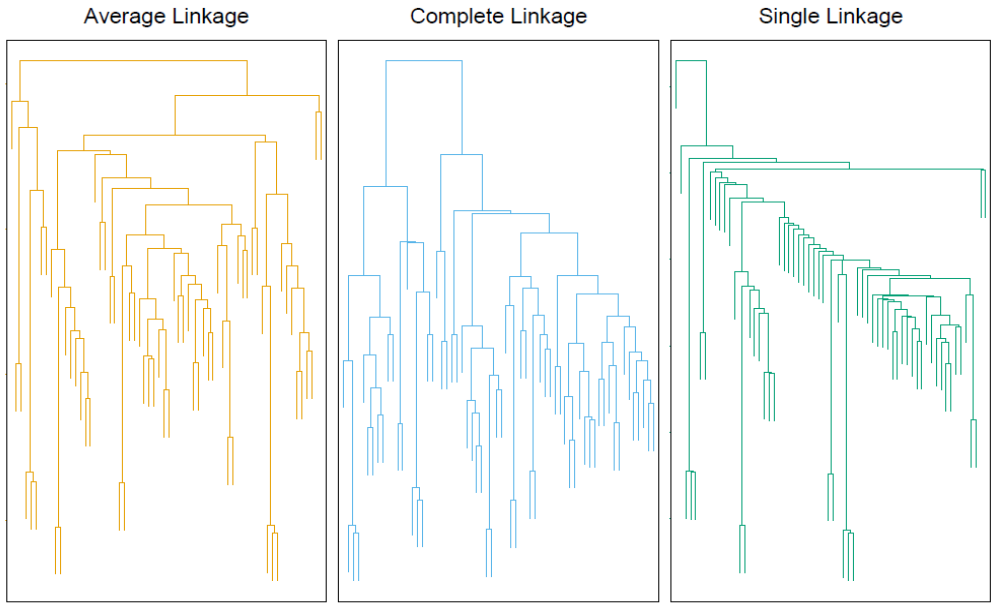
Key Point

How to define inter-cluster similarities?

Inter-Cluster Distance Functions

- Single-linkage (MIN)
  - ***Minimum distance*** between any two points across clusters
- Complete-linkage (MAX)
  - ***Maximum distance*** between any two points across clusters
- Average-linkage (AVG)
  - ***Average distance*** between the points across clusters



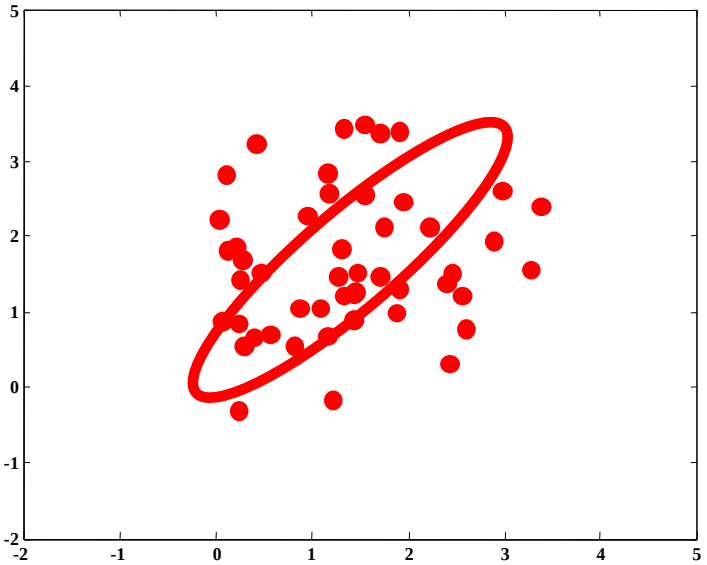


Soft Clustering

- In real life, data points may belong to more than one cluster
- Document clustering
  - A document about “Politics in BCCI” belongs to both “politics” and “sports”
- Handwriting Recognition
  - Given N X N image scans of digits from 0-9, group them as per the digit
  - We would like to quantify the confusion between digits
  - Pr(Image=8)=0.58, Pr(Image=3)=0.42 – We know the confusion was between 8 and 4
- Clustering citizens based on their salary
  - Clearly, different sub-populations (clusters) have different average incomes
  - Confidence assigned to a point depends on the cluster (sub-population) parameters

Multivariate Gaussian

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$



Estimation of Mean and Co-Variance

$$\hat{\mu} = \frac{1}{N} \sum_i x^{(i)}$$

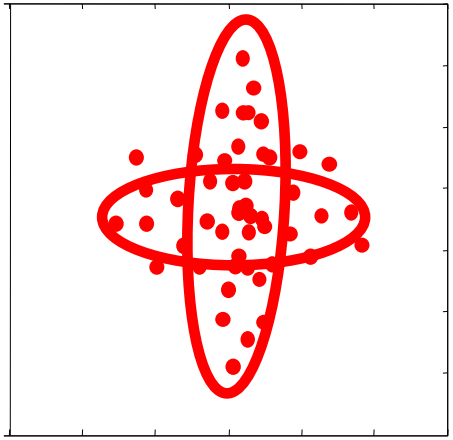
$$\hat{\Sigma} = \frac{1}{N} \sum_i (x^{(i)} - \hat{\mu})^T (x^{(i)} - \hat{\mu})$$

We will model each cluster using a Multi-variate Gaussian

CSE 7306c

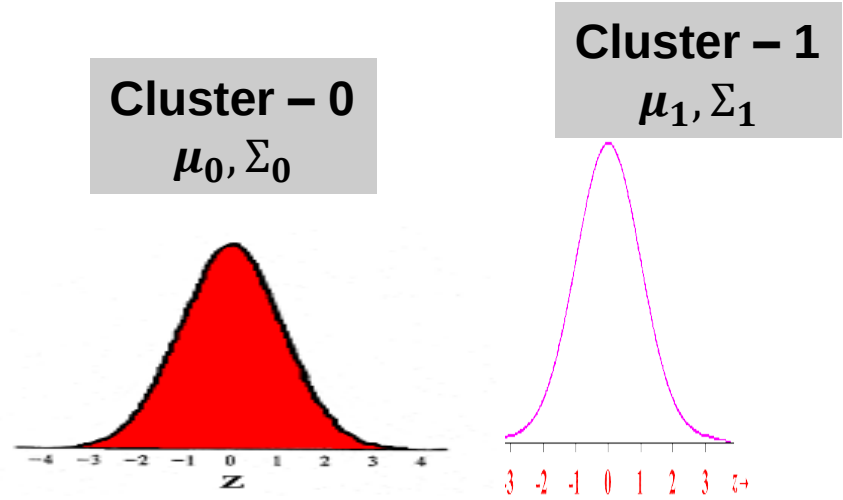
Example - Mixture of Gaussians

- Clusters modeled as Gaussians
  - Not just by their means
- The mean and co-variance of clusters are unknown
  - Isn't that dependent on the cluster assignments?
  - The cluster assignments in turn depend on the cluster distribution parameters
  - Chicken and Egg problem ☺
- Expectation Maximization (EM)
  - Used to learn both the cluster assignments and the parameters



Expectation Maximization (EM)

- Assume two Gaussian clusters
- Each r_i can take two values {0,1}
  - Denotes which cluster it belongs to
  - Hidden variable – needs to be estimated
- Prior probability of cluster-1: π
- If we know r_is, we can estimate Gaussian parameters of each cluster
  - Estimate $\boldsymbol{\mu_0}, \Sigma_0$ with all x_is where r_i=0
  - Estimate $\boldsymbol{\mu_1}, \Sigma_1$ with all x_is where r_i=1



Hidden Variable	r_2	r_3	r_i
x_1	x_2	x_3	x_i

Observed Data

CSE 7306c

EM Objective

Observed Data $x = (x_1, x_2, \ldots, x_N)$ Continuous I.I.D

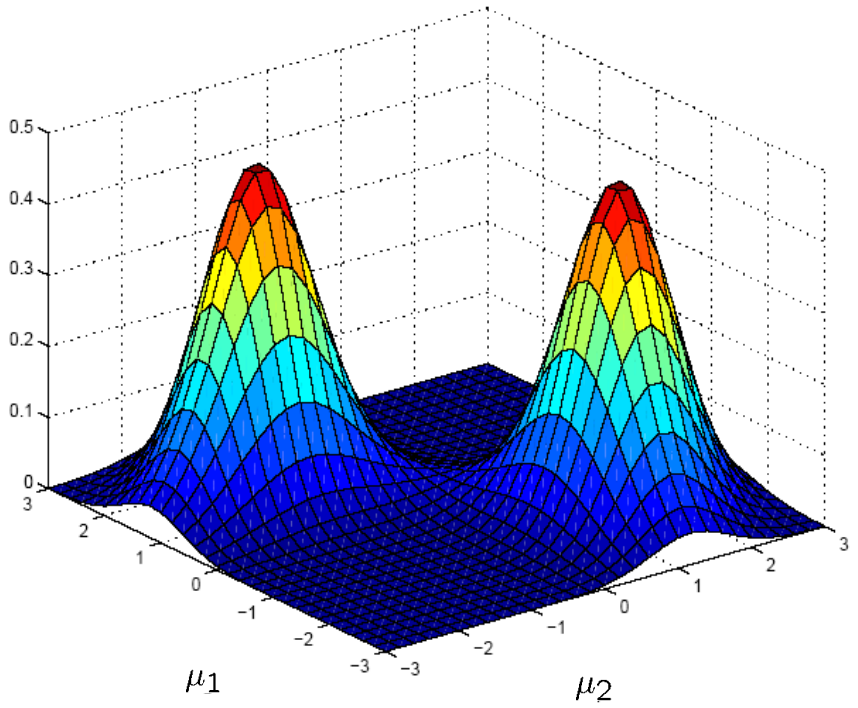
Latent variables $z = (z_1, z_2, \ldots, z_N)$ Discrete 1 ... C

- Maximize the likelihood l of the observed data

$$l(\theta; x) = log\, p(x|\theta) = log\, \prod_x p(x|\theta)$$

$$= \sum_x log\, \sum_z p(x, z|\theta)$$

$$l_c(\theta; x, z) \triangleq \sum_x \log p(x, z \mid \theta)$$



EM Algorithm (Contd..)

- Start with parameters describing each cluster
- $\boldsymbol{\mu_1}, \Sigma_1, \boldsymbol{\mu_2}, \Sigma_2, \ \pi$

- **Expectation Step (E-Step)**
  - For each x_i, compute expected value of r_i

$$z_i = \mathrm{E}(r_i) = \Pr(r_i = 1 | x = x_i) = \frac{\Pr(x = x_i | r_i = 1) \times \Pr(r_i = 1)}{\Pr(x = x_i)}$$

$$z_i = \mathrm{E}(r_i) = \frac{\pi * N(x_i; \boldsymbol{\mu_1}, \Sigma_1)}{(1 - \pi) * N(x_i; \boldsymbol{\mu_0}, \Sigma_0) + \pi * N(x_i; \boldsymbol{\mu_1}, \Sigma_1)}$$

  - If x_i is very likely under the 1st Gaussian, z_i gets high weight

EM Algorithm (Contd..)

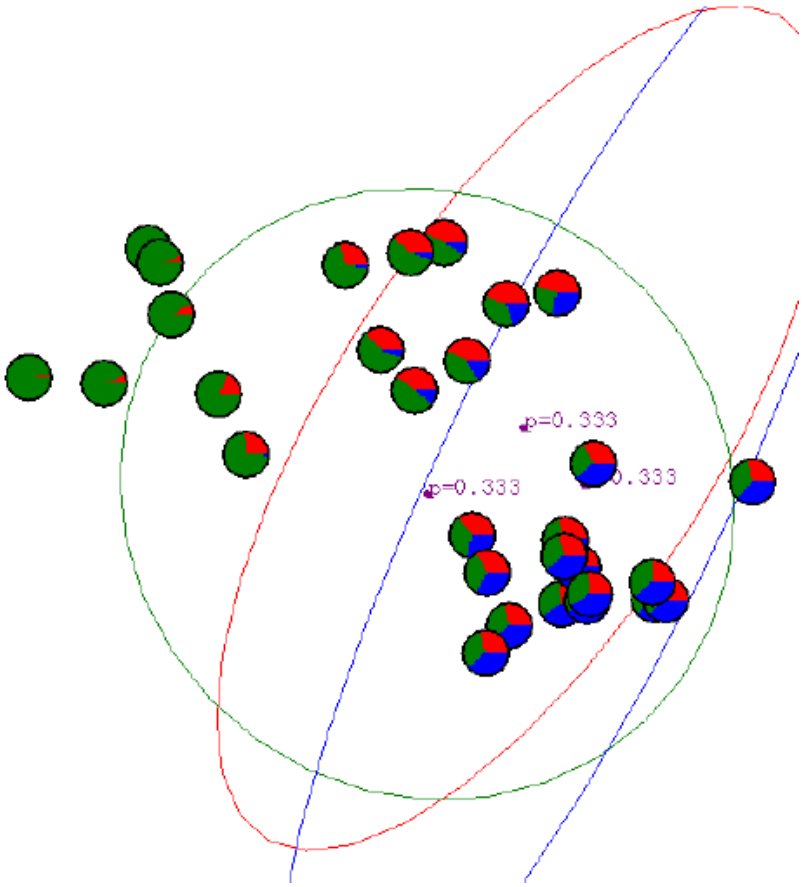
- Maximization Step (M-Step)
  - Based on expected values of hidden variables, update the parameters of Gaussian

$$\mu_1 = \frac{\sum_i z_i \times x_i}{\sum_i z_i}$$

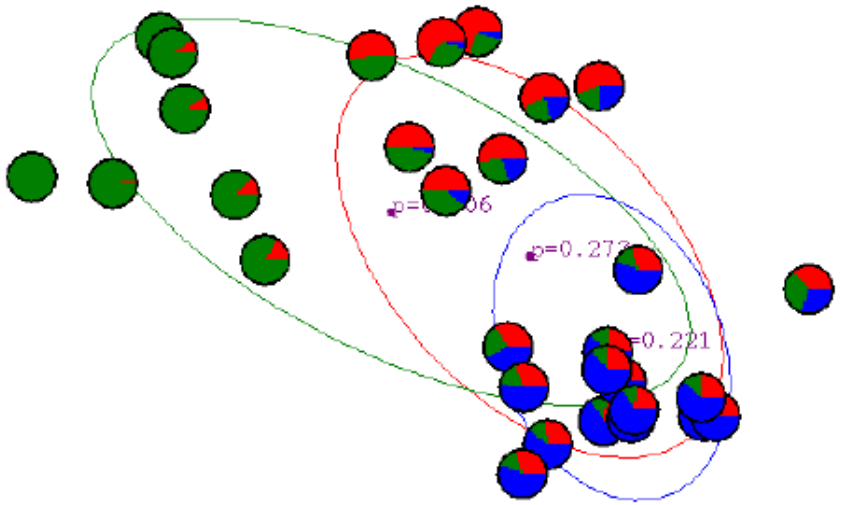
$$\Sigma_1 = \frac{1}{\sum_i z_i} \sum_i z_i \times (x_i - \mu_1)^T (x_i - \mu_1)$$

- Iterate until log-likelihood l doesn't decrease further or maximum number of iterations achieved

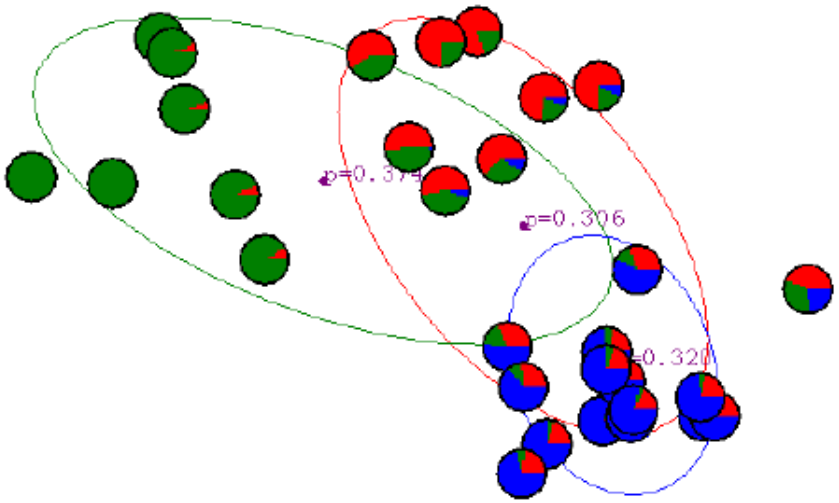
Gaussian Mixture with 3 Clusters



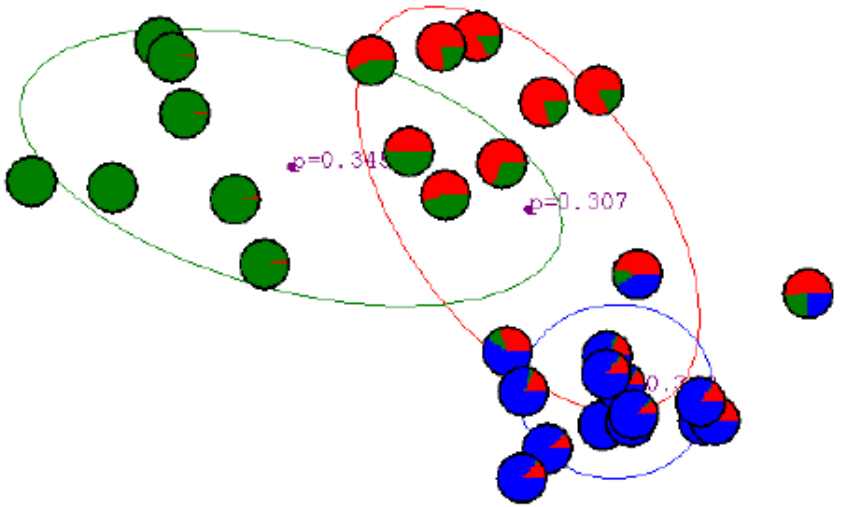
After first iteration



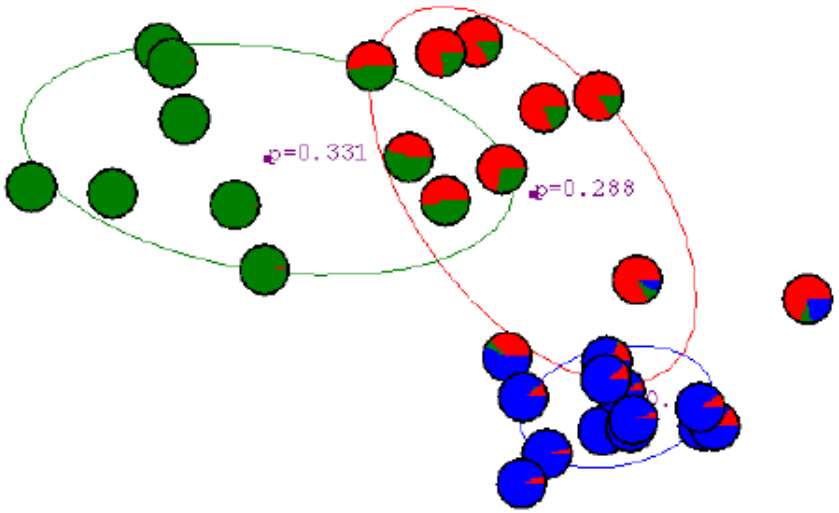
After 2nd iteration



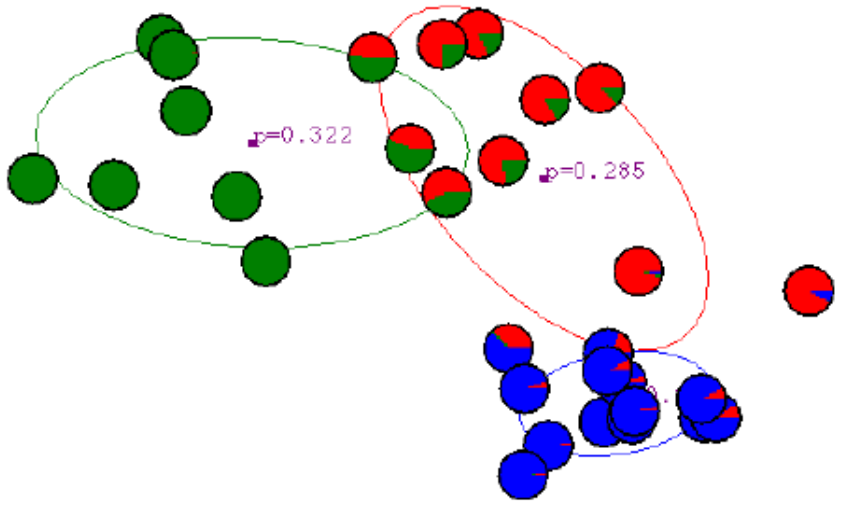
After 3rd iteration



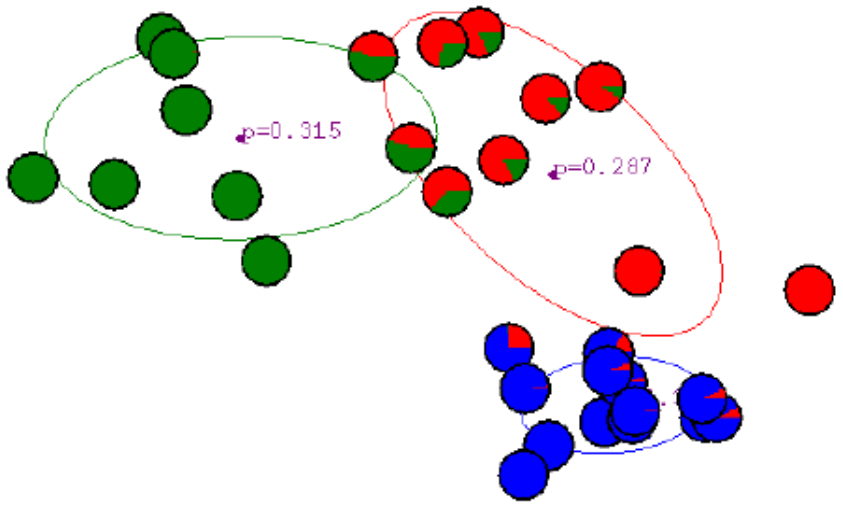
After 4th iteration



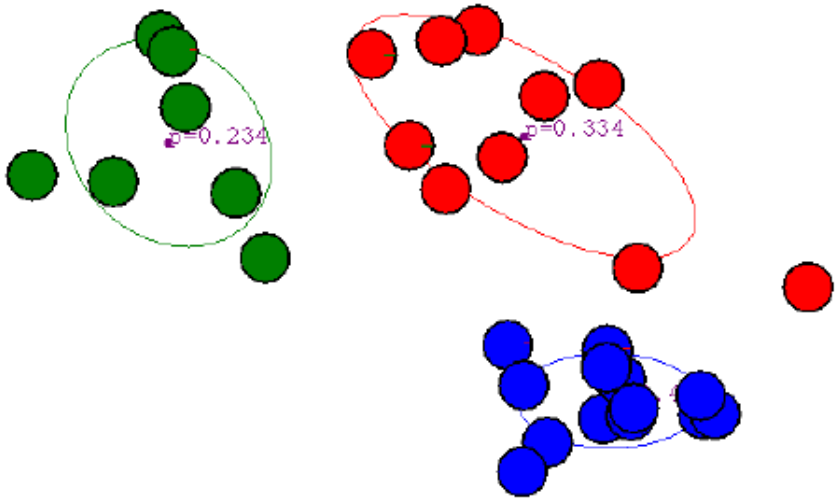
After 5th iteration



After 6th iteration

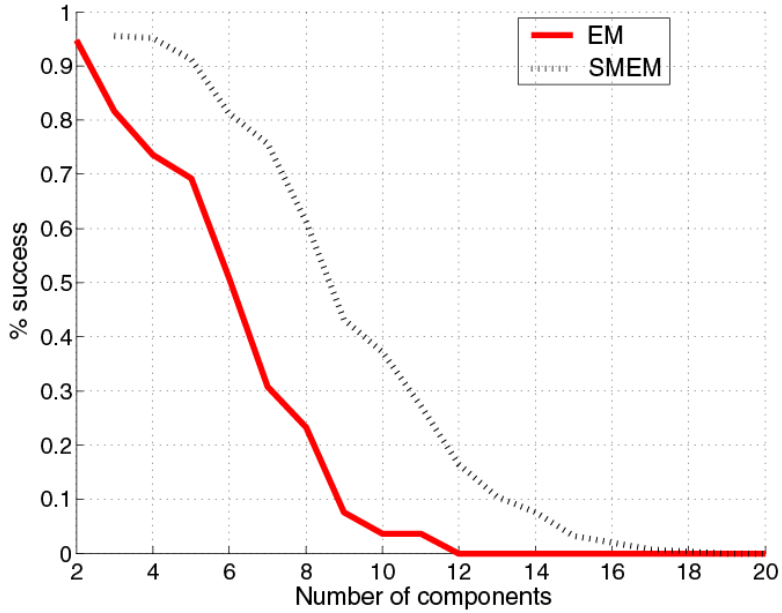


After 20th iteration



EM – Practical Issues

- **Initialization**
  - Mean of data + random offset
  - K-Means
- **Termination**
  - Max # iterations
  - Log-likelihood change
  - Parameter change
- **Convergence**
  - Local maxima
- **Robustness**
  - Not very robust when the number of components are high



Spectral Graph Theory

- Possible approach
  - Represent a similarity graph as a matrix
  - Apply knowledge from Linear Algebra…

- The *eigenvalues* and *eigenvectors* of a matrix provide global information about its structure.

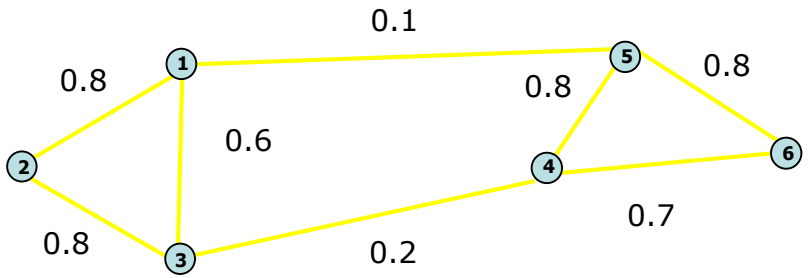
$$\begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- *Spectral Graph Theory*
  - Analyse the "spectrum" of matrix representing a graph.
  - *Spectrum* : The eigenvectors of a graph, ordered by the magnitude(strength) of their corresponding eigenvalues.

$$\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$$

Matrix Representations

- **Adjacency Matrix (A)**
  - $n \times n$ matrix
  - $A = [w_{ij}]$: edge weight between vertex x_i and x_j



	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	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

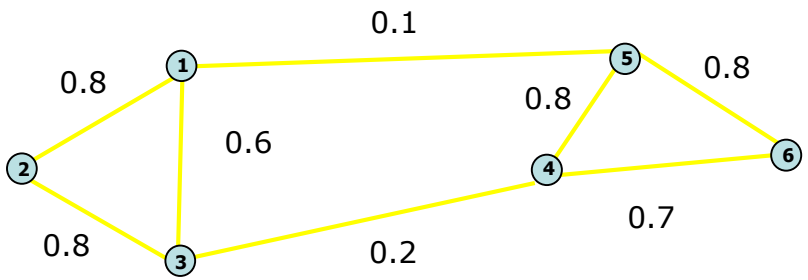
- Important properties:
  - Symmetric matrix

  ⇒ Eigenvalues are real

  ⇒Eigenvector could span orthogonal base

Matrix Representations (continued)

- **Degree matrix (D)**
  - $n \times n$ diagonal matrix
  - $D(i,i) = \sum_j w_{ij}$: total weight of edges incident to vertex x_i

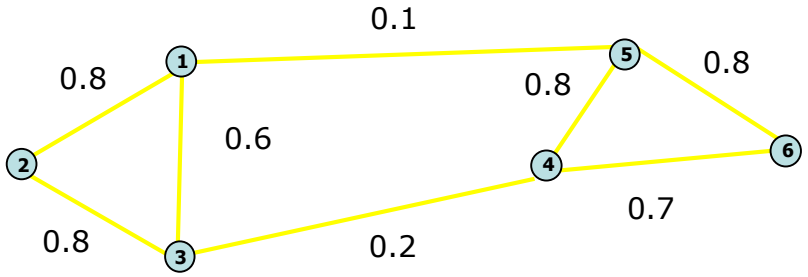


	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	0	0	0	0	0
x_2	0	1.6	0	0	0	0
x_3	0	0	1.6	0	0	0
x_4	0	0	0	1.7	0	0
x_5	0	0	0	0	1.7	0
x_6	0	0	0	0	0	1.5

- Important application:
  - Normalise adjacency matrix

Matrix Representations (continued)

- **Laplacian matrix** (L)
  - *n x n* symmetric matrix



$$L = D - A$$

	x_1	x_2	x_3	x_4	x_5	x_6
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	0	-0.2	1.7	-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

- Important properties:
  - Eigenvalues are non-negative real numbers
  - Eigenvectors are real and orthogonal
  - Eigenvalues and eigenvectors provide an insight into the connectivity of the graph…

Find An Optimal Min-Cut (Hall'70, Fiedler'73)

- Express a bi-partition (A,B) as a vector

$$p_i = \begin{cases} +1 & \text{if } x_i \in A \\ -1 & \text{if } x_i \in B \end{cases} \quad = p^T L\, p$$

Laplacian matrix

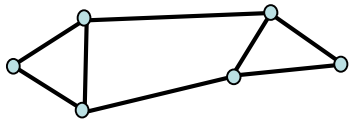
- The laplacian is semi positive
- The *Rayleigh Theorem* shows:
  - The minimum value for $f(p)$ is given by the 2nd smallest eigenvalue of the Laplacian L.
  - The optimal solution for p is given by the corresponding eigenvector λ_2, referred as the *Fiedler Vector*.

We can minimise the cut of the partition by finding a non-trivial vector p that minimises the function

$$f(p) = \sum_{i,j \in V} w_{ij}(p_i - p_j)^2$$

Spectral Bi-Partitioning Algorithm

1. **Pre-processing**
   - Build Laplacian matrix L of the graph



	x_1	x_2	x_3	x_4	x_5	x_6
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	0	-0.2	1.7	-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

2. **Decomposition**
   - Find eigenvalues X and eigenvectors Λ of the matrix L
   - Map vertices to corresponding components of λ_2

$\Lambda =$

0.0
0.4
2.2
2.3
2.5
3.0

$X =$

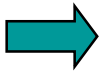
0.4	0.2	0.1	0.4	-0.2	-0.9
0.4	0.2	0.1	-0.	0.4	0.3
0.4	0.2	-0.2	0.0	-0.2	0.6
0.4	-0.4	0.9	0.2	-0.4	-0.6
0.4	-0.7	-0.4	-0.8	-0.6	-0.2
0.4	-0.7	-0.2	0.5	0.8	0.9

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

Spectral Bi-Partitioning Algorithm (Contd..)

- Grouping
  - Sort components of reduced 1-dimensional vector.
  - Identify clusters by splitting the sorted vector in two.
- How to choose a splitting point?
  - Naïve approaches:
    - Split at 0, mean or median value
  - More expensive approaches
    - Attempt to minimise normalised cut criterion in 1-dimension

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



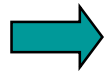
Split at 0

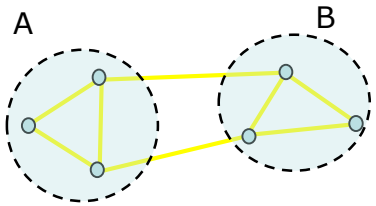
Cluster *A*: Positive points

Cluster *B*: Negative points

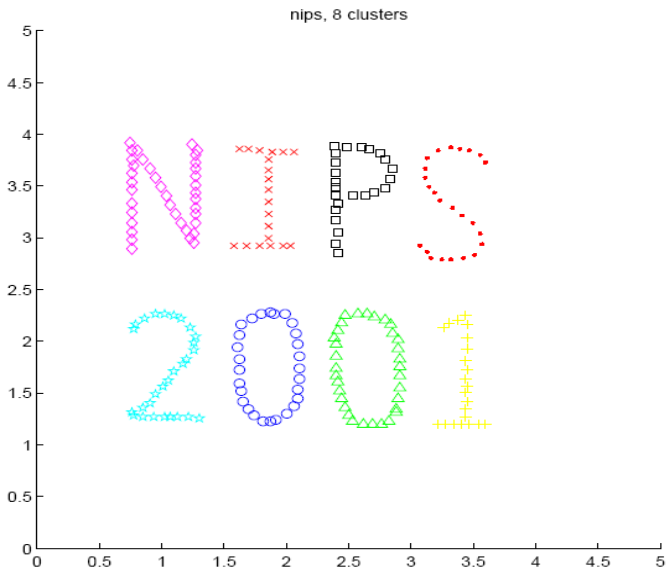
x_1	0.2
x_2	0.2
x_3	0.2

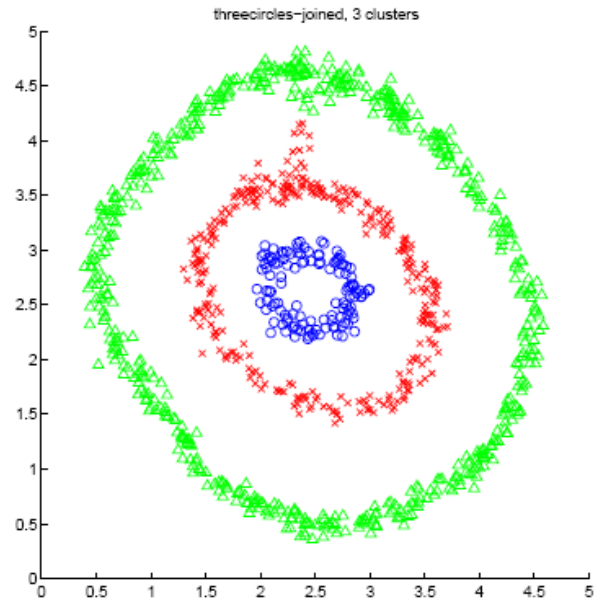
x_4	-0.4
x_5	-0.7
x_6	-0.7

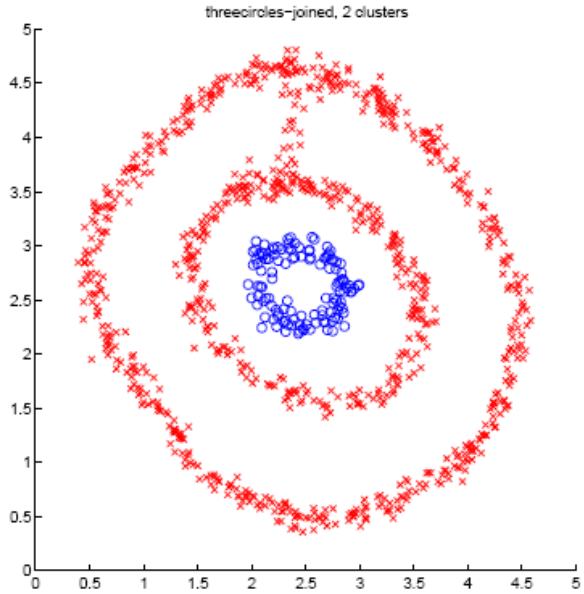


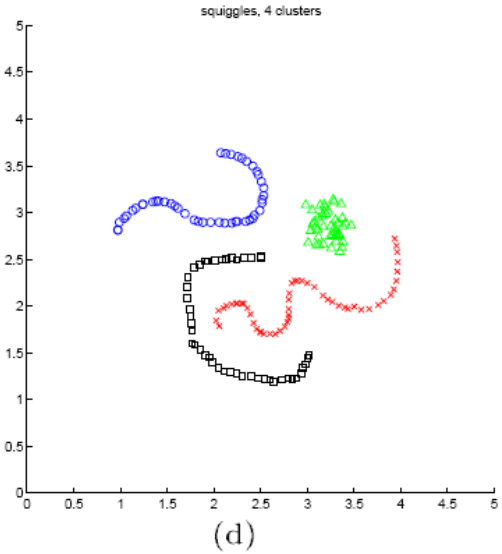


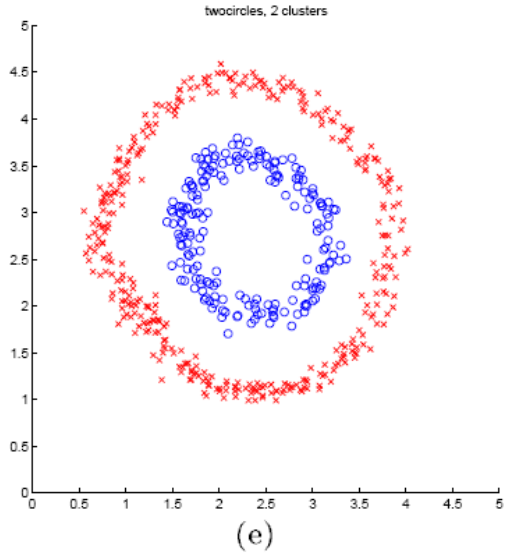
Sample Clusters Learnt











Summary

- Introduced the paradigm of “Unsupervised Learning”
  - The task of discovering intrinsic patterns from data without any supervision
- Depending on the specific objective to be optimized and assumptions made about data, there are many clustering algorithms proposed in literature
- Some clustering algorithms we discussed today
  - K-Means
  - Agglomerative Clustering
  - Expectation Maximization (EM)
- Practical issues while using the above algorithms
- We studied the notion of cluster evaluation
- We also discussed Spectral Clustering which allows us to learn non-spherical and arbitrary forms of clusters

References/Resources

- Classic and Modern Data Clustering

http://learning.stat.purdue.edu/mlss/_media/mlss/meila.pdf

- AutoLab Tutorial on Gaussian Mixture Models

http://www.autonlab.org/tutorials/gmm14.pdf

- Andrew Ng's Lecture on CourseEra

https://class.coursera.org/ml-003/lecture/preview

- The Elements of Statistical Learning – Data Mining, Inference, and Prediction

http://www-stat.stanford.edu/~tibs/ElemStatLearn/download.html

Inspire...Educate...Transform.

International School of Engineering

Plot 63/A, 1st Floor, Road # 13, Film Nagar, Jubilee Hills, Hyderabad - 500 033

For Individuals: +91-9502334561/63 or 040-65743991

For Corporates: +91-9618483483

Web: http://www.insofe.edu.in

Facebook: https://www.facebook.com/insofe

Twitter: https://twitter.com/Insofeedu

YouTube: http://www.youtube.com/InsofeVideos

SlideShare: http://www.slideshare.net/INSOFE

LinkedIn: http://www.linkedin.com/company/international-school-of-engineering