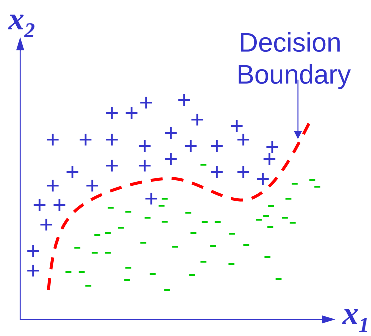
EE 6885 Statistical Pattern Recognition

Fall 2005
Prof. Shih-Fu Chang
http://www.ee.columbia.edu/~sfchang

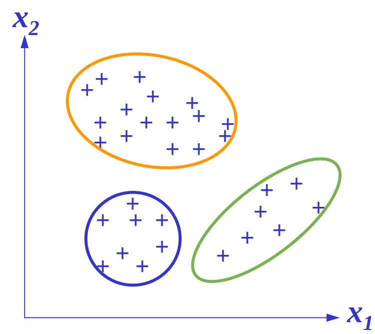
Lecture 20 (12/5/05)

- Topics
  - Clustering
    - GMM and k-means, DHS Chap. 10.2-10.4
    - Criterion Function Maximization, DHS Chap. 10.7
    - Hierarchical Clustering, DHS Chap. 10.9
  - Next lecture: graph-based clustering
- Homework #8, Due Dec. 12^{th} Monday
- Review
  - Dec. 12^{th} Monday
- Final Exam
  - Dec. 16^{th} Friday 1:10-3 pm, Mudd Rm 644

Classification vs. Clustering



- Data with labels
- Supervised
- Find decision boundaries



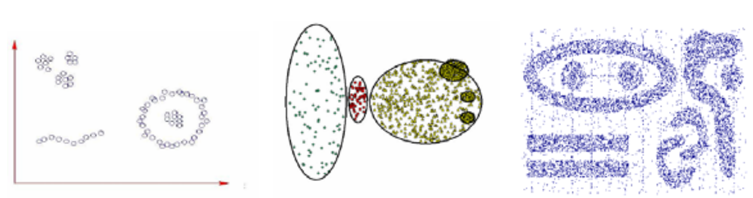
- Data without labels
- Unsupervised
- Find data structures and clusters

EE6887-Chang

20-3

- **Applications**
  - Document Clustering: Text topic discovery
  - Image segmentation: Object vs. background, face vs. non-face
  - Network traffic pattern mining

- **Some challenging cases**



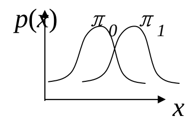
(from A. Jain)

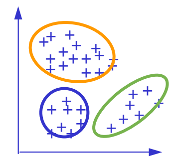
- Non-spherical clusters
- Clusters with different densities

EE6887-Chang

20-4

Review: Mixture Of Gaussians





- Model data distributions as GMM

$$p(x) = \sum_z p(z)p(x \mid z)$$
$$= \sum_z \pi_z N\left(x \middle| \mu_z, \Sigma_z\right) \quad = \sum_{z=1}^{Z} \pi_z \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_z|}} e^{-\frac{1}{2}(x-\mu_z)^T \Sigma_z^{-1}(x-\mu_z)}$$

- Given data $x_1, \ldots, x_N$, log-likelihood:

$$l = \sum_{n=1}^{N} \log\left(\pi_0 N(x_n|\mu_0, \Sigma_0) + \pi_1 N(x_n|\mu_1, \Sigma_1)\right)$$

- Posterior probability of x being generated by a cluster *i*

$$posteriers = \tau^i = p\left(z = i \middle| x, \theta\right) \qquad parameter: \quad \theta = \{\mu_0, \Sigma_0, \mu_1, \Sigma_1\}$$

- Optimization

find $\{\mu_0, \Sigma_0, \mu_1, \Sigma_1\}$ and mixture priors π_z to max. likelihood

Expectation-Maximization (E-M) Solution of GMM

$$Q(\theta \mid \theta_t) = \sum_{n=1}^{N} \sum_z \underbrace{p(z \mid x_n, \theta_t)} \; \underbrace{\log p(x_n, z \mid \theta)} + const$$

- EM for estimating θ and τ_i.

$$Expectation: \; \tau_n^{i(t)} = \frac{\pi_i^{(t)} N\left(x_n \mid \mu_i^{(t)}, \Sigma_i^{(t)}\right)}{\sum_j \pi_i^{(t)} N\left(x_n \mid \mu_j^{(t)}, \Sigma_j^{(t)}\right)}$$

Weight from component *i*

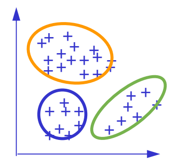
$$Maximation: \quad \mu_i^{(t+1)} = \frac{\sum_n \tau_n^{i(t)} x_n}{\sum_n \tau_n^{i(t)}} \qquad \Sigma_i^{(t+1)} = \frac{\sum_n \tau_n^{i(t)} \left(x_n - \mu_i^{(t)}\right)\left(x_n - \mu_i^{(t)}\right)^T}{\sum_n \tau_n^{i(t)}}$$

Divide data to each group,
Compute mean and variance
from each group

$$\pi_i^{(t+1)} = \frac{\sum_n \tau_n^{i(t)}}{N}$$

GMM for Clustering

- Given the estimated GMM model, compute the probability that *x* is generated by cluster *i*



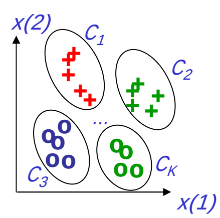
$$posteriers = \tau^i = p\left(z = i \middle| x, \theta\right), \quad \theta = \{\mu_0, \Sigma_0, \mu_1, \Sigma_1, \pi_0\}$$

$$Expectation: \tau_n^{i(t)} = \frac{\pi_i^{(t)} N\left(x_n \mid \mu_i^{(t)}, \Sigma_i^{(t)}\right)}{\sum_j \pi_i^{(t)} N\left(x_n \mid \mu_j^{(t)}, \Sigma_j^{(t)}\right)}$$

- Each sample is assigned to every cluster with a 'soft' decision.

Comparison: K-Mean Clustering

- K-mean clustering
  - Fix K values
  - Choose initial representative of each cluster
  - Map each sample to its closest cluster

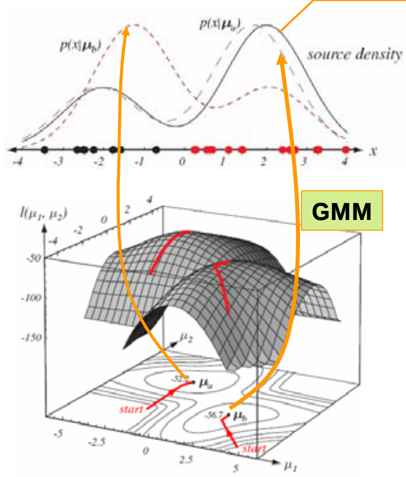


$$for\ i=1,2,...,N,$$
$$x_i \to C_k, if\ \ Dist(x_i, C_k) < Dist(x_i, C_{k'}), k \neq k'$$
$$end$$

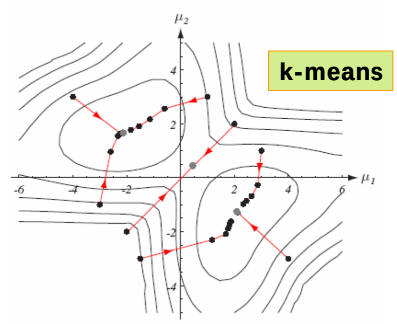
Hard decision

  - Re-compute the centers
- Can be used to initialize the EM for GMM

- **Clustering Example**



$$p(x) = \frac{1}{3}N(-2,1) + \frac{2}{3}N(2,1)$$



- **Soft memberships**
- **Multiple local maximums with different "scores"**

- **Symmetrical wrt diagonal line**
- **Multiple local maximums**
- **May converge to trivial solutions**
- **Hard memberships**

Fuzzy K-Means Clustering

- **incorporate soft membership into k-means clustering**
- **optimize a global cost function in each iteration**

$$J_{fuz} = \sum_{i=1}^{c}\sum_{j=1}^{n}\left[\hat{p}(\omega_i \mid \mathbf{x}_j, \hat{\boldsymbol{\theta}})\right]^b \left\|\mathbf{x}_j - \boldsymbol{\mu}_i\right\|^2$$

- **The optimal solution is:**

E-step

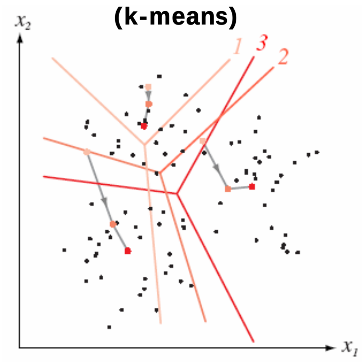
In each iteration, compute $\hat{p}(\omega_i \mid \mathbf{x}_j) = \dfrac{\left(1/d_{ij}\right)^{1/(b-1)}}{\sum_{r=1}^{c}\left(1/d_{rj}\right)^{1/(b-1)}}$ where $d_{ij} = \left\|\mathbf{x}_j - \boldsymbol{\mu}_i\right\|^2$

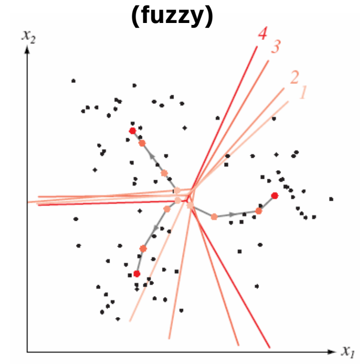
$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n}\left[\hat{p}(\omega_i \mid \mathbf{x}_j)\right]^b \mathbf{x}_j}{\sum_{j=1}^{n}\left[\hat{p}(\omega_i \mid \mathbf{x}_j)\right]^b}$$

M-step

- **What if b=1?**

K-means vs. fuzzy k-means





- **Each point mapped to multiple clusters**
- **The initial centers are close to each other**

Variations of Criterion Functions

- **Sum of squared errors**

$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i}^{n} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

D_i : subset of samples belonging to cluster i
(hard assignment)

$$J_e = \frac{1}{2} \sum_{i=1}^{c} n_i \bar{s}_i$$

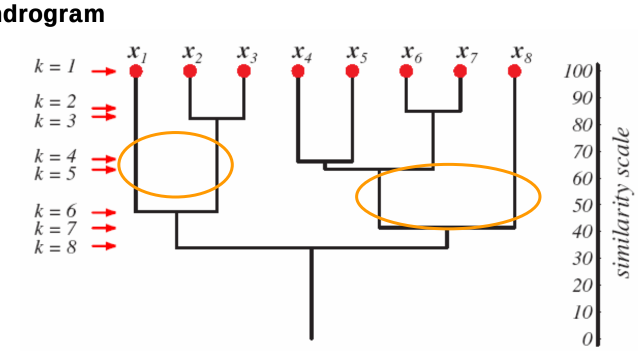
$$\bar{s} = \frac{1}{n_i^2} \sum_{\mathbf{x} \in D_i} \sum_{\mathbf{x}' \in D_i} \left\| \mathbf{x} - \mathbf{x}' \right\|^2$$

Avg. squared distance between points in cluster *i*

- **Sum of squared error criterion tends to favor equal sized clusters**
- **The new formulation allows replacing squared distance with other measures such as average, min., or max. distance etc.**

Hierarchical Clustering

- **Add hierarchical structures to clusters**
  - **many real-world problems have such hierarchical structures**
  - **e.g., biological, semantic taxonomy**
- **Agglomerative vs. Divisive**
- **Dendrogram**

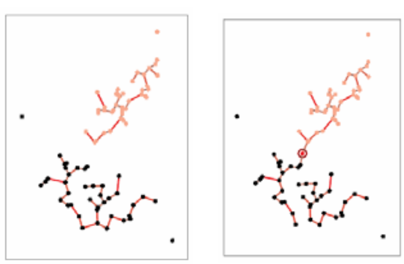


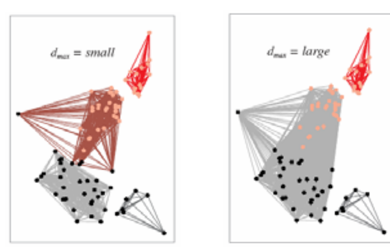
- **Use large gap of similarity to find a suitable number of clusters → clustering validity**

distances or similarity for merging

$$d_{\min}(D_i, D_j) = \min_{\mathbf{x}\in D_i, \mathbf{x}'\in D_j} \|\mathbf{x}-\mathbf{x}'\|$$

$$d_{\max}(D_i, D_j) = \max_{\mathbf{x}\in D_i, \mathbf{x}'\in D_j} \|\mathbf{x}-\mathbf{x}'\|$$





- **Nearest neighbor algorithm, minimal algorithm**
- **Merging results in the min. distance spanning tree**
- **But sensitive to noise/outlier**

- **Farthest neighbor algorithm, maximum algorithm**
- **Use distance threshold to avoid large-diameter clusters**
- **Discourage forming elongated clusters**

Other distance metrics

$$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x} \in D_i} \sum_{\mathbf{x}' \in D_j} \left\| \mathbf{x} - \mathbf{x}' \right\| \qquad d_{mean}(D_i, D_j) = \left\| \mu_i - \mu_j \right\|$$