CS 2750 Machine Learning
Lecture 17

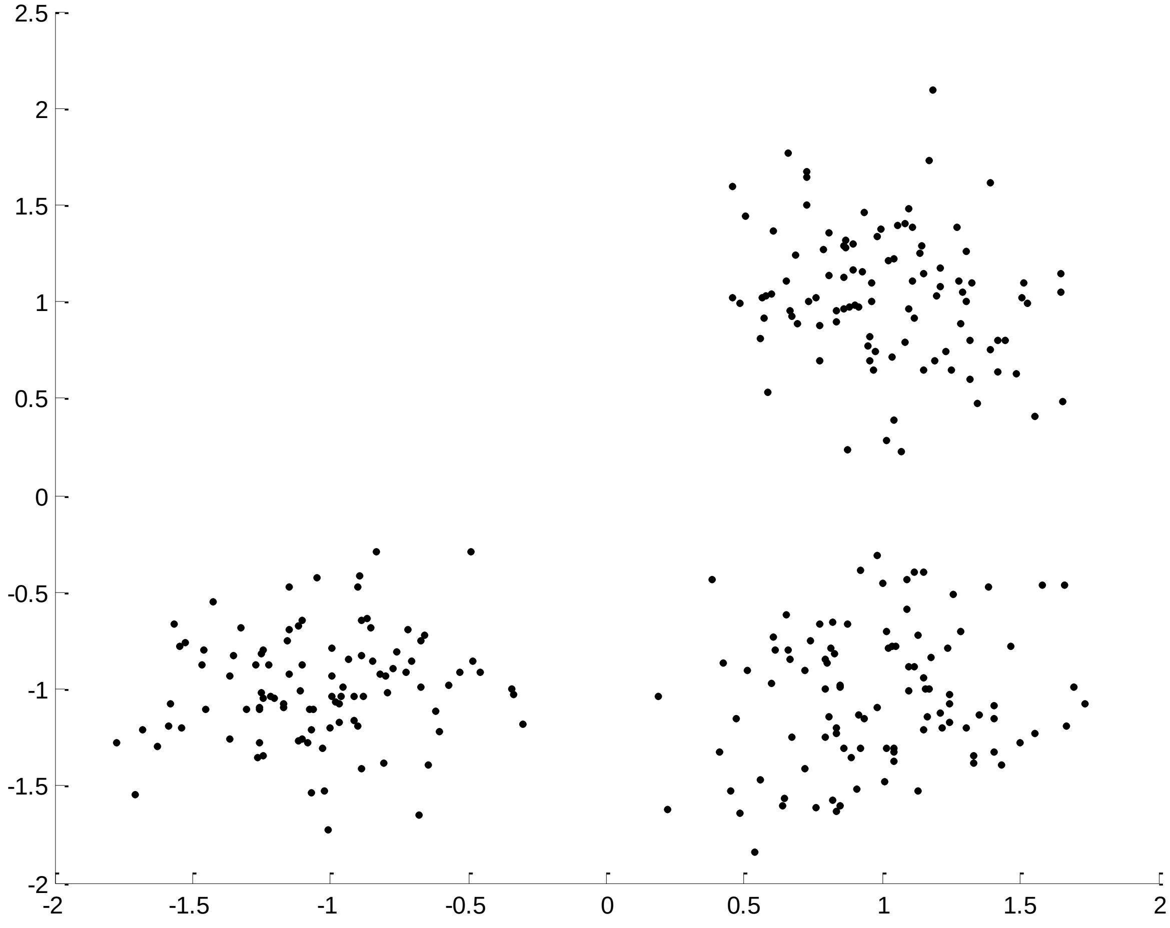
Clustering I.

Milos Hauskrecht

milos@cs.pitt.edu

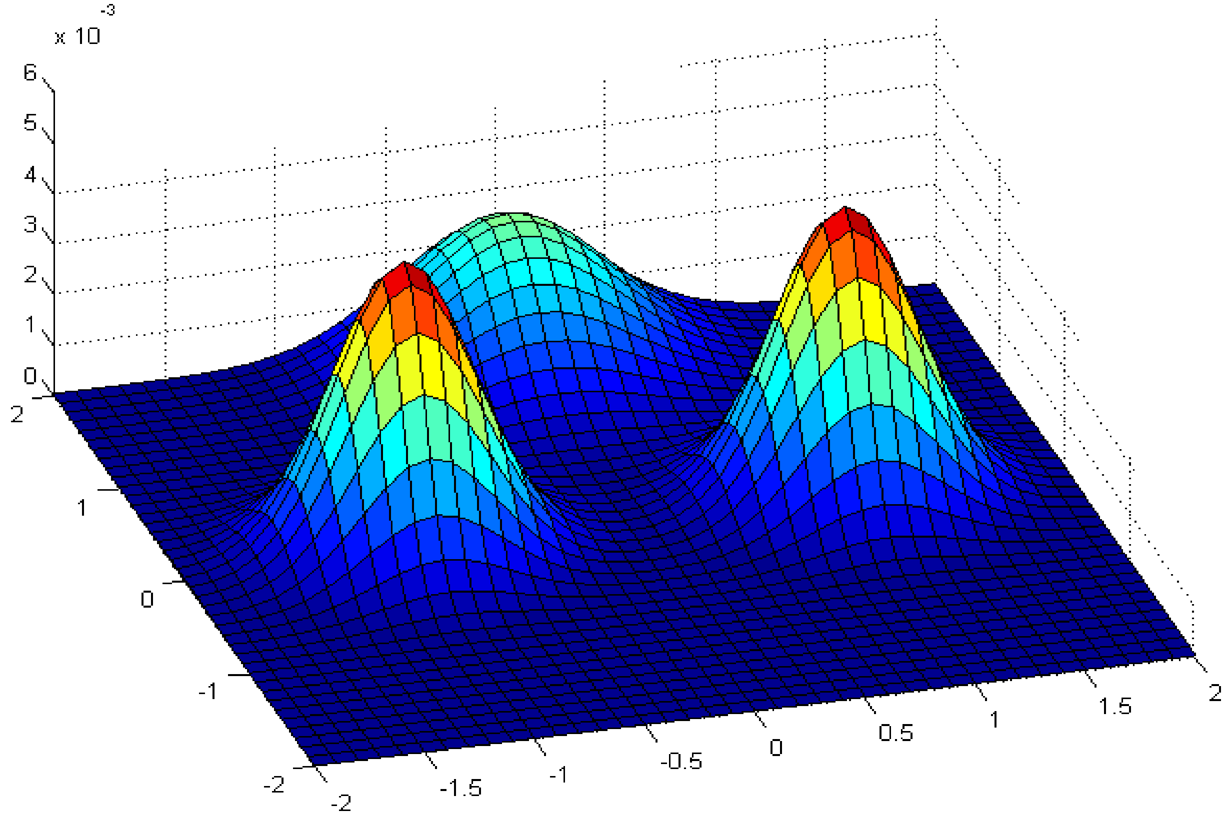
5329 Sennott Square

Gaussian mixture model



Mixture of Gaussians

- Density function for the Mixture of Gaussians model

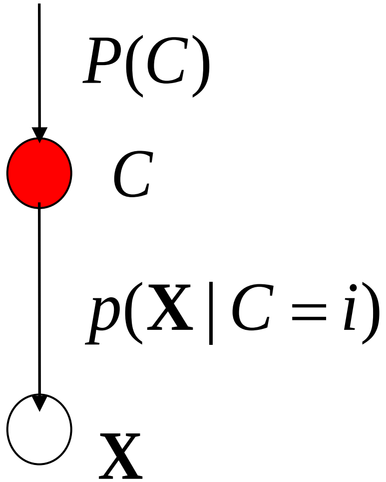


Gaussian mixture model

Probability of occurrence of a data example ***x***

is modeled as

$$p(\mathbf{x}) = \sum_{i=1}^{m} p(C=i)\, p(\mathbf{x} \mid C=i)$$



where

$$p(C=i)$$

= probability of a data point coming from class $C=i$

$$p(\mathbf{x} \mid C=i) \approx N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

= class conditional density (modeled as a Gaussian) for class i

Remember: ***C*** **is hidden !!!!**

Generative classifier model

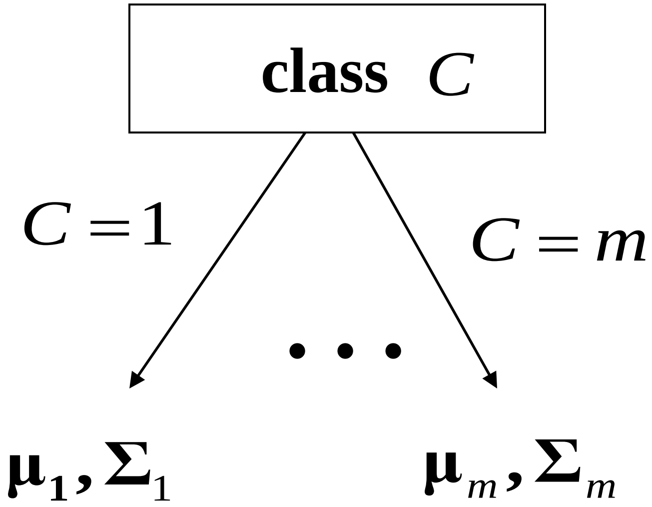
- Generative classifier model with Gaussian densities
- Assume the class labels are known. The ML estimate is

$$N_i = \sum_{j:C_l=i} 1$$

$$\tilde{\pi}_i = \frac{N_i}{N}$$

$$\tilde{\boldsymbol{\mu}}_i = \frac{1}{N_i} \sum_{j:C_l=i} \mathbf{x}_j$$

$$\tilde{\boldsymbol{\Sigma}}_i = \frac{1}{N_i} \sum_{j:C_l=i} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T$$



Gaussian mixture model

- In the Gaussian mixture Gaussians are not labeled
- We can apply **EM algorithm**:
  - re-estimation based on the class posterior

$$h_{il} = p(C_l = i \mid \mathbf{x}_l, \Theta') = \frac{p(C_l = i \mid \Theta') p(x_l \mid C_l = i, \Theta')}{\sum_{u=1}^{m} p(C_l = u \mid \Theta') p(x_l \mid C_l = u, \Theta')}$$

$$N_i = \sum_l h_{il}$$

Count replaced with the expected count

$$\tilde{\pi}_i = \frac{N_i}{N}$$

$$\tilde{\boldsymbol{\mu}}_i = \frac{1}{N_i} \sum_l h_{il} \mathbf{x}_j$$

$$\tilde{\boldsymbol{\Sigma}}_i = \frac{1}{N_i} \sum_l h_{il} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T$$

Gaussian mixture algorithm

- **Special case:** fixed covariance matrix for all hidden groups (classes) and a uniform prior on classes
- **Algorithm**:

Initialize means $\boldsymbol{\mu}_i$ for all classes i

Repeat two steps until no change in the means:

1. Compute the class posterior for each Gaussian and each point (a kind of responsibility for a Gaussian for a point)

Responsibility:

$$h_{il} = \frac{p(C_l = i \mid \Theta')\, p(x_l \mid C_l = i, \Theta')}{\sum_{u=1}^{m} p(C_l = u \mid \Theta')\, p(x_l \mid C_l = u, \Theta')}$$

2. Move the means of the Gaussians to the center of the data, weighted by the responsibilities

New mean:

$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{N} h_{il}\mathbf{x}_l}{\sum_{l=1}^{N} h_{il}}$$

K-means approximation to EM

Mixture of Gaussians with the fixed covariance matrix:

- posterior measures the responsibility of a Gaussian for every point

$$h_{il} = \frac{p(C_l = i \mid \Theta')\, p(x_l \mid C_l = i, \Theta')}{\sum_{u=1}^{m} p(C_l = u \mid \Theta')\, p(x_l \mid C_l = u, \Theta')}$$

- **Re-estimation of means:**

$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{N} h_{il} \mathbf{x}_l}{\sum_{l=1}^{N} h_{il}}$$

- **K- Means approximations**
- Only the closest Gaussian is made responsible for a point

$h_{il} = 1$ If i is the closest Gaussian

$h_{il} = 0$ Otherwise

- Results in moving the means of Gaussians to the center of the data points it covered in the previous step

K-means algorithm

K-Means algorithm:

Initialize k values of means (centers)

Repeat two steps until no change in the means:

- Partition the data according to the current means (using the similarity measure)
- Move the means to the center of the data in the current partition

- **Used frequently for clustering data**

Clustering

Groups together "similar" instances in the data sample

Basic clustering problem:

- distribute data into *k* different groups such that data points similar to each other are in the same group
- Similarity between data points is defined in terms of some distance metric (can be chosen)

Clustering is useful for:

- **Similarity/Dissimilarity analysis**

  Analyze what data points in the sample are close to each other
- **Dimensionality reduction**

  High dimensional data replaced with a group (cluster) label