# CS 2750 Machine Learning
# Lecture 16

## Expectation Maximization (EM).
## Mixtures of Gaussians.

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square



## Learning probability distribution

**Basic learning settings:**

- A set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$
- **A model of the distribution** over variables in ***X*** with parameters $\Theta$
- **Data** $D = \{D_1, D_2, .., D_N\}$

  **s.t.** $D_i = (x_1^i, x_2^i, \ldots x_n^i)$

**Objective:** find parameters $\hat{\Theta}$ that describe the data

**Assumptions considered so far:**

- Known structure and parameterizations
- Hidden variables
- Missing values

# Hidden variables

**Modeling assumption:**

Variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$

- We can add hidden variables – never observed in data

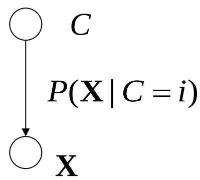
**Why to add hidden variables?**

- **More flexibility in describing the distribution** $P(\mathbf{X})$
- **Smaller parameterization of** $P(\mathbf{X})$
  - **New independences can be introduced via hidden variables**

**Example:**

- Latent variable models
  - hidden classes (categories)

Hidden class variable





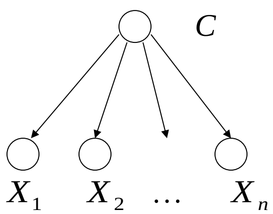

# Naïve Bayes with a hidden class variable

**Introduction of a hidden variable can reduce the number of parameters defining** $P(\mathbf{X})$

**Example:**

- Naïve Bayes model with a hidden class variable

**Hidden class variable**




Attributes are independent given the class

- **Useful in customer profiles**
  - Class value = type of customers

## Learning with hidden variables and missing values: EM

**Expectation maximization method**

**The key idea of the method:**

**Compute the parameter estimates** iteratively by performing the following two steps:

**Two steps of the EM:**

1. **Expectation step**. For all hidden and missing variables (and their possible value assignments) calculate their expectations for the current set of parameters $\mathbf{\Theta}'$
2. **Maximization step**. Compute the new estimates of $\mathbf{\Theta}$ by considering the expectations of the different value completions

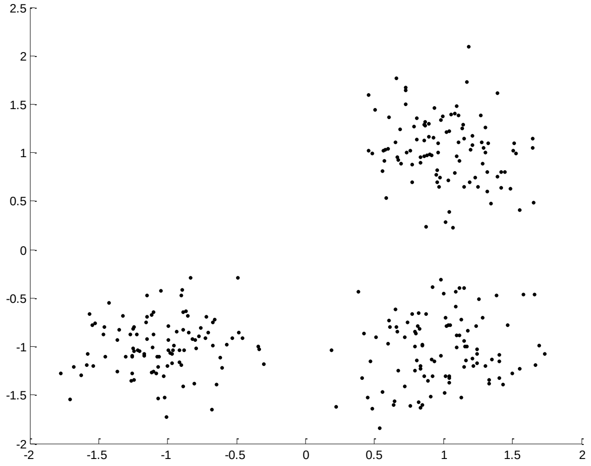
**Stop when no improvement possible**



## Gaussian mixture model

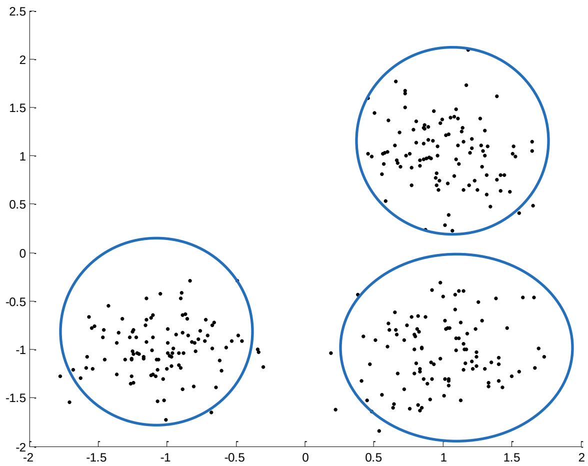
**Assume we have the following data**

**Question:** how to model its distribution?



# Gaussian mixture model

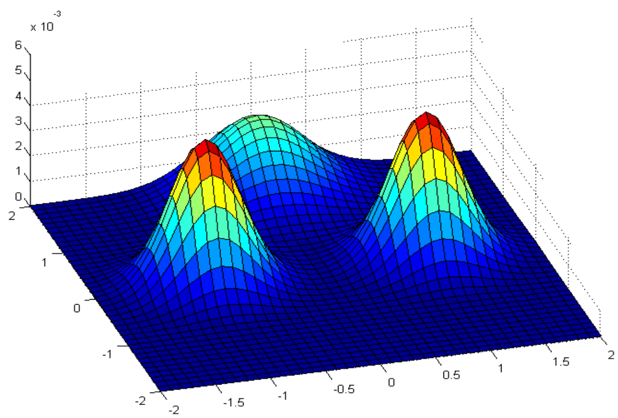
**Idea:** each group of data-points is covered by one Gaussian






# Mixture of Gaussians

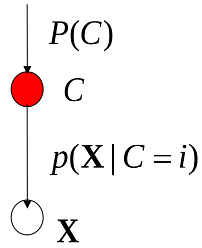
- Density function for the Mixture of Gaussians model



## Gaussian mixture model

Probability of occurrence of a data point $\mathbf{x}$ is modeled generatively as

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




where

$p(C=i)$

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

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

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

**<u>Special feature:</u> *C* is hidden !!!!**



## Generative classifier model

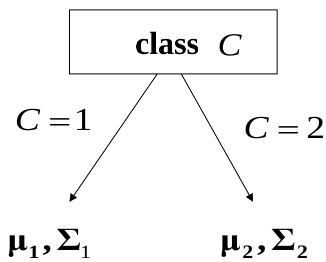
- **Generative classifier model (recall QDA or LDA)**
- Assume the class labels are known. The ML estimate is

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

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

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

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



## 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

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

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

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



## Gaussian mixture algorithm

- **A special case:**
  – a fixed covariance matrix for all hidden groups (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:**
$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{N} h_{il} \mathbf{x}_l}{\sum_{l=1}^{N} h_{il}}$$

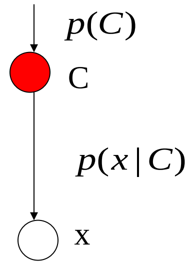
CS 2750 Machine Learning

## Gaussian mixture model. Gradient ascent.

- A set of parameters

$$\Theta = \{\pi_1, \pi_2, .. \pi_m, \mu_1, \mu_2, ... \mu_m\}$$

Assume unit variance terms and fixed priors




$$P(\mathbf{x} \mid C = i) = (2\pi)^{-1/2} \exp\left\{-\frac{1}{2}\|x - \mu_i\|^2\right\}$$

$$P(D \mid \Theta) = \prod_{l=1}^{N} \sum_{i=1}^{m} \pi_i (2\pi)^{-1/2} \exp\left\{-\frac{1}{2}\|x_l - \mu_i\|^2\right\}$$

$$l(\Theta) = \sum_{l=1}^{N} \log \sum_{i=1}^{m} \pi_i (2\pi)^{-1/2} \exp\left\{-\frac{1}{2}\|x_l - \mu_i\|^2\right\}$$

$$\frac{\partial l(\Theta)}{\partial \mu_i} = \sum_{l=1}^{N} h_{il}(x_l - \mu_i)$$

**- very easy on-line update**

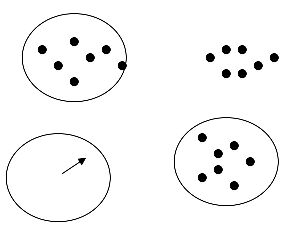


## EM versus gradient ascent

**Gradient ascent**

$$\mu_i \leftarrow \mu_i + \alpha \sum_{l=1}^{N} h_{il}(x_l - \mu_i)$$

**Learning rate**

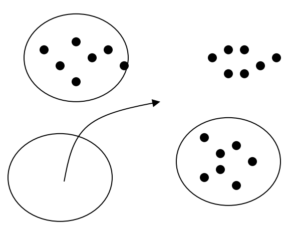


Small pull towards distant uncovered data

**EM**

$$\mu_i \leftarrow \frac{\sum_{l=1}^{N} h_{il} \mathbf{x}_l}{\sum_{l=1}^{N} h_{il}}$$

**No learning rate**



Renormalized – big jump in the first step

## 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**