# CENG 463
# Machine Learning

# Lecture 13
# Mixture of Gaussians

# Clustering

- **Non-parametric (clustering)**
  - No assumptions are made about the underlying densities, instead we seek a partition of the data into clusters
  - These methods are typically referred to as clustering
- **Parametric (mixture models)**
  - These methods model the underlying class-conditional densities with a mixture of parametric densities, and the objective is to find the model parameters
  - These methods are closely related to parameter estimation
  - **Remark**
    - A particular form of the mixture model problem leads to the most widely used clustering method: the k-means algorithm

# Mixture of Gaussians (MoG)

Mixture of Gaussians is a parametric clustering method.

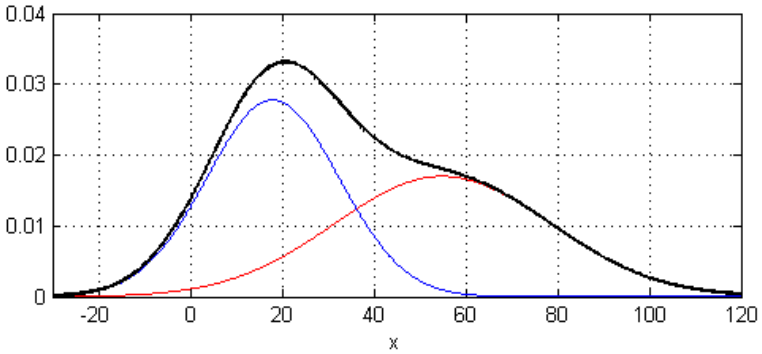
It combines **clustering** and **parameter estimation**.

Clusters are represented as Gaussian distributions, then the training set is assumed to be a mixture of Gaussians.

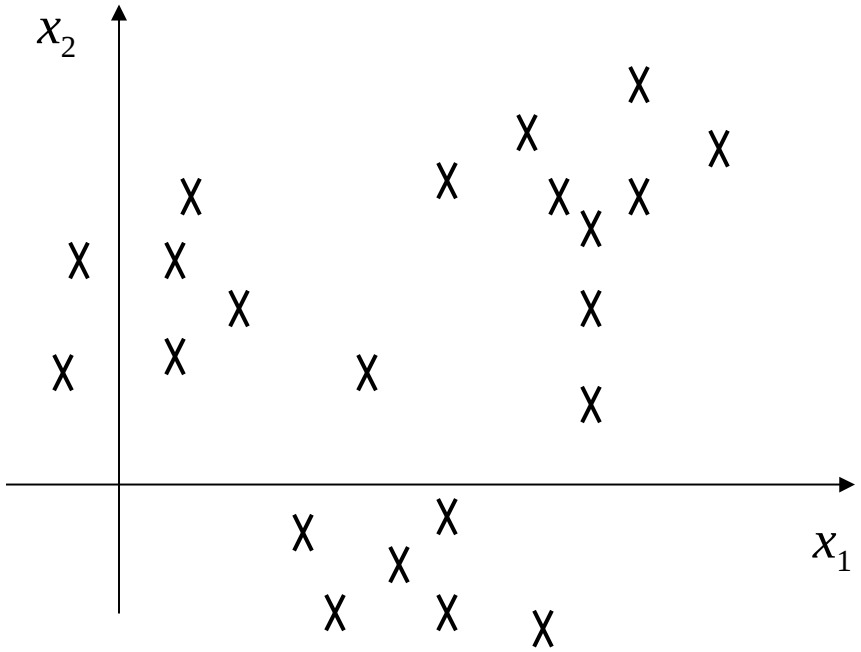
The number of clusters also assumed to be known!

**Mixture Density:**

$$P(x \mid \Theta) = \sum_{j=1}^{K} \underbrace{p(x \mid \Theta_j, C_j)}_{\text{component density}} p(C_j)$$

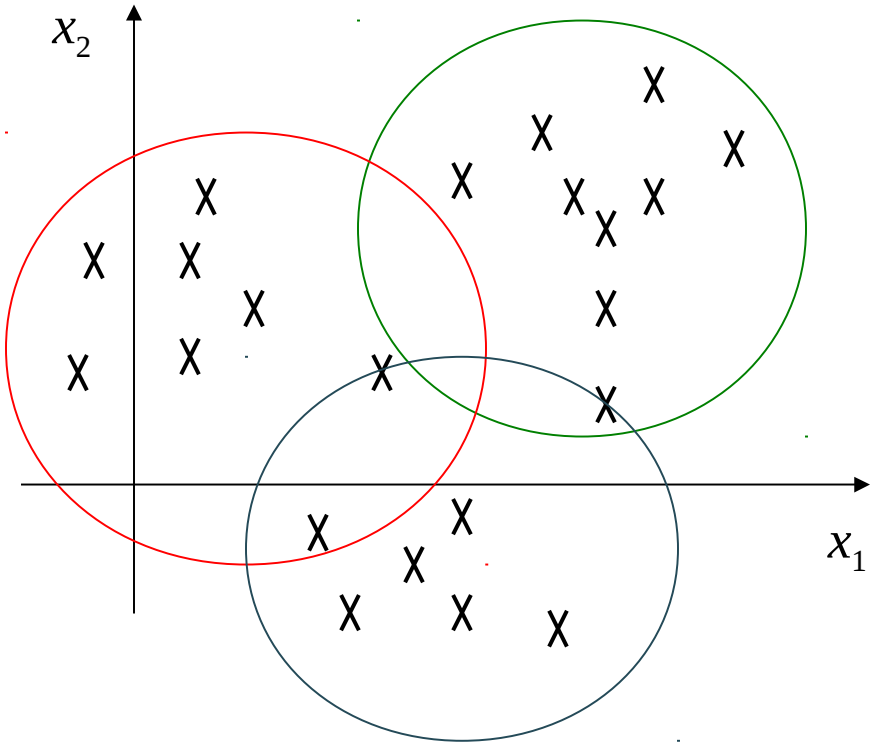


# Gaussian Assumption




**Assumption: We have three means.**

# Gaussian Assumption




**Assumption: We have three means. But these means are now viewed as means of *Gaussian* distribution**

# Gaussian Assumption

Each Gaussian also has a covariance matrix:

- We'll assume covariances of the form $\sigma^2$ for simplicity (please refer to Lecture 2 slides for multivariate Gaussian distributions)

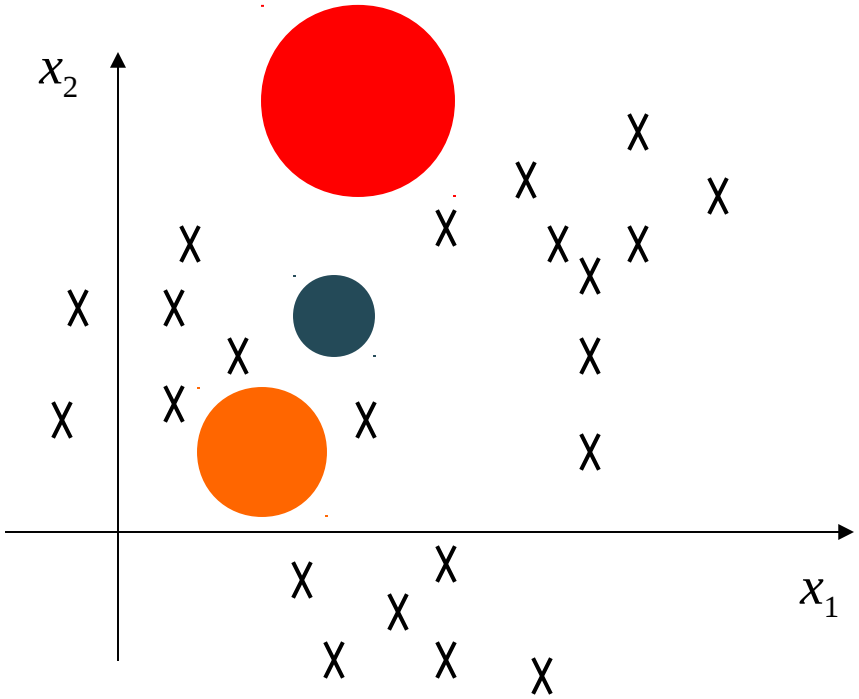
**Class-conditional probabilities:**

$$P(x \mid C_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2\sigma_i^2}\|x - \mu_i\|^2}$$

where $C_i$ is the class label for $i^{\text{th}}$ class

# Example

Consider a problem with three clusters

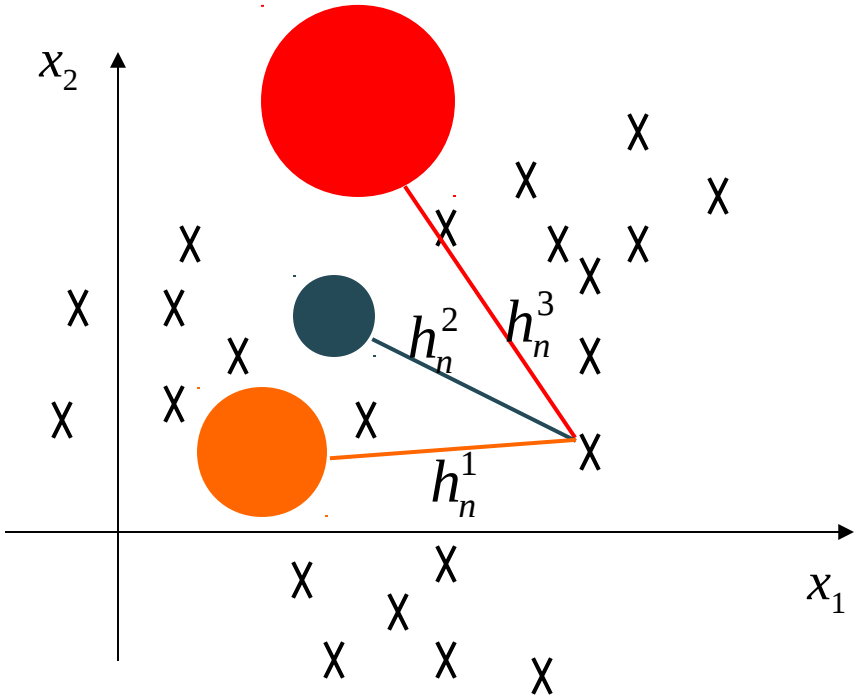


# Expectation-Maximization (EM) Algorithm

EM is an iterative algorithm with two steps:

- Expectation step
  - Compute the “responsibility” of each Gaussian for each data point (gives us a “soft partition”)
- Maximization step
  - Move the means of the Gaussians to the centroid of the data, weighted by the responsibilities

# Expectation Step



# Responsibility $h_n$

Responsibility of data point xn:

$$h_n^i = \frac{\frac{P(C_i)}{\sigma_i} e^{-\frac{1}{2\sigma_i^2}\|x_n - \mu_i\|^2}}{\sum_j \frac{P(C_j)}{\sigma_j} e^{-\frac{1}{2\sigma_j^2}\|x_n - \mu_j\|^2}}$$

is actually a consequence of Bayes rule:

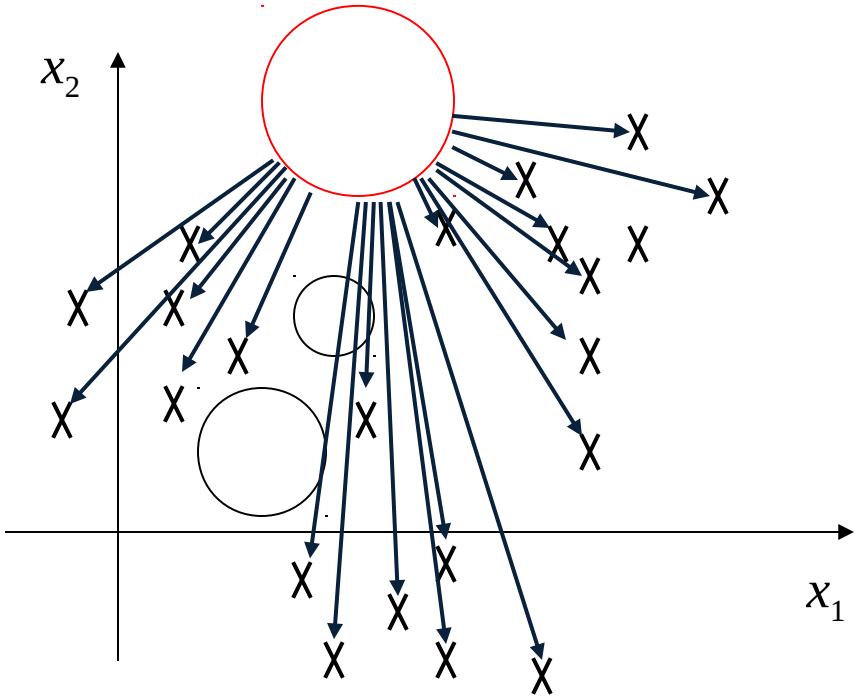
$$h_n^i \equiv P(C_i \mid x_n) = \frac{P(C_i)P(x_n \mid C_i)}{\sum_j P(C_j)P(x_n \mid C_j)}$$

# Responsibility $h_n$

When priors are **equal**, responsibility of data point $x_n$ :

$$h_n^i = \frac{\frac{1}{\sigma_i} e^{-\frac{1}{2\sigma_i^2}\|x_n - \mu_i\|^2}}{\sum_j \frac{1}{\sigma_j} e^{-\frac{1}{2\sigma_j^2}\|x_n - \mu_j\|^2}}$$

# Maximixation Step



# Maximixation Step

In maximixation step, each mean is moved to the centroid of all the data, *weighted by the responsibilities*:

$$\mu_i^{(t+1)} = \frac{\sum_N h_n^i x_n}{\sum_N h_n^i}$$

There are similar update rules for the covariances and for the priors.

# Relation with K-means

- In the limit of very narrow Gaussians, which yield posterior probabilities $h_n^i$ that are zeros and ones, EM reduces to the *K*-means algorithm.
- *K*-means can be a good way to initialize EM.
- Gaussian mixture models are probabilistic (“soft”) versions of *K*-means. By giving clustering a probabilistic foundation, we now see that we can replace “**clustering distance function**” by “**likelihood under a model**.”
- Essentially any model can be used to form the basic clustering element.

# Maximum Likelihood Justification

The EM algorithm doesn't just come out of a hat; it's a method of *maximum likelihood estimation:*

- Let $X=\{x_1, .., x_n\}$ be $N$ unlabeled samples drawn independently from a mixture distribution.
- The probability of a data point $x_n$ is given by the sum over all ways of obtaining the data point:

$$P(x_n \mid \Theta) = \sum_{j=1}^{K} p(x_n \mid \Theta_j, C_j) p(C_j)$$

# Maximum Likelihood Justification

Maximum likelihood:

$$\Theta_{ML} = L(\Theta \mid X) = \arg\max_{\Theta} p(X \mid \Theta) = \arg\max_{\Theta} \prod_{n=1}^{N} \sum_{j=1}^{K} p(x_n \mid \Theta_j, C_j) p(C_j)$$

Log likelihood: $$l(\Theta) = \log p(X \mid \Theta) = \sum_{n=1}^{N} \log \sum_{j=1}^{K} p(x_n \mid \Theta_j, C_j) p(C_j)$$

We differentiate log likelihood and perform optimization to find parameters that maximize ML.

The derivation is not shown here, but the steps lead us to the EM algorithm we have seen.