# Introduction to Machine Learning

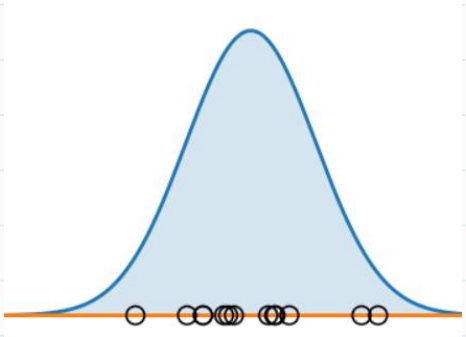
***Gaussian Mixture Model (GMM)***

**Prof. Subrahmanya Swamy Peruru**
Department of Electrical Engineering
IIT Kanpur

# Maximum Likelihood Estimation (MLE)

**Given:** i.i.d. samples $x_1, \dots, x_N$.
**Assume:** $x_n \sim \mathcal{N}(\mu, \sigma^2)$ with unknown $(\mu, \sigma^2)$.

$$p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$



**Likelihood:** Probability of observing the data given the parameters $\mu, \sigma^2$

$$L(\mu, \sigma^2) = p(x_1, \dots, x_N \mid \mu, \sigma^2) = \prod_{n=1}^{N} p\,(x_n \mid \mu, \sigma^2)$$

**MLE:** The parameters that maximize the likelihood function

$$\left(\hat{\mu}, \widehat{\sigma^2}\right) = \underset{\mu, \sigma^2}{\operatorname{argmax}}\ L(\mu, \sigma^2)$$

**Log-likelihood:** log preserves the maximizer and simplifies the objective (converts products to sums)

$$\ell(\mu, \sigma^2) = \log L(\mu, \sigma^2) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2.$$

# MLE for the Mean $\mu$ and Variance $\sigma^2$

**Differentiate log-likelihood w.r.t. $\mu$:**

$$\ell(\mu, \sigma^2) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 \qquad \Rightarrow \qquad \frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2}\sum_{n=1}^{N}(x_n - \mu)$$

Set to zero and solve:

$$\frac{1}{\sigma^2}\sum_{n=1}^{N}(x_n - \mu) = 0 \quad \Rightarrow \quad \boxed{\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n}.$$

**Differentiate log-likelihood w.r.t. $\sigma^2$:**

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{n=1}^{N}(x_n - \mu)^2.$$

Set to zero and solve:

$$-\frac{N}{\sigma^2} + \frac{1}{(\sigma^2)^2}\sum_{n=1}^{N}(x_n - \mu)^2 = 0 \quad \Rightarrow \quad \boxed{\widehat{\sigma^2} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu})^2}.$$

**Interpretation:** MLE estimates are equal to the sample average and variance.

# Multivariate Gaussian

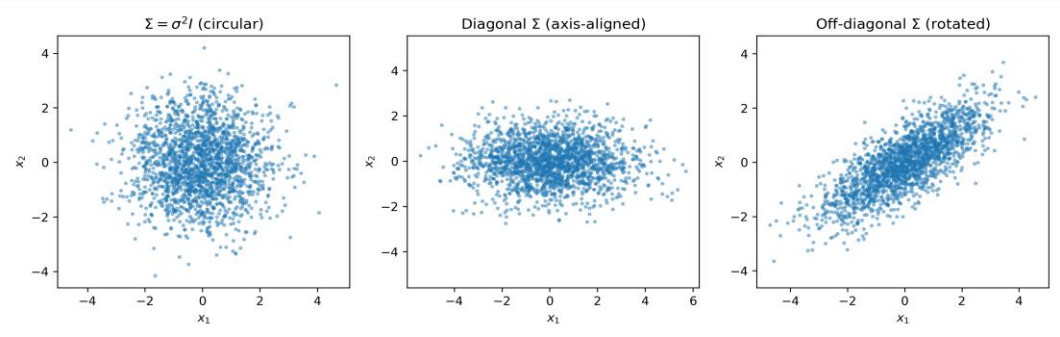
Now let $\mathbf{x}_n \in \mathbb{R}^d$ and assume i.i.d.

$$\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \boldsymbol{\Sigma} \succ 0.$$

**MLE estimates:**

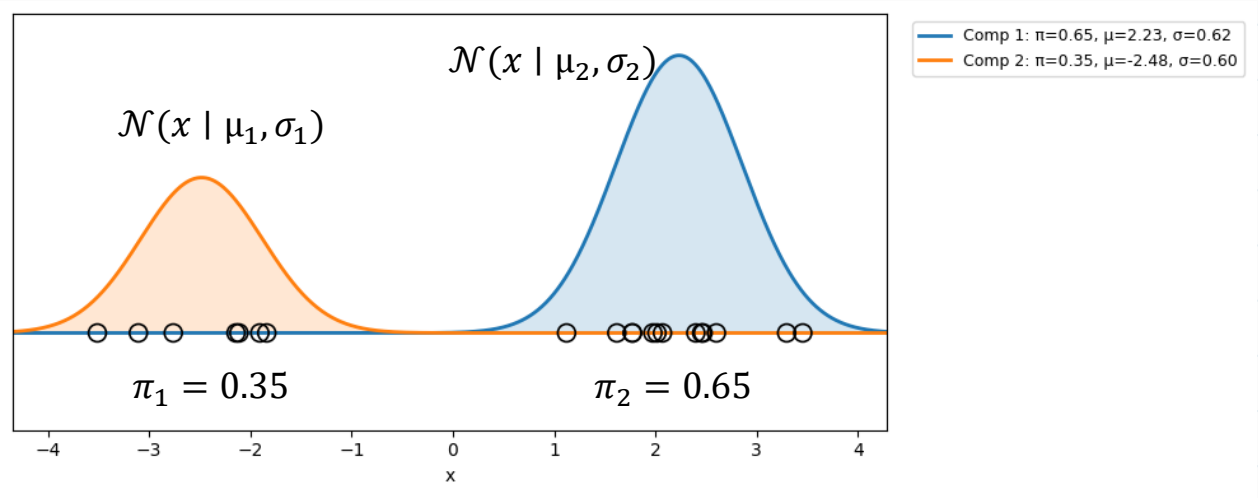
$$\boxed{\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n} \qquad \boxed{\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \hat{\boldsymbol{\mu}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}})^T}.$$

**Remark:** Same pattern as 1D — “mean = average”, “covariance = average outer product of centered data”.




$$\Sigma = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 3 & 0 \\ 0 & 0.8 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 2 & 1.2 \\ 1.2 & 1 \end{bmatrix}$$

# Gaussian Mixture Model (GMM)




When data is generated by a *mixture of more than one distributions*:

$$p(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x \mid \mu_k, \sigma_k), \qquad \pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1.$$

**Mixture weights** $\pi_k$ Probability that a randomly generated sample comes from component $k$.

Normal distribution parameters of $k^{th}$ component $\mu_k, \sigma_k$

# Clustering using Gaussian Mixture Model

**K-means mindset:** partition space into $K$ regions.

**GMM mindset:** Assume data is generated by a *mixture model*:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

**MLE:**

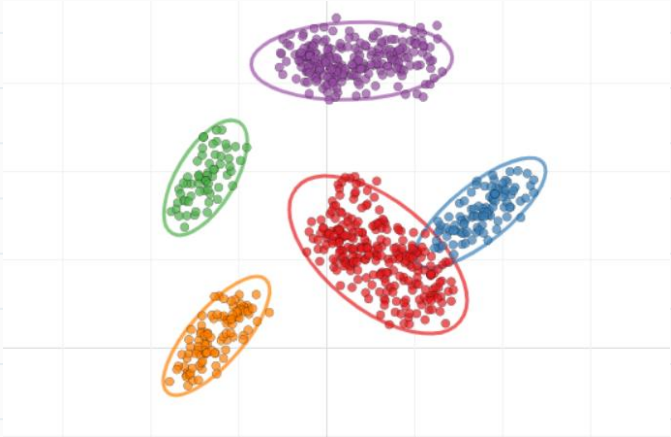
$$\arg \max_{\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\}} \ell(\theta) = \sum_{i=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

**Cluster centres** modelled via $\boldsymbol{\mu_k}$

**cluster sizes** modelled via $\pi_k$.

**shape + orientation** modelled via $\boldsymbol{\Sigma}_k$ (ellipses in 2D).

Gives **soft cluster assignments** (posterior probabilities), not hard labels (like K-means)



Demo: https://lukapopijac.github.io/gaussian-mixture-model/

# MLE for GMM

**Log-likelihood:** $\ell(\mu_k, \sigma_k, \pi_k) = \sum_{n=1}^{N} \log\left(\sum_{j=1}^{K} \pi_j \; \mathcal{N}\left(x_n \mid \mu_j, \sigma_j^2\right)\right)$

**Differentiate w.r.t.** $\mu_k$ (by applying chain rule):

$$\frac{\partial \ell}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)} \left(\pi_k \frac{\partial \, \mathcal{N}\left(x_n \mid \mu_k, \sigma_k^2\right)}{\partial \mu_k}\right)$$

Computing

$$\frac{\partial \, \mathcal{N}\left(x_n \mid \mu_k, \sigma_k^2\right)}{\partial \mu_k} = \frac{\partial}{\partial \mu_k} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu_k)^2}{2\sigma_k^2}\right)$$

$$= \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu_k)^2}{2\sigma_k^2}\right)\right)\left(\frac{x_n - \mu_k}{\sigma_k^2}\right)$$

$$= \mathcal{N}(x_n \mid \mu_k, \sigma_k^2) \frac{x_n - \mu_k}{\sigma_k^2}$$

$$\frac{\partial \ell}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)} \left(\pi_k \, \mathcal{N}(x_n \mid \mu_k, \sigma_k^2) \frac{x_n - \mu_k}{\sigma_k^2}\right)$$

$$\frac{\partial \ell}{\partial \mu_k} = \sum_{n=1}^{N} \frac{\pi_k \, \mathcal{N}(x_n \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)} \left(\frac{x_n - \mu_k}{\sigma_k^2}\right)$$

Define

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)}$$

$$\frac{\partial \ell}{\partial \mu_k} = \sum_{n=1}^{N} \gamma_{nk} \frac{x_n - \mu_k}{\sigma_k^2}$$

Set derivative to zero:

$$\frac{1}{\sigma_k^2} \sum_{n=1}^{N} \gamma_{nk} \, (x_n - \mu_k) = 0 \;\Rightarrow\; \sum_{n=1}^{N} \gamma_{nk} \, x_n = \mu_k \sum_{n=1}^{N} \gamma_{nk} \, .$$

$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} \, x_n}{\sum_{n=1}^{N} \gamma_{nk}}}$$

# Derivatives w.r.t $\sigma_k^2$ and $\pi_k$

From the previous slide:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)}$$

$$\frac{\partial \ell}{\partial \mu_k} = 0 \quad \Rightarrow \quad \boxed{\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk}\, x_n}{\sum_{n=1}^{N} \gamma_{nk}}}$$

**Note:**

- The variables $\{\gamma_{nk}\}$ depend on the unknown parameters $\{\mu_k, \sigma_k^2, \pi_k\}$.
- The parameters depend on $\{\gamma_{nk}\}$
- So the equations are coupled and cannot be solved directly in closed form.

Similarly if we take the derivates w.r.t $\sigma_k^2$ and $\pi_k$ we obtain:
For variance:

$$\frac{\partial \ell}{\partial \sigma_k^2} = 0 \quad \Rightarrow \quad \boxed{\sigma_k^2 = \frac{\sum_{n=1}^{N} \gamma_{nk}\, (x_n - \mu_k)^2}{\sum_{n=1}^{N} \gamma_{nk}}}$$

For mixing coefficients (use Lagrange multiplier since we have a constraint $\sum_k \pi_k = 1$):

$$\boxed{\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}}$$

# Iterative Algorithm for GMM

Initialize the parameters

**Iterate these two steps until convergence:**

1. Given the distribution parameters $\{\mu_k, \sigma_k, \pi_k\}$, update the cluster assignment probabilities:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \sigma_k^2)}{\sum_j \pi_j \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)}$$

2. Given the soft cluster assignments $\{\gamma_{nk}\}$, update the distribution parameters:

$$\mu_k = \frac{\sum_n \gamma_{nk}\, x_n}{\sum_n \gamma_{nk}}, \quad \sigma_k^2 = \frac{\sum_n \gamma_{nk}\, (x_n - \mu_k)^2}{\sum_n \gamma_{nk}}, \quad \pi_k = \frac{\sum_n \gamma_{nk}}{N}$$

Demo

**Soft Cluster assignments:** This formula is nothing but **Bayes' rule** — $\gamma_{nk}$ gives the posterior probability that a point $x_n$ belongs to cluster $k$

**Weighted updates:** Instead of hard counts, we use fractional (soft) counts $\gamma_{nk}$ to compute weighted means and variances.

This algorithm is an instance of a general framework called **Expectation-Maximization (EM)**

**Note:**

- EM converges to a local maximum of the log-likelihood.
- Hence multiple runs with different initializations are recommended.

# K-means vs GMM

**K-means (hard clustering)**

*Models centres via* $\{\boldsymbol{\mu_k}\}$

Assumes equal sizes

Assumes spherical shapes for all

**Iterate:**

1. **Assign (hard):**

$$r_{nk} = 1 \text{ if } k = \text{argmin}_j \, ||\mathbf{x}_n - \boldsymbol{\mu_j}||^2$$

2. **Update cluster parameters:**

$$\boldsymbol{\mu_k} \leftarrow \frac{\sum_n r_{nk}\, \mathbf{x_n}}{\sum_n r_{nk}}$$

**GMM (probabilistic clustering)**

*Cluster Center* $\{\boldsymbol{\mu_k}\}$

*Cluster sizes* $\{\pi_k\}$

Allows different shapes and orientations $\{\boldsymbol{\Sigma_k}\}$

**Iterate:**

1. **Assign (soft):**

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma_k})}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma_j})}.$$

2. **Update cluster parameters :**

$$\pi_k \leftarrow \frac{\sum_n \gamma_{nk}}{N}, \quad \boldsymbol{\mu_k} \leftarrow \frac{\sum_n \gamma_{nk}\, \mathbf{x_n}}{\sum_n \gamma_{nk}}, \quad \boldsymbol{\Sigma_k} \leftarrow \frac{\sum_n \gamma_{nk}\, (\mathbf{x_n} - \boldsymbol{\mu}_k)(\mathbf{x_n} - \boldsymbol{\mu}_k)^\top}{\sum_n \gamma_{nk}}$$

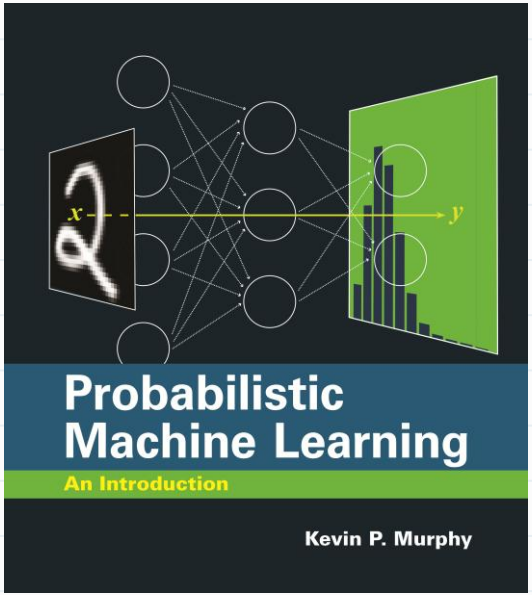
**Summary:**

- Same two-step loop (cluster assignment ↔ parameter update)
- Both require multiple random initializations; pick best solution
- Number of clusters $K$ must be tuned if not specified

# Reference

**Chapter 21**

of “Probabilistic Machine Learning: An Introduction” book by Kevin P. Murphy



# Thank You