GAUSSIAN MIXTURE MODEL, EXPECTATION-MAXIMIZATION ALGORITHM

PRACTICALS MACHINE LEARNING 1, SS23

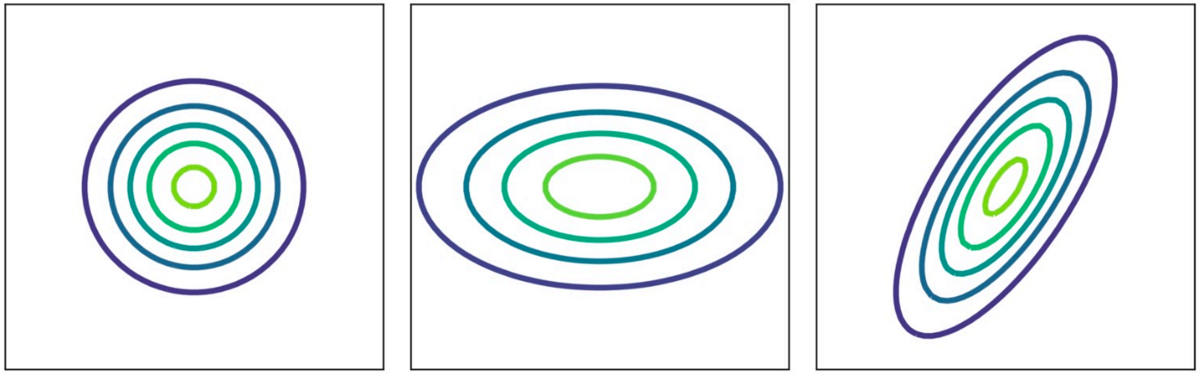
GAUSSIAN MIXTURE MODEL (GMM)

The Gaussian distribution

1-dimensional: $\mathcal{N}(x \mid \mu, \sigma^2) = \dfrac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

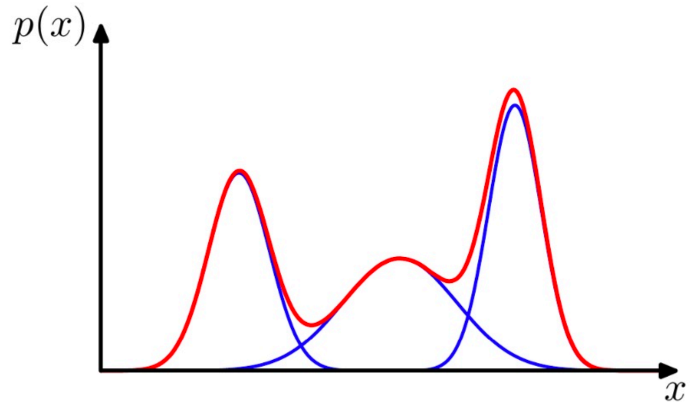
D-dimensional: $\mathcal{N}(x \mid \mu, \Sigma) = \dfrac{1}{(2\pi)^{\frac{D}{2}}} \dfrac{1}{|\Sigma|^{\frac{D}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$

2-dimensional:



$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$ $\Sigma = \begin{bmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{22}^2 \end{bmatrix}$ $\Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix}$

Gaussian mixtures



- Idea: Create mixture distributions
- Superposition of K Gaussians

$$p(x \mid \Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

- Mixing coefficients (also called weights) $0 \leq \pi_k \leq 1$, and

$$\sum_{k=1}^{K} \pi_k = 1$$

The scaling by π_k ensures that $\int_{-\infty}^{\infty} p(x \mid \Theta)\, dx = 1$.

- $\Theta = \{\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K\}$

Properties of Gaussian Mixture Models

- Rule of total probability that relates marginal probabilities to conditional probabilities:

$$p(x) = \sum_{k=1}^{K} p(x \mid k) p(k)$$

- Posterior probabilities (*responsibilities*):

$$p(k \mid x) = \frac{p(k)p(x \mid k)}{p(x)} = \frac{p(k)p(x \mid k)}{\sum_{j=1}^{K} p(j)p(x \mid j)}$$

Properties of Gaussian Mixture Models

- Rule of total probability that relates marginal probabilities to conditional probabilities:

$$p(x) = \sum_{k=1}^{K} p(x \mid k) p(k)$$

- Posterior probabilities (*responsibilities*):

$$p(k \mid x) = \frac{p(k)p(x \mid k)}{p(x)} = \frac{p(k)p(x \mid k)}{\sum_{j=1}^{K} p(j)p(x \mid j)}$$

$$\gamma_k(x) = p(k \mid x)$$

$$\gamma_k(x) = \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)}$$

Estimation of parameters Θ

- By Maximum Likelihood method

- Given $X = \{x_1, \dots, x_N\},\ x \in \mathbb{R}^D$,
  estimate the parameters $\Theta_{ML} = \arg\max_{\Theta} \ln p(X \mid \Theta)$

- Log-likelihood function:

$$L(X \mid \Theta) = \ln p(X \mid \Theta) = \sum_{n=1}^{N} \ln p(x_n \mid \Theta)$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)$$

Estimation of parameters Θ

Find $\dfrac{\partial \ln p(X \mid \Theta)}{\partial \Theta} \overset{!}{=} 0$

…

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

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

where $N_k = \sum_{n=1}^{N} \gamma_{nk}.$

EXPECTATION-MAXIMIZATION (EM) ALGORITHM

EM algorithm

- Goal: **maximize** the likelihood function w.r.t. the parameters $\Theta = \{\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K\}$.
- Initialize Θ
- Iteratively (repeat) until convergence:

(1) (Expectation) **E-step:** Evaluate the responsibilities γ_{nk} using the current parameter values Θ

(2) (Maximization) **M-step:** Re-estimate the parameters using the current responsibilities γ_{nk}

Evaluate the log-likelihood function:

$$\ln p(X \mid \Theta) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)$$

EM algorithm

(1) E-step: Evaluate the responsibilities using the current parameter values Θ

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

EM algorithm

(2) M-step: Re-estimate the parameters using the current responsibilities γ_{nk}

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

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k^{\text{new}})(x_n - \mu_k^{\text{new}})^T,$$

$$\pi_k^{\text{new}} = \frac{N_k}{N},$$

where $N_k = \sum_{n=1}^{N} \gamma_{nk}$.

EM algorithm - Properties

- The log-likelihood $\ln p(X \mid \Theta)$ is monotonically increasing with each iteration
- A local optimal solution found (no guarantee for finding the global optimum)
- The solution depends on the initialization of Θ

Relation to K-Means algorithm

- There is a close similarity
- K-Means algorithm performs a hard assignment of data points to clusters (“**hard-clustering**”)
- EM algorithm makes a soft assignment based on the posterior probabilities (“**soft clustering**”)
- K-Means can be derived as a particular limit of EM for Gaussian mixtures:
  - Covariance matrices of mixture components are diagonal matrices, with *a variance parameter shared* by all of the components, and *fixed*.
  - Responsibilities would be values with a limit 0 or 1 (binary indicator variable r_{nk} in the K-Means).