Gaussian Mixture Model (GMM)

Assumptions:

(1) 1-hot-encoded discrete latent variable $z_k \in \{0,1\}$ for the K clusters, with prior

$$p(z_k = 1) = \pi_k \quad \pi_k \in [0,1] \quad \boxed{\sum_{k=1}^{K} \pi_k = 1}$$

(2) The clusters are Gaussian with different parameters

$$p(x \mid z_k = 1) = \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

(3) the joint distribution

$$p(x, z_k = 1) = p(x \mid z_k = 1)\, p(z_k = 1)$$
$$= \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

(4) full generative model

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

Given the data $X = \{x_1, x_2 \cdots x_N\}$

$$\ln p(x \mid \pi, \mu, \Sigma) = \ln \prod_{n=1}^{N} p(x_n \mid \pi, \mu, \Sigma)$$

$$\ln p(X|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln p(x_n|\pi,\mu,\Sigma)$$

$$\ln p(X|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \boxed{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}$$

because of this summation term
It's difficult to get a closed form
of the likelihood.
One alternative is to go for EM
(expectation Maximization algorithm)

Multivariate Gaussian:

$$\mathcal{N}(x|\mu_k,\Sigma_k) = \frac{1}{(2\pi)^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)}$$

$$\frac{\partial \mathcal{N}(x|\mu_k,\Sigma_k)}{\partial \mu_k} = \mathcal{N}(x|\mu_k,\Sigma_k)(x-\mu_k)^T\Sigma^{-1}$$

$$\frac{\partial \ln p(X|\pi,\mu,\Sigma)}{\partial \mu_k} = \sum_{n=1}^{N} \frac{\partial}{\partial \mu_k}\left[\ln \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)\right]$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)} \frac{\partial}{\partial \mu_k}\left[\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)\right]$$

$$= \sum_{n=1}^{N} \boxed{\frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}} (x_n-\mu_k)^T\Sigma^{-1} = 0$$

↳ γ_{nk} this is posterior probability or responsibilities.

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)^T \Sigma^{-1} = 0$$

$$\Rightarrow \sum_{n=1}^{N} \gamma_{nk} x_n = \sum_{n=1}^{N} \gamma_{nk} \mu_k$$

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

μ_k = the weighted average over the points x_n for which cluster k takes the responsibility.

Maximize w.r.t π_k

but we need to take care of constraint that $\sum_{k=1}^{K} \pi_k = 1$, this can be achieve by lagrange multiplier.

$$L = \sum_{n=1}^{N} \ln p(x_n | \pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$\frac{\partial L}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\partial}{\partial \pi_k} \left(\ln \left(\sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right) \right) + \lambda$$

$$= \sum_{n=1}^{N} \frac{1}{\pi_k} \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k)} + \lambda = 0$$

$$\Rightarrow \frac{1}{\pi_k} \sum_{n=1}^{N} r_{nk} + \lambda = 0$$

$$\Rightarrow \boxed{\pi_k = -\frac{1}{\lambda} \sum_{n=1}^{N} r_{nk}}$$

So, π_k also depends on λ.

$$\text{So, } \frac{\partial}{\partial \lambda} L(\pi_k, \lambda) = \frac{\partial}{\partial \lambda} \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) = 0$$

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

$$\Rightarrow \sum_{k=1}^{K} -\frac{1}{\lambda} \sum_{n=1}^{N} r_{nk} = 1$$

$$\Rightarrow \lambda = -\sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk}$$

$$\Rightarrow \boxed{\lambda = -N}$$

$$\boxed{\sum_{k=1}^{K} r_{nk} = 1}$$

$$\text{So, } \boxed{\pi_k = \frac{1}{N} \sum_{n=1}^{N} r_{nk}}$$

fraction of points for which cluster k takes responsibility

So, for Σ_K

$$\Sigma_K = \frac{1}{N_K} \sum_{n=1}^{N} r_{nk} (x_n - \mu_k)(x - \mu_k)^T$$ ✓

Ex:

$P(A) = 1/2$

$P(B) = \mu$ $\quad 0 \le \mu \le 1/6$

$P(C) = 2\mu$

$P(D) = 1/2 - 3\mu$

estimate μ from the data given.

$$P(D|\theta) = \underset{\theta}{\operatorname{argmax}}\ P(\theta|D)$$

$$L = P(a,b,c,d|\mu) = (1/2)^a (\mu)^b (2\mu)^c (1/2 - 3\mu)^d$$

by likelihood

$$\frac{\partial L}{\partial \mu} = \frac{\partial}{\partial \mu}\left[\log (1/2)^a + \log(\mu)^b + \log(2\mu)^c + \log(\tfrac{1}{2} - 3\mu)^d\right] = 0$$

$$\Rightarrow \frac{\partial}{\partial \mu}\left[a \log 1/2 + b \log \mu + c \log 2\mu + d \log(\tfrac{1}{2} - 3\mu)\right] = 0$$

$$\Rightarrow \frac{b}{\mu} + \frac{c}{2\mu} + \frac{d}{\frac{1}{2} - 3\mu} = 0$$

$$\Rightarrow \frac{1}{\mu}\left(b + \frac{c}{2}\right) + \frac{2d}{1 - 6\mu} = 0$$

$$\Rightarrow \quad \frac{1-6u}{u}\left(b+\frac{c}{2}\right)+2d=0$$

$$\Rightarrow \left(\frac{1}{u}-6\right)\left(b+\frac{c}{2}\right)+2d=0$$

$$\Rightarrow \quad \frac{b}{u}+\frac{c}{2u}-6b-3c+2d=0$$

$$\Rightarrow \quad \frac{1}{u}\left(b+\frac{c}{2}\right)=6b+3c-2d$$

$$\Rightarrow \quad u=\frac{2b+c}{2(6b+3c-2d)}$$

SVM

$$w^*, b^* = \underset{w,b}{\operatorname{argmin}} \frac{\|w\|}{2} + C\sum_{i=1}^{n} \varepsilon_i \quad \text{such that}$$

$$y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

$$\& \quad \varepsilon_i \geq 0$$

for correctly classified point $\varepsilon_i = 0$.

using lagrangian formulation for this optimization problem.

Introducing two lagrangian multiplier for each inequality constraint.

$$L(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \frac{\|w\|}{2} + C\sum_{i=1}^{n} \varepsilon_i - \sum_{i=1}^{n} \alpha_i \{y_i(w^T x_i + b) - 1 + \varepsilon_i\} - \sum_{i=1}^{n} \beta_i \varepsilon_i$$

Take derivatives of L w.r.t w, b, ε_i

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\Rightarrow \boxed{w = \sum_{i=1}^{n} \alpha_i y_i x_i}$$

$$\frac{\partial L}{\partial b} = \boxed{\sum_{i=1}^{n} \alpha_i y_i = 0}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \Rightarrow \boxed{C = \alpha_i + \beta_i}$$

Now Substitute the values.

$$L(w^*, b^*) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{x_i^T x_j}$$

s.t $0 \le \alpha_i \le C$

dual form

$x_i^T x_j$ can be written as cosine-sim(x_i, x_j)

Observations:

1. for every x_i there is an α_i
2. x_i's only occur in the form of $x_i^T x_j$
3. $f(x_q) = \sum_{i=1}^{n} \alpha_i y_i x_i^T x_q + b$
4. $\alpha_i > 0$ only for support vectors
   $= 0$ for all non-support vectors.

at runtime

$$f(x_q) = \sum_{i=1}^{n} \alpha_i y_i (x_i^T x_q) + b$$

$\rightarrow K(x_i, x_q)$

this is similarity function, can be any similarity.

Replacing this form (dot product) to $K(x_i, x_j)$ is called kernel trick & such SVM is called Kernel SVM.

Linear SVM : $x_i^T x_j$

Kernel SVM : $K(x_i, x_j)$