Gaussian Mixture Model

~~lets look at it~~

lets consider the data to be in 1-dimension

suppose 'N' number of points in 1-d is given and we are asked to divide it into 'k' clusters.

so we estimate μ (mean) and σ^2 (variance) for each k

using E-M Algorithm (Expectation-maximization) we estimate the model parameter ~~iteratively~~ iteratively.

E - estimation step
M - Maximization step

Implementation of em algorithm for GMM in 1D is given below:

initialization:

Initial means (μ_k) → choose ~~using~~ random point ~~given~~ from given for each k

Initial variances (σ^2_k) → variance of the given points.

Initial mixing coefficient (π_k) → $1/K$

mixing coefficient defines how much points among given points belong to a respective cluster

$$\sum^{K} \pi_k = 1$$

pdf for univariate GD :-

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

~~σ - variance~~

if log :-

$$-\frac{1}{2} \times \log(2\pi) - \log(\sigma) - \left(\frac{(x-\mu)^2}{2\sigma^2}\right)$$

E-step

compute the γ_{nk} (i.e probability that data point x_n belongs to k)

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

Using log-sum-exp trick to handle small probability (when x_n is far from μ_k)

$$\log(\gamma_{nk}) = \log(\pi_k) + \log(N(x_n \mid \mu_k, \sigma_k^2)$$

$$- \log \left(\sum_{j=1}^{K} \pi_{j} N(x_n | \mu_j, \sigma_j^2) \right)$$

Also

$$\log \left(\sum_{j=1}^{K} \pi_j N(x_n | \mu_j, \sigma_j^2) \right)$$

$$= \log \left(\sum_{j=1}^{K} \exp\left(\log(\pi_j) + \log(N(\cdots)) \right) \right)$$

Apply it in find equation of $\log(\gamma_{nk})$

M-step

update parameters based on γ_{nk}

$$\pi_k = \frac{\sum \gamma_{nk}}{N}$$

these are parameters for each k cluster

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

$$\sigma_k^2 = \frac{\sum^{N} \gamma_{nk} (x_n - \mu_k)^2}{\sum^{N} \gamma_{nk}}$$

Using the log-sum trick:

Or $$\pi_k = \frac{\sum^{N} \exp(\log(\gamma_{nk}))}{N}$$

$$\mu_k = \frac{\sum_{n=1}^{N} (\exp(\log(r_{nk})) \cdot x_n)}{\sum^{N} \exp(\log(r_{nk}))}$$

$$\sigma_k^2 = \frac{\sum_{n=1}^{N} (\exp(\log(r_{nk})) . (x_n - \mu_k)^2)}{\sum^{N} \exp(\log(r_{nk}))}$$

compare the values of parameters ~~[illegible]~~ with previous iteration, if the difference is below a threshold, we can say its ~~converging~~ converging. (so we can stop)

In my code, I didnt consider changing π_k as ~~[illegible]~~ the initial mean is calculated randomly and not by k-means.

Other converging criteria is the use of log likelihood ~~[illegible]~~

$$\Rightarrow \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \sigma_k^2) \right)$$

~~using log-sum-exp trick~~

if its change in adjacent iteration are below a threshold we can stop.

finally , for a point x_1

if $r_{1k_1} > r_{1k_2}$, then we can say

that x_1 belongs to k_1 (probability is high)

μ_k and σ_k gives us the final params of k gaussian distributions

Note: In my code all the parameters must converge and also log likelihood must be stable to end the iterations.