# Gaussian Mixture Models (GMMs)

Setting: unsupervised learning.

$\{x^{(1)}, x^{(2)}, \ldots x^{n}\}$

No ground-truth labels available.

## Recap of K-means.

clustering approach ⇒ goal → group similar data points together (in a cluster)

$$\min \sum_{i=1}^{n} \|x_i - \mu_i\|^2$$

$x_i$: datapoint; $\mu_i$: centroid (mean).

outcome was $x_i$ belongs to cluster K

⇓ resulted in

hard clustering.

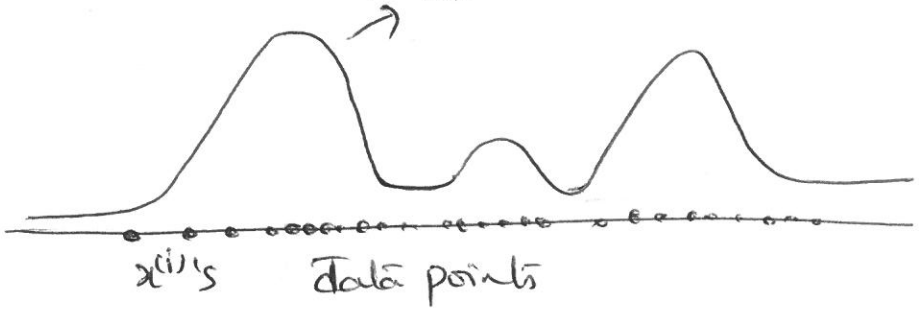
And also, no model assumption was made. all we did was to try and minimize the distance between $x_i$ and centroid ($\mu_i$) of the cluster to which $x_i$ belongs to.

What if we want soft clustering??

can we come up with a probability based approach?? and also can we assume a model from which data points $x^{(i)}$ can be drawn.

Graphical representation

Fig. 1



Lets make some assumptions.

Based on ~~the~~ Fig1 we view $x^{(1)}, x^{(2)} \ldots x^{(n)}$ as random variables drawn from an unknown distribution with density $p(x)$

Such that $p(x) = \sum_{j=1}^{k} \pi_j \underbrace{w_i(x ; \mu_j, \Sigma_j)}_{\text{component}}$ ← some distribution

~~However, coming up~~

For GMM, this distribution $p(a)$ has a mixture of $K$ components

- $w(x ; \mu_j, \Sigma_j) \rightarrow$ Gaussian (Normal) distribution with unknown parameters $(\mu_j, \Sigma_j)$

⇓

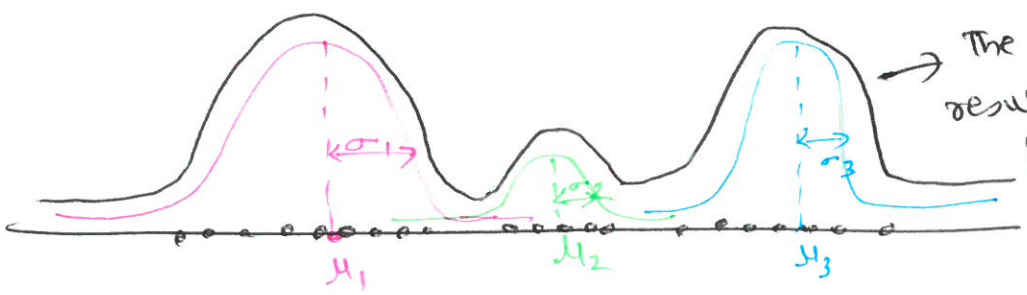
each one is a multivariate Gaussian distribution.

$\pi_j \rightarrow$ unknown probability of selecting component $j$

Such that $\sum_{j=1}^{K} \pi_j = 1$ (Total probability should add up to 1).

Graphical rep.

Fig2




covariance $\Sigma_2 \approx \sigma^2$ similarly for GD1, GD3

Let us now consider a latent component indicator $z^{(i)}$
(hidden/unobserved) (latent class/cluster for datapoint $x_i$)

See $z_i$'s as labels in supervised learning setting.

But in our case these $z_i$'s are not available or we don't (an supervised setting) get to see them.

$z^{(i)} \sim$ multinomial $(w_k) \rightarrow \{w_1, w_2 \cdots w_K\}$ K components

↓ in binary classification $z^{(i)}$ takes 1 or 0.

Here it takes probability of datapoint $x^{(i)}$ coming from a specific cluster

$$x^{(i)}|_{z^{(i)}} \approx \mathcal{N}(\mu_{z^{(i)}}, \Sigma_{z^{(i)}})$$

$x^{(i)}$ conditioned to $z^{(i)}$ → prob. of $x^{(i)}$ being observed in cluster $z^{(i)}$

↓ Gaussian/normal distribution

---

Now if we knew $z^{(i)}$, it becomes supervised problem we can go ahead and calculate joint distribution

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|_{z^{(i)}}) \, p(z^{(i)})$$

The problem for us is we do not know $z^{(i)}, \mu_j, \Sigma_j$

$\mu_j, \Sigma_j$ → helps us calculate → $z$ → helps us calculate → $\mu_j, \Sigma_j$

So how do we solve this??

# Estimation – Maximization (EM) algorithm.

Similar to what we did in k-means

1. Estimate centroids (Random select)
2. compute new centroids based on data point assignments.

In EM we have 2 steps

Initialize $w, \mu, \Sigma$ Randomly.

Until convergence

E Step :- compute values of $z^{(i)}$

for each i, j set (train & component set)

$$w_j^{(i)} = p\left(z^{(i)} = j \mid x^{(i)} ; w, \mu, \Sigma\right)$$

gives the class memberships (soft allocations)

$$= \frac{p\left(x_i \mid z_i = j, \mu, \Sigma\right) p\left(z^{(i)} = j ; w\right)}{\sum_{l=1}^{k} p\left(x^{(i)} \mid z^{(i)} = l ; \mu, \Sigma\right) p\left(z^{(i)} = l ; \phi\right)}$$

gives the prob. of $x^{(i)}$ observed by Gaussian $(\mu_j, \Sigma_j)$

probability of Gaussian j $(w_j)$ (prior)

Total probability over all components.

M-Step :- update parameters based on Gaus.

$$w_j := \frac{1}{n} \sum_{i=1}^{n} w_j^{(i)} \quad ; \quad \mu_j := \frac{\sum_{i=1}^{n} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{n} w_j^{(i)}}$$

→ update means

updated weights

no. of samples

$$\Sigma_j := \frac{\sum_{i=1}^{n} w_j^{(i)} \left(x^{(i)} - \mu_j\right)\left(x^{(i)} - \mu_j\right)^T}{\sum_{i=1}^{n} w_j^{(i)}}$$

Update covariance

Repeat.

until no new assignments of data points to k components.