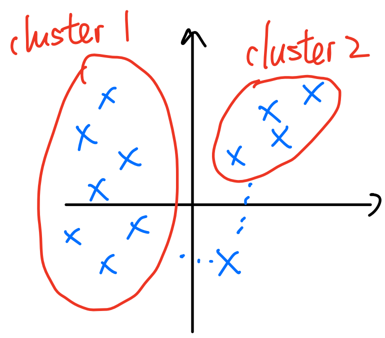
1. What is GMM?

2. Inference – assign datapoint to a cluster

3. Learning – decide $\underbrace{\text{mean}}_{\text{location}}$ & $\underbrace{\text{covariance}}_{\text{shape}}$ of each cluster




Data: $x^{(1)}, \dots, x^{(n)}$

How is the data generated?

1. randomly choose cluster 1 or 2

↑ latent variable

2. sample from a (multivariate) Gaussian distribution for the choose cluster

For this dataset : # cluster $= k = 2$

$$\begin{cases} \pi_1 = 2/3, \quad \pi_2 = 1/3 \\ \mu_1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\ \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \end{cases}$$

← $\pi_i$ : prob of choosing cluster $i$ in step 1

params to learn from data

Inference: infer the probability distribution of a $\underbrace{\text{latent}}_{\text{unobserved in the data}}$ random variables

Learning: fitting parameters

Terminology: For each $i = 1, \dots, n$

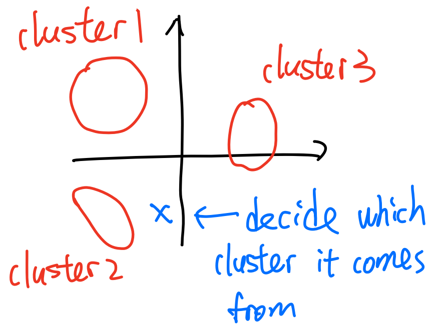
$Z_i$ is latent variable denoting cluster choice $\in \{1, \dots, k\}$

$X_i$ is random variable we observe as $x^{(i)}$

Inference problem: Given

- $X_i = x^{(i)}$
- knowing $\pi_{1:k}, \mu_{1:k}, \Sigma_{1:k}$

Compute $P(Z_i \mid X_i = x^{(i)}; \pi_{1:k}, \mu_{1:k}, \Sigma_{1:k})$




$$P(Z_i = c \mid X_i = x^{(i)})$$

$$= \frac{P(Z_i = c)\, P(X_i = x^{(i)} \mid Z_i = c)}{\sum_{b=1}^{k} P(Z_i = b)\, P(X_i = x^{(i)} \mid Z_i = b)}$$

① $P(Z_i = c) = \pi_c$

② $P(X_i = x^{(i)} \mid Z_i = c)$ Gaussian w/ mean $\mu_c$, covariance $\Sigma_c$

conditioned on being in cluster $c$, what is prob of observing $x^{(i)}$

$$= \frac{1}{(2\pi)^{d/2}} \cdot \frac{1}{\sqrt{\det(\Sigma_c)}} \cdot \exp\left(-\frac{1}{2} \cdot (x^{(i)} - \mu_c)^T \Sigma_c^{-1} (x^{(i)} - \mu_c)\right)$$

d : dimension of data

compare with univariable Gaussian :

$$\frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\sigma^2}} \cdot \exp\left(-\frac{1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}\right)$$

## Learning GMMs:

Comparison with k-means

| k-means | Gaussian-mixture |
|---|---|
| • Assignments ('hard assignment') | Latent variables ('soft ...') |
| • Centroids | parameters |
| • alternate between update assignments / centroids | expectation-maximization |

Expectation - Maximization (EM)

Generally used when have both

- Latent variables
- unknown parameters

① E-step: infer latent variables distributions given current guess of params ] ≈ make assignments

② M-step: choose best params that fit the data based on the inferred dist. of latent variables ] choose centroids

EM for GMM's

E-step For each $i = 1, \dots, n$, infer dist. for $z_i$

call $r_{ic} = P(z_i = c \mid x_i = x^{(i)}; \text{current params guess})$

$$\begin{array}{c|ccc|}
i=1 & 0.6 & 0.3 & 0.1 \\
\vdots & & R & \\
n & & & \\
\hline
 & 1 & 2 & 3 \quad \text{cluster}
\end{array}$$

M-step: we have $\begin{cases} \text{actual value of all the } x_i\text{'s} \\ \text{dist. for each } z_i \end{cases}$

$\Rightarrow$ can't do MLE

we will maximize Expected Complete Loglikelihood (ECLL)

$$ECLL(\pi_{1:k}, \mu_{1:k}, \Sigma_{1:k}) =$$

$$\underbrace{\sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic}}_{\text{"expected"}} \log \underbrace{p(x_i = x^{(i)}, z_i = c}_{\text{"complete" b/c compute likelihood of both } x_i \text{ and } z_i} ; \pi, \mu, \Sigma)$$

What choice of $\pi_{1:k}, \mu_{1:k}, \Sigma_{1:k}$ maximize ECLL?

start with $\mu_1$ $(\mu_2 \cdots \mu_k)$

Taking gradient with $\mu_1$, set to 0

$$\nabla_{\mu_1} ECLL = \sum_{i=1}^{n} r_{i1} \nabla \log P(x_i = x^{(i)}, z_i = 1)$$

$$= \sum_{i=1}^{n} r_{i1} \nabla [\underbrace{\cancel{\log p(z_i = 1)}}_{\text{doesn't depend on } \mu_1} + \log p(x_i = x^{(i)} | z_i = 1)]$$

$$= \sum_{i=1}^{n} r_{i1} \nabla \log p(x_i = x^{(i)} | z_i = 1)$$

$$\cancel{\frac{1}{(2\pi)^{d/2}}} \cdot \cancel{\frac{1}{\sqrt{\det(\Sigma_c)}}} \cdot \exp\left(-\frac{1}{2}\cdot(x^{(i)}-\mu_c)^T \Sigma_c^{-1} (x^{(i)}-\mu_c)\right)$$

const const

w.r.t. $\mu_1$

$$= \sum_{i=1}^{n} r_{i1} \nabla \left[-\frac{1}{2}\cdot \underbrace{(x^{(i)}-\mu_1)^T \Sigma_1^{-1} (x^{(i)}-\mu_1)}_{\text{quadratic form}}\right]$$

$$\nabla_x x^T A x = 2Ax$$

$$= \cancel{-\frac{1}{2}} \sum_{i=1}^{n} r_{i1} \cdot \cancel{2}\, \Sigma_1 (x^{(i)}-\mu_1) \cdot \cancel{(-1)} = 0$$

$$= \sum_{i=1}^{n} r_{i1}\, \Sigma_1^{-1} (x^{(i)}-\mu_1) = 0$$

$$\Sigma_1 (\Sigma_1^{-1}) \cdot \sum_{i=1}^{n} r_{i1} (x^{(i)}-\mu_1) = \Sigma_1 0$$

$$\sum_{i=1}^{n} r_{i1} (x^{(i)}-\mu_1) = 0$$

$$\sum_{i=1}^{n} r_{i1} x^{(i)} = \mu_1 \sum_{i=1}^{n} r_{i1}$$

$$\Rightarrow \mu_1 = \frac{\sum_{i=1}^{n} r_{i1} x^{(i)}}{\underbrace{\sum_{i=1}^{n} r_{i1}}}$$

weighted average of $x^{(i)}$'s where weights are how likely example is in cluster 1

$= P(\text{example } 1 \text{ in cluster } 1)$
$+ \cdots$
$+ P(\text{example } n \text{ in cluster } 1)$
$= \mathbb{E}$ number of examples in cluster 1

$$\pi_c = \frac{\sum_{i=1}^{n} r_{ic}}{n}$$ "soft version" of counting $\frac{\text{\# points in cluster } c}{\text{total \# points}}$

$$\Sigma_c = \frac{\sum_{i=1}^{n} r_{ic} (x^{(i)} - \mu_c)(x^{(i)} - \mu_c)^T}{\sum_{i=1}^{n} r_{ic}}$$

expectation of $(x - \mu_c)(x - \mu_c)^T$ using $r_{ic}$ as weights

↑

definition of covariance