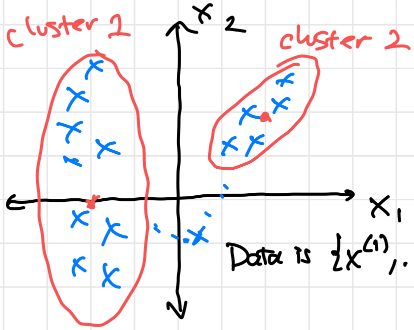
3/23/2023: Gaussian Mixture Models (GMM)

1. What is a 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 is $\{x^{(1)}, \dots, x^{(n)}\}$

How was the data generated?

1. Randomly choose cluster 1 or 2 ← Latent variable
2. Sample from a Gaussian distribution for the chosen cluster

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

$\pi_1 = 2/3$, $\pi_2 = 1/3$ ← π_i = Probability of choosing cluster i in step 1

$\mu_1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$

$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$

Reasonable parameters we want to learn given this dataset

Inference: Infer probability distribution of a $\boxed{\text{latent}}$ random variable

unobserved in the data

(In contrast, learning means fitting parameters)

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

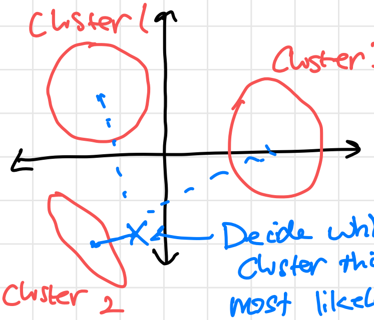
Z_i is latent variable denoting cluster choice $\in \{1, \dots, k\}$

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

Inference problem: Given:

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

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



Decide which cluster this most likely comes from

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

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

① $P(Z_i = c) = \pi_c$

② $P(X_i = x^{(i)} \mid Z_i = c)$

Gaussian w/ mean μ_c, covariance Σ_c

Conditioned on being in cluster c, what is probability 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

Comparison: Univariate Gaussian

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

Announcements

Learning GMMs:

Comparison with k-Means

- Assignments ("hard assignments") $\approx$ Latent variables ("soft assignments")
- Centroids $\approx$ parameters

k-Means: Alternate between updating assignments & updating centroids

Today: Expectation-Maximization (EM)

Generally used when you have both
- Latent variables AND
- Unknown parameters

① E-step: Infer latent variable's distributions given current guess of parameters

$\approx$ make assignments based on current centroids

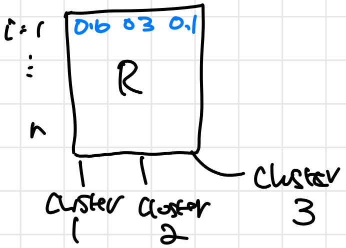
② M-step: Choose best parameters that fit the data based on the inferred distribution of latent variables

$\approx$ choose new centroids based on current assignment

EM for GMMs:

$\boxed{\text{E-step}}$ For each $i = 1, \dots, n$, infer distribution for z_i

Call $r_{ic} = P(Z_i = c \mid X_i = x^{(i)};$ current parameter guess$)$



Comes from inference procedure above

$\boxed{\text{M-step}}$ We have:

Actual value of all the X_i's

Distribution for each Z_i

Can't directly do MLE, but can do something similar

we will maximize Expected Complete Loglikelihood (ECLL)

$$ECLL(\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}) = \underbrace{\sum_{i=1}^{n} \sum_{c=1}^{K} r_{ic}}_{\text{"Expected"}} \log P(\underbrace{X_i = x^{(i)}, Z_i = c}_{\text{"Complete" b/c compute likelihood of both } X_i \text{ \& } Z_i}; \pi, \mu, \Sigma)$$

What choices of $\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}$ maximize ECLL?

Start with μ_1 ($\mu_2, \dots, \mu_K$ are symmetric)

Taking gradient wrt μ_1 & set it $= 0$

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

$$= \sum_{i=1}^{n} r_{i1} \nabla \left[\underbrace{\cancel{\log P(Z_i = 1)}}_{\text{doesn't depend on } \mu_1} + \log P(X_i = x^{(i)} \mid Z_i = 1) \right]$$

$$= \sum_{i=1}^{n} r_{i1} \nabla \log P(X_i = x^{(i)} \mid Z_i = 1)$$

$$= \underbrace{\cancel{\frac{1}{(2\pi)^{d/2}}}}_{\text{Constant}} \cdot \underbrace{\cancel{\frac{1}{\sqrt{\det(\Sigma_c)}}}}_{\text{Constant wrt } \mu_1} \cdot \exp\left(-\frac{1}{2} \cdot (x^{(i)} - \mu_c)^T \Sigma_c^{-1} (x^{(i)} - \mu_c)\right)$$

$$= \sum_{i=1}^{n} r_{i1} \nabla_{\mu_1} \left[-\frac{1}{2} (x^{(i)} - \mu_1)^T \Sigma_1^{-1} (x^{(i)} - \mu_1) \right]$$

$$\boxed{\nabla_x \, x^T A x = 2Ax}$$

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

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

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

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

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

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

☆

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

p(example 1 in cluster 1)
+ p(example 2 in cluster 1)
+ ⋮
" "n in cluster 1

= Expected number of examples in cluster 1

$$\pi_c = \frac{\sum_{i=1}^{n} r_{ic}}{n}$$

Soft version of $\frac{\text{Counting \# points in cluster c}}{\text{total \# points}}$

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

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

This is one definition of covariance