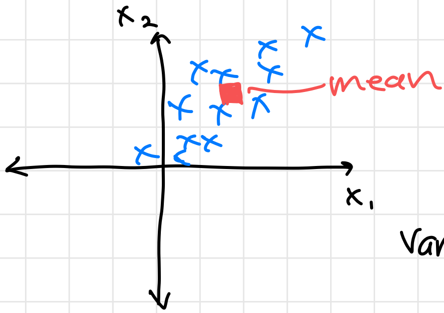
10/31/2023: Gaussian Mixture Models (GMM)

1. How to think of clusters w/ non-spherical shape?
2. What is a GMM?
3. Inference — Assign datapoint to cluster?
4. Learning — Decide shapes of clusters & assign points to clusters at same time



Mean $\mu = \mathbb{E}[X]$

$$= \frac{\sum_{i=1}^{n} x^{(i)}}{n}$$

Variance of $x_1 = \mathbb{E}[(X_1 - \mu_1)^2]$

$$= \frac{1}{n} \sum_{i=1}^{n} (x_1^{(i)} - \mu_1)^2$$

Covariance between X_1 & X_2

$$= \mathbb{E}[(X_1 - \mu_1)(X_2 - \mu_2)] = \frac{1}{n} \sum_{i=1}^{n} (x_1^{(i)} - \mu_1)(x_2^{(i)} - \mu_2)$$

Correlation between X_1 & X_2 $= \dfrac{\text{Covariance}(X_1, X_2)}{\sqrt{Var(X_1)\, Var(X_2)}}$

positive

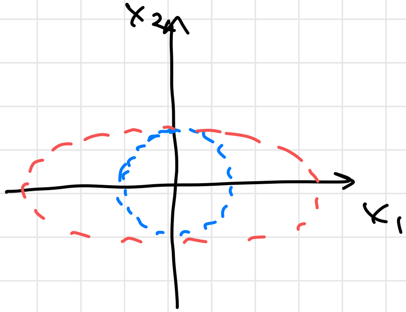
Covariance > 0 $\iff$ positively correlated,

Covariance < 0 $\iff$ negatively correlated

Covariance Matrix $\Sigma = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{pmatrix}$

matrix that summarizes
- variance in every dimension
- Covariance between every two dimensions

captures shape of distribution

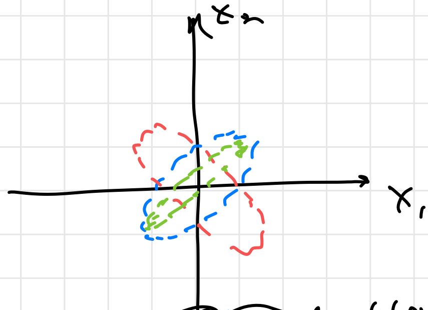


$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$$

large variance for x_1

small variance for x_2



$$\Sigma = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}$$

correlation is 0.5

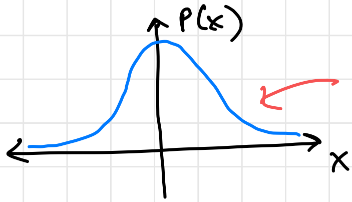
$$\Sigma = \begin{pmatrix} 1 & -1/2 \\ -1/2 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

correlation is 0.9

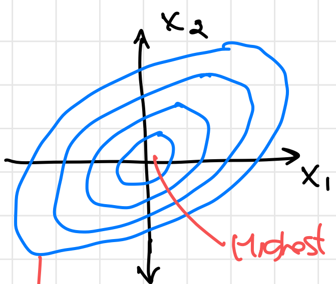
Multivariate Gaussians

In 1 Dimension: Univariate Gaussian distribution



Determined by mean $= \mu$, variance $= \sigma^2$

In d-dimensions: Multivariate Gaussian distribution



Highest density

points with same probability density

Determined by mean $\mu \in \mathbb{R}^d$

and Covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$

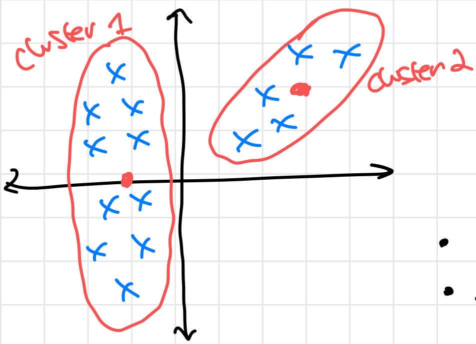
Formula for probability density:

$$P(x; \mu, \Sigma) = \overbrace{\boxed{\frac{1}{(2\pi)^{d/2}} \cdot \frac{1}{\sqrt{\det(\Sigma)}}}}^{\text{constants}} \cdot \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

Compare with 1-D case:

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

What is a GMM?



Goal: given data, produce clusters with custom shapes

How? For each cluster, we will learn:

- π_1, π_2: size of each cluster
- μ_1, μ_2: center of each cluster
- Σ_1, Σ_2: covariance of each cluster

$\Bigg\}$ params of GMM model

For this dataset, we want to learn:

$\pi_1 = \frac{2}{3}, \quad \pi_2 = \frac{1}{3}$

$\mu_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$

$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}, \quad \Sigma_2 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$

How to learn?

- Probabilistic story
- Convert to loss
- Minimize loss

Probabilistic Story of GMMs:

For each $i = 1, \dots, n$: — # of examples in dataset

① Randomly sample cluster Z_i where $P(Z_i = j) = \pi_j$

② Randomly sample example X_i from multivariate Gaussian with mean μ_{Z_i}, covariance Σ_{Z_i}

We only observe that $\underbrace{X_i}_{\text{random variable}} = \underbrace{x^{(i)}}_{\text{observed value}}$ in the dataset

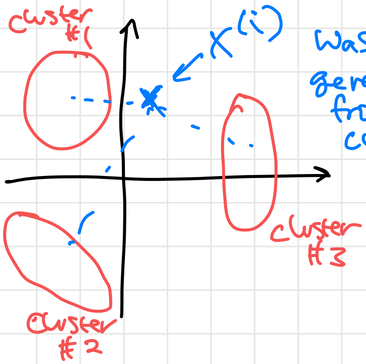
We never directly observe Z_i ("latent variable")

Inference: Inferring a probability distribution of a latent random variable conditioned on observed random variables

For GMMs: Given:

- Observed value $x^{(i)}$ for X_i
- Known parameters $\pi_{1:K}$, $\mu_{1:K}$, $\Sigma_{1:K}$

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



Was $x^{(i)}$ generated from cluster 1, 2, or 3?

How? Bayes Rule

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

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

$P(Z_i = j) = \pi_j$

$P(X_i = x_i^{(i)} \mid Z_i = j)$

= multivariate Gaussian pdf with mean μ_j, covariance Σ_j

Result: for each $x^{(i)}$, we get distribution

$$\underbrace{\begin{aligned} P(Z_i = 1 \mid X_i = x^{(i)}) &= 0.6 \\ P(Z_i = 2 \mid X_i = x^{(i)}) &= 0.1 \\ P(Z_i = 3 \mid X_i = x^{(i)}) &= 0.3 \end{aligned}}_{}$$

"soft assignment" / "soft clustering" b/c it's all probabilistic

produce a "hard" assignment by choosing most likely options

assign to cluster #1

"hard assignment"

Finally: How do we learn $\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}$?

Algorithm: Expectation-Maximization (EM)

Very general method whenever you have:
- Latent variables
- Unknown parameters

Strategy: Alternate between updating each one

① **E-Step**: Infer latent variable distribution using current guess of parameters

← make assignments based on k-means centroids

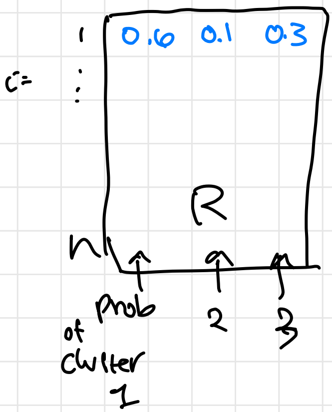
Alternate

② **M-Step**: Chose parameters that best fit the data based on inferred distribution of latent variables

← choose new k-means centroids based on newest assignment

$\boxed{\text{E-step}}$ For each $i = 1, \ldots, n$, we infer

$$R_{ij} = P(Z_i = j \mid X_i = x^{(i)};\ \text{current guess of parameters})$$



Uses the inference procedure from before

$\boxed{\text{M-step}}$ Takes in :

- Actual values of all X_i's
- Inferred distributions of all Z_i's

less info than in supervised learning

Cant do MLE w/o actual values of Z_i's
But something similar:
We can maximize Expected Complete Log-Likelihood (ECLL)

$$ECLL(\pi_{1:K}, \mu_{1:K}, \Sigma_{1:K}) = \underbrace{\sum_{i=1}^{n} \sum_{j=1}^{K} \underbrace{R_{ij}}_{\text{Prob of example } i \text{ in cluster } j}}_{\text{expectation}} \underbrace{\log P(X_i = x^{(i)}, Z_i = j ; \pi, \mu, \Sigma)}_{\text{complete log likelihood (includes } X_i \,\&\, Z_i)}$$