S Saravanan
IPhD20025

(1) Generative Vs Discriminative

(2) Maximum Likelihood Estimation

(3) covariance Matrix - Closed form

Some Examples for GMM - motivation

(4) GMM for 1D

EM for 1D

(5) GMM for $d \geq 2$

EM for $d \geq 2$

References :

(i) NPTEL - IIT madras BS Deg Prog.
MLT Dr Arun Rajkumar

(ii) Ritvik math - GMM (YOUTUBE)

(iii) Machine learning TV - GMM

★ Broad Paradigms of ML

Supervised learning
> Classification
> Regression - logistic
> Decision Tree

Unsupervised learning
> Clustering
  K means
> Representation learning
  ↓
  better way to represent the i/p features

Sequential learning → make a decision, observe feedback based on it make the next decision
> Reinforcement learning

- Generative Model

- $P(x,y)$

given (x^i, y^i) we must be able to associate it with a density / probability func when features are continuous / discrete
If we make assumptions on $p(x,y)$ how should the density look like then we r doing generative modelling
here we are modelling how the features themselves are generated & also how the labels are classified
(How sentences in English are generated)
Cares about how features are also generated
Models feature generation also

- Discriminative Model

- $P(y|x)$

Doesn't model the joint distribution of (x,y) but it only models $P(y|x)$
(P(y given x)

~~to~~ be able to acquire this ability to discriminate b/w English & non English sentences., you don't necessarily need to Know how sentences themselves are generated
How is y given x is generated.
, not how x itself is generated
examples: K means Decision Tree

If you know how to generate well you know how to discriminate well.
∴ If you know how data could have been generated, then gen. model is great.

* Estimation :

"There is some probabilistic mechanism that generates the data"

Goal: Given data find/estimate the distribution

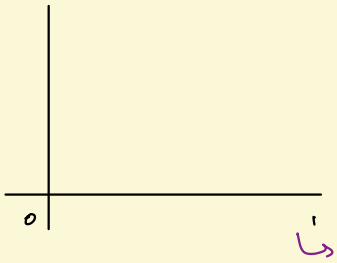
Assumptions: All observations are independent-identically distributed

$P(x_i | x_j) = x_i \quad \forall i, j$

$P(x_i = 1) = P(x_j = 1) = p$ (same coin is tossed)

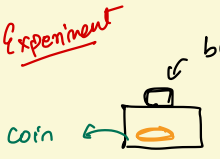
i.i.d: How determining the outcome is same everytime.

* Maximum likelihood Estimation.



↳ All choices of p that could have possibly generated the data that you see

Experiment



Every time you press the button the coin gets flipped. The coin is an unbiased coin with prob. of head occuring to be 'p'. What is the choice of 'p' which generates data like $\{1\ 0\ 1\ 1\}$

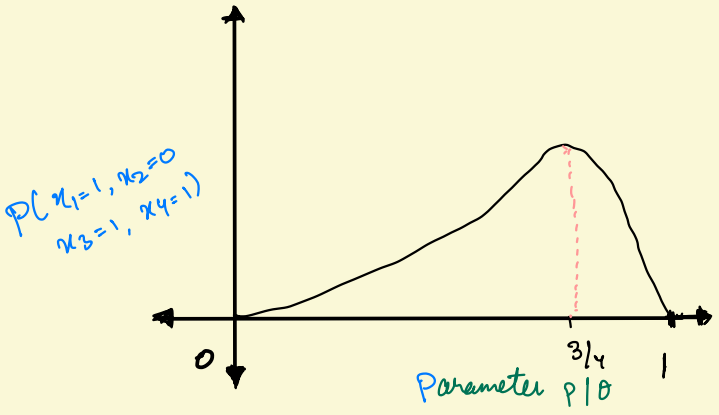
1 - head
0 - tail

Among all choices of p which choice has max. prob. of generating the above observation in 4 tosses?

How can we resolve it?

For each p between 0 & 1 calculate the chance of getting $\{1\ 0\ 1\ 1\}$.

i.e plot



$p = 3/4$ is most likely to generate the o/p

Fisher's Principle of Maximum likelihood.

Likelihood as function

Prob. of $\{x_1, x_2, \ldots x_n\}$ given the parameter (here prob. of head) is p.

$$L(p, \{x_1, x_2, \ldots x_n\}) = P(x_1, x_2, \ldots x_n ; p)$$

$$= P(x_1; p)\, P(x_2; p) \cdots P(x_n; p) \quad \text{(By independence)}$$

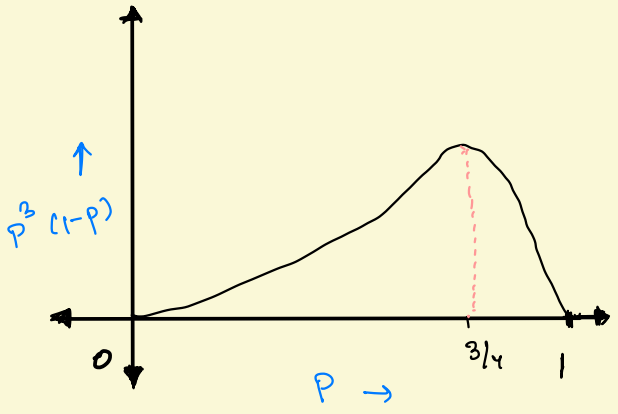
$$= \prod_{i=1}^{n} p^{x_i} (1-p)^{1-x_i}$$

(for this prob)

$$p^{x_i}(1-p)^{x_i} = \begin{cases} p, & x_i = 1 \\ 1-p, & x_i = 0 \end{cases}$$

$$\therefore P(x_1=1, x_2=0, x_3=1, x_4=1) = (p^1)(1-p)\,p\cdot p = p^3(1-p)$$

Given δ the parameter involved by looking @ the observation



$$\hat{p}_{ML} = \arg\max_p h(p, \{x_1, x_2, \ldots x_n\})$$

$$\hat{p}_{ML} = \arg\max_p \prod_{i=1}^{n} p^{x_i}(1-p)^{x_i}$$

↓ easier to look at

$$\hat{p}_{ML} = \arg\max_p \log\left(\prod_{i=1}^{n} p^{x_i}(1-p)^{x_i}\right)$$

(∵ log is monotone ↑)

log likelihood func.

log [L({x1, x2 .. xn}, p)]

$$\hat{p}_{ML} = \arg\max_p \underbrace{\sum_{i=1}^{n} x_i \log p + (1-x_i)\log(1-p)}_{(\log[L(p)])}$$

$$\frac{\partial}{\partial p}(\log[L(p)]) = 0 \Rightarrow \sum_{i=1}^{n} \frac{x_i}{p} - \frac{(1-x_i)}{(1-p)} = 0$$

$$\Rightarrow \sum_{i=1}^{n} \frac{x_i - x_i p - p + x_i p}{p(1-p)} = 0 \Rightarrow \sum_{i=1}^{n}(x_i - p) = 0$$

$$\Rightarrow \sum_{i=1}^{n} x_i - np = 0 \Rightarrow \hat{p}_{ML} = \frac{\sum_{i=1}^{n} x_i}{n}$$

*If Data = $\{x_1, x_2, \ldots, x_n\}$, $x_i \in \mathbb{R}$

Find θ s.t $x_i \sim$ Gaussian (μ, σ^2) $\forall i$

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

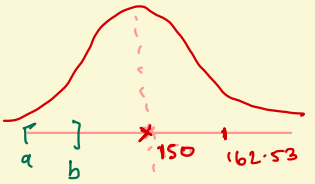
$$L(\mu, \sigma^2, \{x_1, x_2, \ldots, x_n\}) = P(\{x_1, x_2, \ldots, x_n\}, \mu, \sigma^2)$$

$$= \prod_{i=1}^{n} P(x_i, \mu, \sigma^2)$$

Assume σ^2 is known.

Eg: If $\mu = 150$ what is prob. that $x_1 = 162.53$?

$x_1 = 162.53$



for continuous dist. $P(162.53, \mu = 150) = 0$ ∵ it's a point

Only intervals will have non zero values for probability.

$$\therefore P(x_i, \mu, \sigma^2) = 0 \quad \text{for any } x_i$$

So, $h(\mu, \sigma^2, \{x_1, x_2, \ldots x_n\}) = f_{x_1, x_2, \ldots x_n}(x_1, x_2, \ldots x_n, \mu, \sigma^2)$ where f is the pdf

$$= \prod_{i=1}^{n} f_{x_i}(x_i, \mu, \sigma^2)$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\therefore \log(L(\mu, \sigma^2, \{x_1, x_2, \ldots, x_n\})) = \sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi}\,\sigma}\right) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{\partial \log L}{\partial \mu} = -2 \sum_{i=1}^{n} \frac{(x_i-\mu)}{2\sigma^2}(-1) = 0.$$

$$\Rightarrow \sum_{i=1}^{n} x_i - n\mu = 0 \Rightarrow \mu_{ML} = \frac{\sum_{i=1}^{n} x_i}{n}$$

If we assume the data $\{x_1, x_2, \ldots x_n\}$ is generated acc. to a Gaussian with unknown mean μ, the best guess acc. to ML principal is the sample mean

If I change the assumption that underlying dist. is Gaussian instead something else like Laplacian then I get a diff. mean

* Covariance Matrix

variables →

	A	B
Sub1	1	1
Sub2	3	0
Sub3	−1	−1

Link b/w happiness one derives from eating A
happiness they derive by eating B A, B = 2 Features

	A	B
A	cov(A,A)	cov(A,B)
B	cov(B,A)	cov(B,B)

High degree of covariance:
If Both are highly +ive or highly −ive

low degree of cov:
One is very +ive, other is very −ive

cov(A,A) = Variance(A)

cov(B,B) = Variance(B)

Correlation: Bounded b/w −1 & 1

Covariance: Not necessarily bdd b/w −1 & 1

Cov. matrix: Square matrix
Symmetric

$Cov(A,B) = E(AB) - E(A)E(B)$ → expected value of the prod. of the variables − the prod. of exp. value of each of the variables.

$E(B) = 0$ $E(A) = \frac{-1+3+1}{3} = 1$

$cov(A,B) = \frac{2}{3} = cov(B,A)$

	AB
S1	1
S2	0
S3	1

$E(AB) = \frac{1+0+1}{3} = \frac{2}{3}$

$Cov(A,A) = E(A^2) - (E(A))^2$

$= \frac{11}{3} - 1 = 8/3$

A^2
1
9
1

$cov(B,B) = E(B^2) - (E(B))^2$

$\frac{2}{3}$

$$cov = \begin{bmatrix} 8/3 & 2/3 \\ 2/3 & 2/3 \end{bmatrix}$$

★ Closed Form

$X = N \times D$

Samples: $x_1, x_2, \dots, x_N$ in $\mathbb{R}^D$

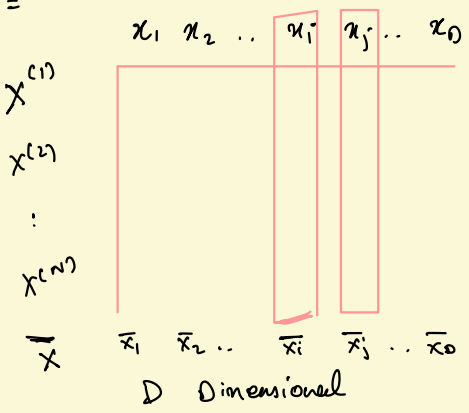
Cov Blw i^{th} & j^{th} variables $\equiv$ Cov. blw i^{th} fruit & j^{th} fruit, the happiness index of those fruits

Cov Blw p^{th} & j^{th} variable:

$$\text{Cov}(x_i, x_j) = \frac{1}{N} \sum_{k=1}^{N} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

x_{ki} = k^{th} sample i^{th} feature

$X =$



Cov Matrix:

$$S = \begin{bmatrix} \frac{1}{N}\sum_{k=1}^{N}(x_{ki}-\bar{x}_i)(x_{ki}-\bar{x}_i) & \frac{1}{N}\sum_{k=1}^{N}(x_{k1}-\bar{x}_1)(x_{k2}-\bar{x}_2) & \cdots & \frac{1}{N}\sum_{k=1}^{N}(x_{k1}-\bar{x}_1)(x_{kD}-\bar{x}_D) \\ \frac{1}{N}\sum_{k=1}^{N}(x_{k2}-\bar{x}_2)(x_{k1}-\bar{x}_1) & \frac{1}{N}\sum_{k=1}^{N}(x_{k2}-\bar{x}_2)(x_{k2}-\bar{x}_2) & \cdots & \frac{1}{N}\sum_{k=1}^{N}(x_{k2}-\bar{x}_2)(x_{kD}-\bar{x}_D) \\ \vdots & \vdots & & \vdots \\ \frac{1}{N}\sum_{k=1}^{N}(x_{kD}-\bar{x}_D)(x_{k1}-\bar{x}_1) & \frac{1}{N}\sum_{k=1}^{N}(x_{kD}-\bar{x}_D)(x_{k2}-\bar{x}_2) & \cdots & \frac{1}{N}\sum_{k=1}^{N}(x_{kD}-\bar{x}_D)(x_{kD}-\bar{x}_D) \end{bmatrix}$$

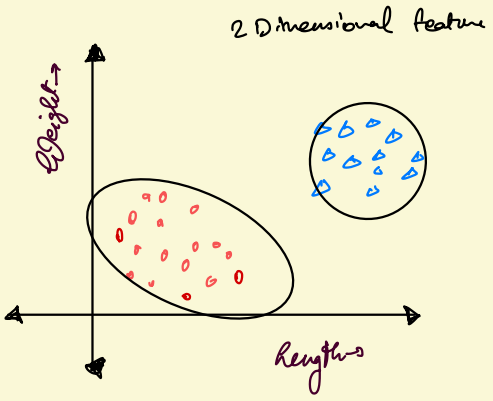
$$S = \frac{1}{N} \underbrace{\begin{bmatrix} x_{11}-\bar{x}_1 & x_{21}-\bar{x}_1 & \cdots & x_{N1}-\bar{x}_1 \\ x_{12}-\bar{x}_2 & x_{22}-\bar{x}_2 & \cdots & x_{N2}-\bar{x}_2 \\ \vdots & \vdots & & \vdots \\ x_{1D}-\bar{x}_D & x_{2D}-\bar{x}_D & \cdots & x_{ND}-\bar{x}_D \end{bmatrix}}_{D \times N} \underbrace{\begin{bmatrix} x_{11}-\bar{x}_1 & x_{12}-\bar{x}_2 & \cdots & x_{1D}-\bar{x}_D \\ x_{21}-\bar{x}_2 & x_{22}-\bar{x}_2 & \cdots & x_{2D}-\bar{x}_D \\ \vdots & \vdots & & \vdots \\ x_{N1}-\bar{x}_1 & x_{N2}-\bar{x}_2 & \cdots & x_{ND}-\bar{x}_D \end{bmatrix}}_{N \times D}$$

Python

$$S = \frac{1}{N}(X-\bar{X})^T(X-\bar{X}) = \frac{1}{N}\,\text{np.dot}((X-\bar{X}).T,\ X-\bar{X})$$

An Example.

★ Gaussian Mixture

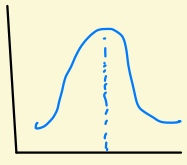


Based on weight & length classify the fish as Tuna or Salmon

GMM ⇒ all classes are distributed in Gaussian Distribution.

o = Tuna
o = Salmon

One dimension
Normal



For higher dimension

For each of class (Tuna / Salmon) the mean is μ & instead of variance we have covariance Σ (this describes the interaction b/o weight & length & hence decides the shape - either circular / ellipti etc, tilted or straight)

π = prob. for each of the class to occur.

Goal: GMM tries to figure out optimal [μ, Σ, π] for each of the class

If x = tuna then it's distributed according to μ_T & Σ_T

Tuna Distribution $N(x \mid \mu_T, \Sigma_T)$

Salmon Distribution. $N(x \mid \mu_S, \Sigma_S)$

$P(\text{TUNA}) = \pi_T$

$P(\text{Salmon}) = \pi_S$

$x = \begin{pmatrix} l_i \\ w_i \end{pmatrix}$ $\quad P(x) = \pi_T N(x \mid \mu_T, \Sigma_T) + \pi_S N(x \mid \mu_S, \Sigma_S)$ 2 classes

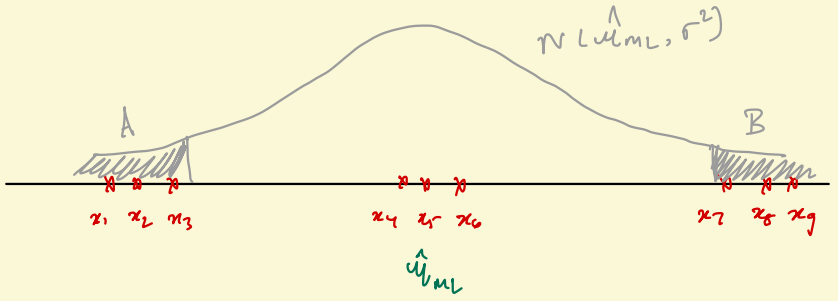
x = training Example

Bayes

Q How do we pick the best π, μ, Σ?

We want to maximize $P(x \mid \pi, \Sigma, \mu)$ seeing the data x given the parameters μ, π, Σ

Gaussian Mixture Model



What could be a generative story?

Is Gaussian a good model for the data with mean $\hat{\mu}_{ML} = \frac{\sum x_i}{9}$

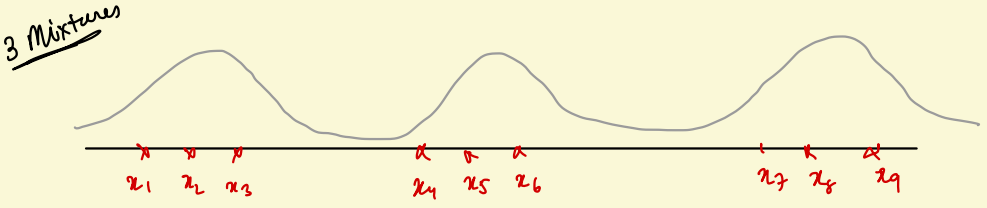
The problem is, this Gaussian with $\hat{\mu}_{ML}$ has very less prob. in the regions A & B But there are data points in the region.

∴ Problem is we have assumed model is Gaussian.

We want a density like ↓ to explain the data points.

We require a new Generative model Mixture of Gaussians.

cluster the data into several parts & assume that each of the cluster forms a Gaussian



Why Gaussian?

- most of obs. follow Gaussian
- easy to do math. manipulations eg- diff.

* Univariate

$$G(x \mid \mu, \sigma) = \frac{1}{(\sqrt{2\pi})\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

* Multivariate

$$N(x \mid \mu, \Sigma) = \frac{1}{(\sqrt{2\pi})(|\Sigma|^{1/2})} e^{\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]}$$

Step 1 Generate a mixture component among $\{1, 2, \ldots K\}$ call z_i, (z_i a number in $1, 2, \ldots K$) $1 \le i \le n$

For each x_i assign a mixture z_i according to probabilities

$P(z_i = l) = \pi_l$ ($\sum_{1}^{K} \pi_i = 1$ $0 \le \pi_i \le 1$ $\forall i$) Pick the one with highest prob.

Step 2 Generate $x_i \sim N(\mu_{z_i}, \sigma^2_{z_i})$

$\{x_1, x_2, \ldots x_N\}$ = observed

$\{z_1, z_2, \ldots z_n\}$ = unobserved / latent variable

$\Pi = [\pi_1, \pi_2, \ldots, \pi_K]$ i.e (μ_K, σ_K^2)

Total: $2K + K - 1 = 3K - 1$

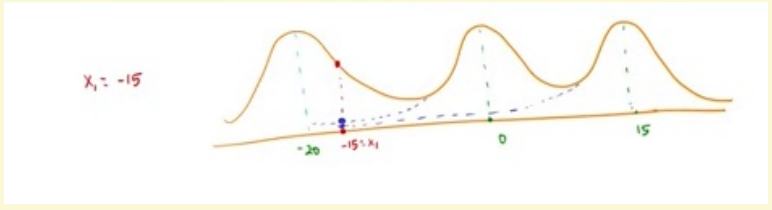
Determine/ Estimate $3K-1$ parameters

★ Likelihood function for GMM

$$L\left(\begin{matrix}\mu_1,\dots,\mu_K\\ \sigma_1^2,\dots,\sigma_K^2\\ \pi_1,\dots\pi_K\end{matrix}, x_1, x_2, \dots x_n\right) = \prod_{i=1}^{n} f\left(x_i;\ \begin{matrix}\mu_1,\mu_2,\dots,\mu_K\\ \sigma_1,\sigma_2,\dots\sigma_K^2\\ \pi_1,\pi_2,\dots\pi_K\end{matrix}\right)$$

→ Product since $x_1, x_2, \dots x_n$ are independ & iid.

$$= \prod_{i=1}^{n}\left(\sum_{k=1}^{K}\pi_k\, f(x_i, \mu_k, \sigma_k^2)\right)$$



For one dimension

$$L(\theta) = \prod_{i=1}^{n}\left(\sum_{k=1}^{K}\pi_k \frac{e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k}\right)$$

log likelihood

$$\log(L(\theta)) = \sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\pi_k \frac{e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k}\right)$$

Not possible to solve this analytically hence we cannot arrive at a closed form or explicit expression for π_k or μ_k or σ_k

And also constraints are involved.

It's a maximization problem.

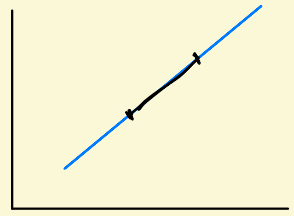
Need an alternate way to solve the above

★ Convex functions and Jensen's Inequality

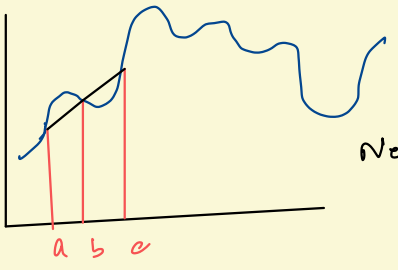
Convex functions $x, y \in D$

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y), \quad \theta \in [0,1]$$

Concave func. $$f(\theta x + (1-\theta)y) \ge \theta f(x) + (1-\theta) f(y), \quad \theta \in [0,1]$$



Both convex & concave



Neither concave nor convex

★ Jensen's Inequality: $f\left(\sum_{k=1}^{K}\lambda_k a_k\right) \ge \sum_{k=1}^{K}\lambda_k f(a_k)$ with $\sum_{k=1}^{K}\lambda_k = 1$

- log is a concave function.

π_k = prob. of each cluster k

Recall $\log(L(\theta)) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K}\left(\pi_k e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}\right)\right)$

For every data point i introduce the parameters $\{\lambda_1^i, \lambda_2^i, \dots \lambda_k^i\}$ s.t $\forall i \sum_{k=1}^{K} \lambda_k^i = 1$

$0 \le \lambda_k^i \le 1 \quad \forall i, k$

$$\log(h(\theta)) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \lambda_k^i \left(\frac{\pi_k e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\sigma_k \lambda_k^i}\right)\right)$$

By Jensen's for each i

$$\log\left(\sum_{k=1}^{K} \lambda_k^i \frac{(\pi_k e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}})}{\sqrt{2\pi}\sigma_k \lambda_k^i}\right) \ge \sum_{k=1}^{K} \lambda_k^i \log\left(\frac{\pi_k e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\sigma_k \lambda_k^i}\right)$$

By Jensen's inequality,

$$\log(h(\theta)) \ge \sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k^i \log\left(\frac{\pi_k e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\sigma_k \lambda_k^i}\right)$$

↳ modified log likelihood obtained by Jensen's

↳ gives a lower bound for the true log likelihood @ θ.

for any choice of $\lambda = \begin{Bmatrix} \lambda_1^1, \dots, \lambda_k^1 \\ \lambda_1^2, \dots, \lambda_k^2 \\ \vdots \\ \lambda_1^n, \dots, \lambda_k^n \end{Bmatrix}$ we are just getting a lower Bound for our likelihood.

But what are we gaining?

Key insight: If we fix some lambda, it's easy to maximize wrt θ

If we fix some θ, it's easy to maximize wrt λ.

★ Fix λ and maximize over θ

$$l = \underset{\theta}{\text{maximize}} \sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k^i \log \pi_k - \lambda_k^i \frac{(x_i-\mu_k)^2}{2\sigma_k^2} - \lambda_k^i \log(\sqrt{2\pi}\sigma_k) - \lambda_k^i \log(\lambda_k^i)$$

① $\frac{\partial l}{\partial \mu_k} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_k^i \frac{(x_i-\mu_k)}{\sigma_k^2} = 0 \Rightarrow$

$$\hat{\mu}_k^{ML} = \frac{\sum_{i=1}^{n} \lambda_k^i x_i}{\sum_{i=1}^{n} \lambda_k^i}$$

(2) $\frac{\partial l}{\partial \sigma_k} = 0 \Rightarrow \sum_{i=1}^{n} \left(\frac{\lambda_k^i (x_i - \mu_k)^2}{\sigma_k^3} - \frac{\lambda_k^i}{\sqrt{2\pi}\sigma_k} \cdot \sqrt{2\pi} \right) = 0$

$\Rightarrow \sum_{i=1}^{n} \lambda_k^i \left((x_i - \mu_k)^2 - \sigma_k^2 \right) = 0 \Rightarrow$

$\hat{\sigma}_k^{ML} = \frac{\sum_{i=1}^{n} \lambda_k^i (x_i - \mu_k)^2}{\sum_{i=1}^{n} \lambda_k^i}$

λ_k^i = prob. that i^{th} point goes to the k^{th} cluster

What you think is the prob.

(8) $\max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log \pi_k$ s.t $\sum_{k=1}^{K} \pi_k = 1$, $\pi_k \geq 0$

can solve this using the method of lagrange multipliers γ = multiplier

$L^* = \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log(\pi_k) - \gamma \left(\sum_{k=1}^{K} \pi_k - 1 \right)$ Find γ^*, π_k^*

(1) $\frac{\partial L^*}{\partial \pi_k^*} = 0 \Rightarrow \sum_{i=1}^{n} \frac{\lambda_k^i}{\pi_k^*} - \gamma^* = 0 \Rightarrow \gamma^* = \frac{\sum_{i=1}^{n} \lambda_k^i}{(\pi_k^*)}$

$\therefore \pi_k^* = \left(\frac{\sum_{i=1}^{n} \lambda_k^i}{\gamma^*} \right)$

(2) Substituting in the constraint, $\sum_{k=1}^{K} \pi_k^* = 1 \Rightarrow \sum_{k=1}^{K} \sum_{i=1}^{n} \frac{(\lambda_k^i)}{(\gamma^*)} = 1$

$\therefore (\gamma^*)^{-} = \sum_{k=1}^{K} \sum_{i=1}^{n} (\lambda_k^i)$ $\quad \forall i, \sum_{k=1}^{K} (\lambda_k)^i = 1$

$\therefore \gamma^* = n$

$\hat{\pi}_k^{ML} = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}$

★ Now fix θ & maximize λ

modified likelihood function:

$$\max_{\lambda_k^i} \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log\left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k \, \lambda_k^i} \right)$$

$$\max_{\lambda_k^i} \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \log\left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k} \right) - \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k^i \ln \lambda_k^i$$

For any i,

$$\max_{\lambda_1^i, \lambda_2^i \cdots \lambda_K^i} \sum_{k=1}^{K} \lambda_k^i \log\left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k} \right) - \sum_{k=1}^{K} \lambda_k^i \log(\lambda_k^i)$$

such that $\sum_{k=1}^{K} \lambda_k^i = 1$, $0 \le \lambda_k^i \le 1$

Lagrange multiplier,

$$L^* = \sum_{k=1}^{K} \lambda_k^i \log\left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k} \right) - \sum_{k=1}^{K} \lambda_k^i \log(\lambda_k^i) + \alpha\left(\sum_{k=1}^{K} \lambda_k^i - 1 \right)$$

① $$\frac{\partial L^*}{\partial \lambda_k^i} = \log\left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k} \right) - \log(\lambda_k^{i\,*}) - 1 + \alpha^* = 0$$

$$\Rightarrow \frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k \, (e^{\alpha^* - 1})} = \lambda_k^{i\,*}$$

② $$\sum_{k=1}^{K} \lambda_k^{i\,*} = 1 \Rightarrow \sum_{k=1}^{K} \left(\frac{\pi_k \, e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}}}{\sqrt{2\pi}\,\sigma_k} \right) = e^{\alpha^* - 1}$$

hence,

$$\lambda_k^{i\,*} = \frac{\pi_k \left(\frac{1}{\sqrt{2\pi}\,\sigma_k} e^{-\frac{(x - \mu_k)^2}{2\sigma_k^2}} \right)}{\sum_{l=1}^{K} \frac{\pi_l}{\sqrt{2\pi}\,\sigma_l} e^{-\frac{(x_i - \mu_l)^2}{2\sigma_l^2}}}$$

$$\hat{\gamma}_k^i = P(z_i = k \mid x_i)$$

★ Algorithm :

① Initialize $\theta^0 = \{ \mu_1^0, \ldots \mu_K^0 \; ; \; \sigma_1^2, \ldots \sigma_K^2 \; ; \; \pi_1^0, \ldots \pi_K^0 \}$

② until convergence $\| \theta^{t+1} - \theta^t \| \leq \varepsilon$ ← Tolerance

$\lambda^{t+1} = \arg\max_{\lambda}$ modified $\log L(\theta^t, \lambda)$ → Expectation Step

$\theta^{t+1} = \arg\max_{\theta}$ modified $\log L(\theta, \lambda^{t+1})$ → Maximization Step

end.

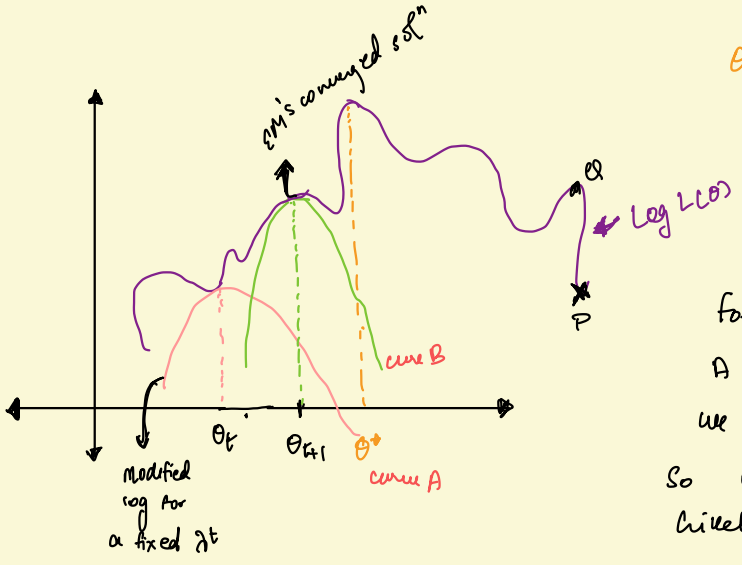
① K Means Algorithm. : Step 1 → compute means
Step 2 → Reassignment

② EM produces soft clustering.

λ_k^i = probability that i^{th} datapoint goes to cluster k

EM takes variance into account

But are we solving for the original problem? We are maximizing only the modified log function



θ^* is the actual value that we wish to get.

But the θ_t that we get from any choice of λ solving the modified log likelihood is always a lower bound for θ^*.

For a fixed λ^t, we find the max θ^t for curve A with that θ_t we find λ^{t+1}. For λ_{t+1} we find θ_{t+1} of curve B

So EM converges to a local-maximum of log likelihood.

So if I start with an initialization @ P I can only reach the local maxima @ Q which is not close to the actual maximum θ^*

So given the data run K means clustering, get the hard clusters. Once we have the clusters compute sample mean and sample variance for each cluster which will give the μ_k & σ_k^2 for each cluster k & to get the π_k's look at the fraction of data points in each of the clusters.

$\therefore$ θ^0 is got from K means.

* EM Algorithm.

Run K means on the data & Find θ^0 from K means algorithm

(1) Initialize $\theta^0 = \begin{cases} \mu_1^0, \dots \mu_K^0 \\ \sigma_1^2, \dots \sigma_K^2 \\ \pi_1^0, \dots \pi_K^0 \end{cases}$

(2) until convergence $\| \theta^{t+1} - \theta^t \| \leq \varepsilon$ → Tolerance

$\lambda^{t+1} = \arg\max_{\lambda}$ modified $\log L(\theta^t, \lambda)$ → Expectation Step

$\theta^{t+1} = \arg\max_{\theta}$ modified $\log L(\theta, \lambda^{t+1})$ → Maximization Step

end.

As the no. of clusters increase ($K \uparrow$) GMM can approximate almost any distribution in $\mathbb{R}^n$

(Sir's Slides
$\pi_k = \omega_k$
$\lambda_k^i = \xi_k^i$)

* Derivation for case where dim ≥ 2

Instead of variance we have the covariance matrix.

For each cluster $k \in \{1, 2, \dots K\}$ we have $\{\mu_k, \pi_k, \Sigma_k\}$
(μ_k: d-dimensional vector; Σ_k: $d \times d$)

$$L\left(\begin{matrix}\mu_1, \dots, \mu_K \\ \Sigma_1, \dots, \Sigma_K \\ \pi_1, \dots \pi_K\end{matrix}, x_1, x_2, \dots x_n\right) = \prod_{i=1}^{n} f\left(x_i; \begin{matrix}\mu_1, \mu_2, \dots, \mu_K \\ \Sigma_1, \Sigma_2, \dots \Sigma_K \\ \pi_1, \pi_2, \dots \pi_K\end{matrix}\right)$$

→ Product since $x_1, x_2, \dots x_n$ are indepen & i.i.d.

$$= \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \pi_k f(x_i, \mu_k, \Sigma_k) \right)$$

$$= \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \frac{\pi_k}{(\sqrt{2\pi})(|\Sigma_k|^{1/2})} e^{\left[-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)\right]} \right)$$

$$\log(L(\theta, x)) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \frac{\pi_k \, e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{(\sqrt{2\pi})(|\Sigma_k|)^{1/2}} \right)$$

Introduce for each i $\sum_{k=1}^{K} \lambda_k^i = 1$

Latent variable
λ_k^i = prob. of sample i occurring in cluster k

$$\log(L(\theta, x)) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \lambda_i^k \left(\frac{\pi_k \, e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{(\sqrt{2\pi})(|\Sigma_k|)^{1/2} \, \lambda_i^k} \right) \right)$$

log is a concave func. Jensen's Ineq ⇒

$$\log(L(\theta, x)) \geq \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_i^k \log\left(\frac{\pi_k \, e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{(\sqrt{2\pi})(|\Sigma_k|)^{1/2} \, \lambda_i^k} \right)$$

$$\log(L(\theta, x)) \geq \sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_i^k \left((\log \pi_k) - \left(\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) - \frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(|\Sigma_k|) - \ln(\lambda_i^k) \right) \right)$$

modified log function

(A) Maximize modified log function Q given λ_k^i $\forall i$

① $\frac{\partial Q}{\partial \mu_k} = 0 \Rightarrow \sum_{i=1}^{n} (x_i - \mu_k) \Sigma_k^{-1} \lambda_i^k = 0 \quad (\Sigma_k \neq 0)$

$$\boxed{\mu_k = \frac{\sum_{i=1}^{n} (x_i) \lambda_i^k}{\sum_{i=1}^{n} \lambda_i^k}}$$

$\frac{\partial}{\partial A}(x^T A^{-1} y) =$

$-(A^T)^{-1} x y^T (A^T)^{-1}$

$\frac{\partial}{\partial A}(\ln |A|) =$

$(A^{-1})^T = (A^T)^{-1}$

② $\frac{\partial Q}{\partial \Sigma_k} = 0 \Rightarrow \sum_{i=1}^{n} \frac{1}{2} (\Sigma_k^T)^{-1} (x_i - \mu_k)(x_i - \mu_k)^T (\Sigma_k^T)^{-1} \lambda_i^k - \frac{1}{2} \sum_{i=1}^{n} (\Sigma_k^T)^{-1} \lambda_i^k = 0$

Post mul. by (Σ_k^T)

$\Rightarrow \frac{(\Sigma_k^T)^{-1}}{2} \left(\sum_{i=1}^{n} (x_i - \mu_k)(x_i - \mu_k)^T \lambda_i^k - \sum_{i=1}^{n} \lambda_i^k (\Sigma_k^T) \right) = 0$

$\Rightarrow$

$$\boxed{\Sigma_k^T = \frac{\sum_{i=1}^{n} (x_i - \mu_k)(x_i - \mu_k)^T \lambda_i^k}{\sum_{i=1}^{n} \lambda_i^k}}$$

③ Q with constraint $\sum_{k=1}^{K} \pi_k = 1$ $\alpha \in \mathbb{R}$ is a Lag multip.

$$Q^* = Q + \alpha \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

(i) $\frac{\partial Q^*}{\partial \pi_k} = 0 \Rightarrow \sum_{i=1}^{n} \frac{\lambda_k^i}{\pi_k^*} - \alpha^* = 0$

$$\Rightarrow \pi_k^* = \frac{\left(\sum_{i=1}^{n} \lambda_k^i \right)}{\alpha^*}$$

(ii) $\sum_{k=1}^{K} \pi_k^* = 1$

$$\Rightarrow \sum_{k=1}^{K} \sum_{i=1}^{n} \frac{\lambda_k^i}{\alpha^*} = 1 \qquad \text{Now for each } i, \quad \sum_{k=1}^{K} \lambda_k^i = 1$$

$$\sum_{i=1}^{n} \frac{1}{\alpha^*} = 1 \Rightarrow \alpha^* = \sum_{i=1}^{n} 1 = n$$

$\therefore$ we get,

$$\boxed{\pi_k^* = \frac{\sum_{i=1}^{n} \lambda_k^i}{n}}$$

(B) Maximize λ_k^i given parameters (μ_k, Σ_k, π_k)

Q with $\sum_{k=1}^{K} \lambda_k^i = 1$ for each i $\quad \alpha \in \mathbb{R}$ is a lag multiplier

We can maximize for each i seperately.

for each i,

$$Q^* = Q + \alpha \left(\sum_{k=1}^{K} \lambda_k^i - 1 \right)$$

(i) $$\left. \frac{\partial Q^*}{\partial \lambda_k^i} \right|_{\lambda_k^{i*}, \alpha^*} = 0 \Rightarrow \frac{\partial}{\partial \lambda_k^i} \Big(\lambda_k^i \big[(\log \pi_k) - \big(\tfrac{1}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) - \tfrac{1}{2} \ln(2\pi) - \tfrac{1}{2} \ln(|\Sigma_k|) - \ln(\lambda_k^i) \big) \big] + \frac{\partial}{\partial \lambda_k^i} \big[\alpha \big(\sum_{i=1}^{K} \lambda_k^i - 1 \big) \big] = 0$$

$$\Rightarrow \log\left(\frac{\pi_k^* e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{\sqrt{2\pi}\,|\Sigma_k|} \right) + \frac{\partial}{\partial \lambda_k^i} \big[\lambda_k^i \log(\lambda_k^i) \big] + \alpha^* = 0$$

$$\Rightarrow \log\left(\frac{\pi_k^* e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{\sqrt{2\pi}\,|\Sigma_k|} \right) - \log \lambda_k^{i*} - 1 + \alpha^* = 0 \Rightarrow$$

$$\lambda_k^{i*} = \frac{\pi_k^* e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{\sqrt{2\pi}\,|\Sigma_k| \,(e^{\alpha^* - 1})}$$

(ii) $$\sum_{k=1}^{K} \lambda_k^i = 1 \Rightarrow \sum_{k=1}^{K} \frac{\pi_k^* e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{\sqrt{2\pi}\,|\Sigma_k| \,(e^{\alpha^* - 1})} = 1$$

$$e^{d^*} - 1 = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{K} \pi_k \frac{e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{|\Sigma_k|}$$

Hence

$$\gamma_k^i = \frac{\frac{1}{|\Sigma_k|} \pi_k \, e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}{\sum_{k=1}^{K} \frac{1}{|\Sigma_k|} \pi_k \, e^{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)}}$$

EM Algorithm for GMMs

given a training set as $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

EM algorithm for GMMs

initialize $\{w_m^{(0)}, \mu_m^{(0)}, \Sigma_m^{(0)}\}$, set $n = 0$

while not converged **do**

E-step: for all $m = 1, \cdots, M$ and $i = 1, \cdots, N$:

Given $\{w_m^{(n)}, \mu_m^{(n)}, \Sigma_m^{(n)}\} \cup \{\mathbf{x}_i\} \longrightarrow \{\xi_m^{(n)}(\mathbf{x}_i)\}$ find — Expectation

M-step: for all $m = 1, \cdots, M$:

Given $\{\xi_m^{(n)}(\mathbf{x}_i)\} \cup \{\mathbf{x}_i\} \longrightarrow \{w_m^{(n+1)}, \mu_m^{(n+1)}, \Sigma_m^{(n+1)}\}$ find — Maximization

$n = n + 1$

end while

- k-means only considers the mean to update the centroid while GMMs takes into account the mean as well as the variance of the data!

$w_m = \pi_m$

$\xi_m^n = \gamma_k^i$