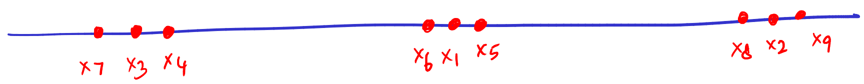
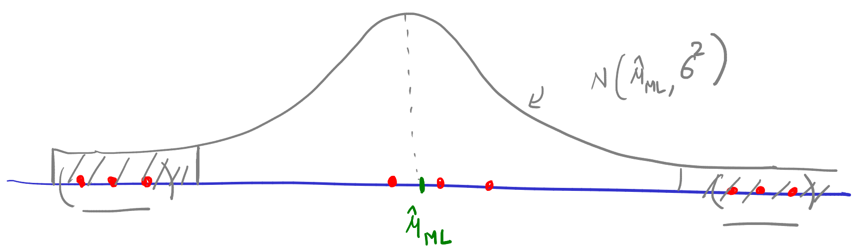
Unsupervised learning

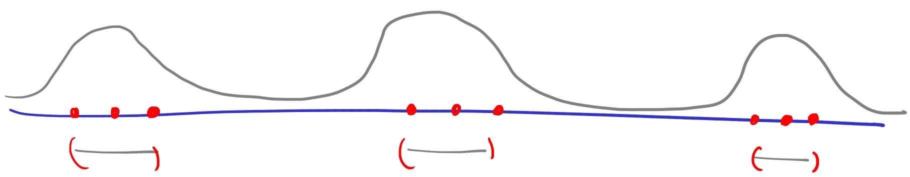
- Rep. learning — PCA / kernel PCA
- Clustering — Llyod's / k-means
- Estimation
  - Max. likelihood
  - Bayesian Modeling.

Today : Estimation for. Slightly complicated data



- What could be a good "generative" story ?





Want a density like above to explain this data.

A NEW GENERATIVE MODEL

MIXTURE OF GAUSSIANS

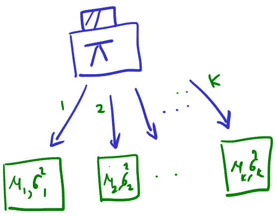
STEP 1: Pick which mixture a data point comes from.

STEP 2: Generate data point from that mixture.

STEP 1: Generate a mixture component among $\{1, \dots, k\}$ $\quad z_i \in \{1, \dots, k\}$

$$P(z_i = \ell) = \pi_\ell \qquad \left[\sum_{i=1}^{K} \pi_i = 1 \quad 0 \le \pi_i \le 1 \;\; \forall i \right]$$

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



$\{x_1, \dots, x_n\}$ -> OBSERVED

$\{z_1, \dots, z_n\}$ -> UNOBSERVED/LATENT.

Latent variable Models

Parameters: $\pi = [\pi_1 \; \pi_2 \dots \pi_K]$ $\qquad$ Total: $2K + K-1$

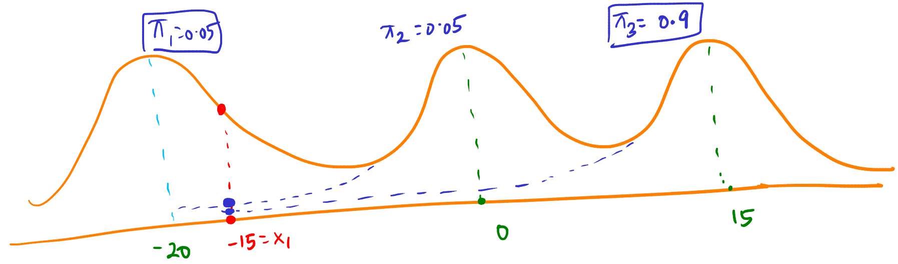
$\forall k \; (\mu_k, \sigma_k^2)$ $\qquad$ $3K-1$

Max. Likelihood for GMM

$$L\Big(\underbrace{\begin{matrix}\mu_1, \dots, \mu_K \\ \sigma_1^2, \dots, \sigma_K^2, \\ \pi_1, \dots, \pi_K\end{matrix}}\; x_1, \dots, x_n\Big) = \prod_{i=1}^{n} f_{mix}\left(x_i \;;\; \begin{matrix}\mu_1, \dots, \mu_K \\ \sigma_1^2, \dots, \sigma_K^2 \\ \pi_1, \dots, \pi_K\end{matrix}\right)$$

$$= \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \cdot \underbrace{f(x_i \,;\, \mu_k, \sigma_k^2)}_{\text{NORMAL/Gaussian Density.}} \right]$$

$x_1 = -15$



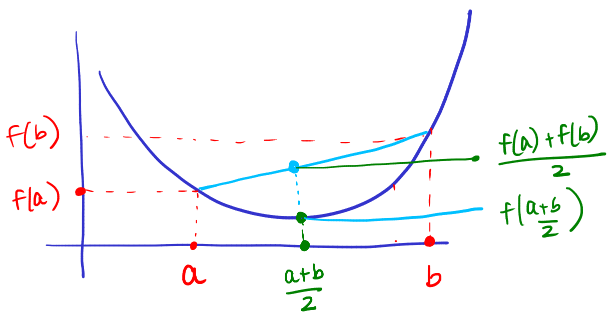
$$L(\theta) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \frac{e^{-(x_i - \mu_k)^2 / 2\sigma_k^2}}{\sqrt{2\pi}\,\sigma_k} \right]$$

↑
all parameters

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

- Not possible to solve this analytically.
- Need an alternate way to solve this efficiently!

Quick detour - Convex functions

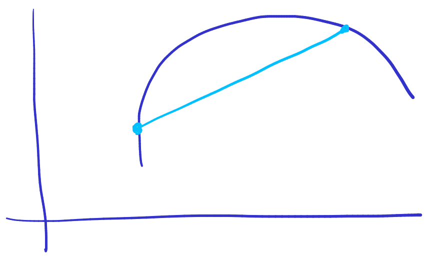


$\forall a, b$

$$f\left(\frac{a+b}{2}\right) \le \frac{f(a)+f(b)}{2}$$

⇓

CONVEX FUNCTION

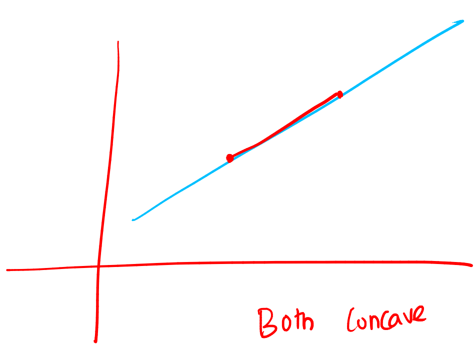


$\forall a, b$

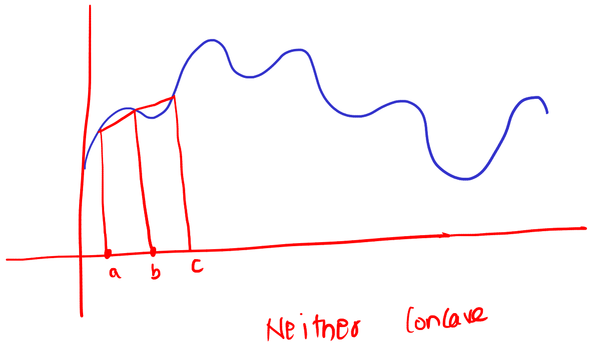
$$f\left(\frac{a+b}{2}\right) \ge \frac{f(a)+f(b)}{2}$$

⇓

CONCAVE FUNCTION

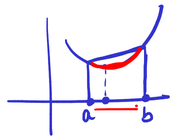


Both concave and convex.



Neither concave nor convex.

$$f\left(\frac{1}{2}\cdot a + \frac{1}{2} b\right) \leq \frac{1}{2} f(a) + \frac{1}{2} f(b)$$



$$\Rightarrow \quad f(\lambda a + (1-\lambda) b) \leq \lambda f(a) + (1-\lambda) f(b)$$

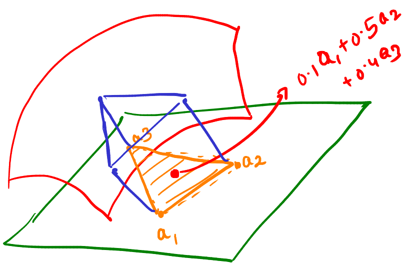
$\lambda \in [0,1]$

For concave

$$f(\lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_k a_k) \geq \lambda_1 f(a_1) + \dots + \lambda_k f(a_k)$$

$\left[\sum_{i=1}^{K} \lambda_i = 1 \quad 0 \leq \lambda_i \leq 1 \right.$

JENSEN'S INEQUALITY.



$$\boxed{f\left(\sum_{k=1}^{K} \lambda_k a_k\right) \geq \sum_{k=1}^{K} \lambda_k f(a_k)}$$

- Log is a concave function! [Why? Exercise]

- How can we exploit Jensen's for performing maximum likelihood.

Recall

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

- INTRODUCE for every data point i, the parameters

$$\{\lambda_1^i, \ldots, \lambda_K^i\} \quad \text{s.t} \quad \forall i \quad \sum_{k=1}^{K} \lambda_k^i = 1, \quad 0 \le \lambda_k^i \le 1 \ \forall k$$

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

By Jensen's

$$\log L(\theta) \ge \text{modified\_}\log L(\theta, \lambda)$$

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

- Note that the above modified log likelihood gives a lower bound for the true log likelihood at θ for any choice of λ

$$\theta \rightarrow \left\{ \begin{matrix} \mu_1, \ldots, \mu_K \\ \sigma_1^2, \ldots, \sigma_K^2 \\ \pi_1, \ldots, \pi_K \end{matrix} \right.$$

$$\lambda \rightarrow \left\{ \begin{matrix} \lambda_1^1, \ldots, \lambda_K^1 \\ \lambda_1^2, \ldots, \lambda_K^2 \\ \vdots \\ \lambda_1^n, \ldots, \lambda_K^n \end{matrix} \right\}$$

- But what are we gaining?

Key insight:

- If we fix λ, it is easy to maximize w.r.t θ
- If we fix θ, it is easy to maximize w.r.t λ.

Fix λ and maximize over θ

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

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

Take derivative w.r.t μ, σ to get

$$\hat{\mu}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i x_i}{\sum_{i=1}^{n} \lambda_k^i} \qquad \hat{\sigma}_k^{2\,MML} = \frac{\sum_{i=1}^{n} \lambda_k^i (x_i - \hat{\mu}_k^{MML})^2}{\sum_{i=1}^{n} \lambda_k^i}$$

$$\max_{\pi_1, \dots, \pi_K} \sum_{i=1}^{n} \left(\sum_{k=1}^{K} \lambda_k^i \log \pi_k \right)$$

$$\text{s.t} \quad \sum_k \pi_k = 1 \; ; \; \pi_k \geq 0$$

Can solve using method of Lagrange multiplies,

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

Fixing λ, we get

$$\hat{\mu}_k^{MML} = \frac{\sum_{i=1}^{n} \lambda_k^i x_i}{\sum_{i=1}^{n} \lambda_k^i} \qquad \hat{\sigma}_k^{2\,MML} = \frac{\sum_{i=1}^{n} \lambda_k^i (x_i - \hat{\mu}_k^{MML})^2}{\sum_{i=1}^{n} \lambda_k^i}$$

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

- Fix θ and maximize λ

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

$$= \sum_{i=1}^{n} \Big[\underbrace{\sum_{k=1}^{K} \lambda_k^i \log(a_{ik}) - \lambda_k^i \log \lambda_k^i}\Big]$$

where $a_{ik} = \pi_k \, e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}} \frac{1}{\sqrt{2\pi}\sigma_k}$

Fix any i,

$$\max_{\lambda_1^i, \dots \lambda_K^i} \sum_{k=1}^{K} \left[\lambda_k^i \log(a_k^i) - \lambda_k^i \log \lambda_k^i\right]$$

s.t $\sum_{k=1}^{K} \lambda_k^i = 1 \qquad 0 \le \lambda_k^i \le 1$

Can be solved analytically

$P(x_i / z_i = k)$ $\quad$ $P(z_i = k)$

$$\hat{\lambda}_k^{i\,MML} = \frac{\left(\frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}}\right) \cdot \pi_k}{\sum_{l=1}^{K}\left(\frac{1}{\sqrt{2\pi}\sigma_l} e^{-\frac{(x_i-\mu_l)^2}{2\sigma_l^2}} \cdot \pi_l\right)} \leftarrow P(x_i)$$

$$P(z_i = k \mid x_i) = \frac{P(x_i / z_i = k) \cdot P(z_i = k)}{P(x_i)}$$

ALGORITHM - EM ALGORITHM (1970's Dempster et. al)

$\rightarrow$ Initialize $\theta^0 = \{ \mu_1^0, \dots, \mu_K^0, \; \sigma_1^0, \dots \sigma_K^0, \; \pi_1^0, \dots \pi_K^0 \}$

iteration

usually comes from Llyod's

$\rightarrow$ until convergence $(\|\theta^{t+1} - \theta^t\| \le \epsilon)$

Tolerance parameter

$$\lambda^{t+1} = \arg\max_{\lambda} \text{ modified\_} \log L(\theta^t, \lambda)$$

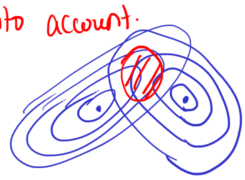
$$\theta^{t+1} = \arg\max_{\theta} \text{ modified\_} \log L(\theta, \lambda^{t+1})$$

Maximization step

Expectation step

$\rightarrow$ end.

- EM produces "soft clustering"
- EM takes variances into account.



- EM clusters need not be voronoi regions!

EM Converges to a local- maximum of log likelihood.