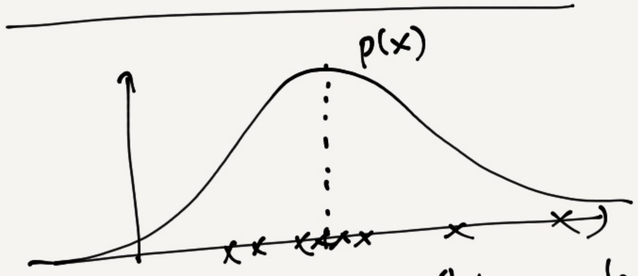
Lecture 4

CS5487 Lecture Notes (2020)
Dr. Antoni B. Chan
Dept of Computer Science
City University of Hong Kong

Mixture Models & Clustering

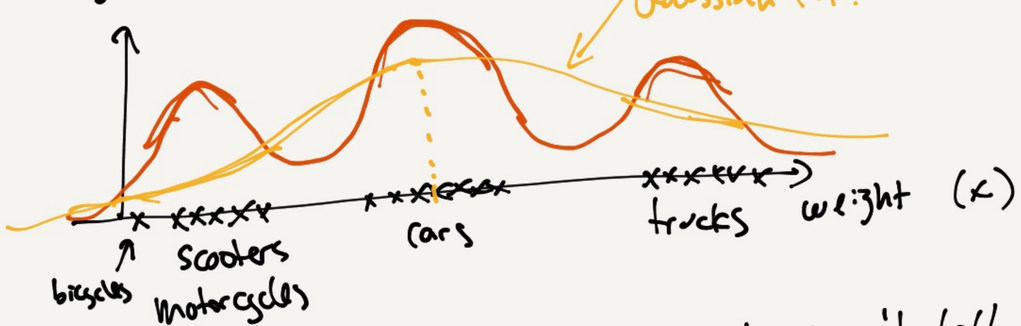


So far our probabilistic models have one mode (peak)

e.g. $p(x) = N(x|\mu, \sigma^2)$

What if the data is more complicated?

e.g. Bridge sensor - measure weight of vehicles



Gaussian doesn't fit well — it doesn't tell the whole story.

Gaussian Mixture Model (GMM)

two random variables:

1) z = hidden state: e.g. vehicle type
   ↑ not observed
   $z \in \{\text{scooter, car, truck}\}$

   $p(z=j) = \pi_j$, $\sum_{j=1}^{K} \pi_j = 1$
   ↑ prior probability of state $z=j$ occurring.

2) x = observation e.g. weight measured by sensor
   likelihood of x is conditioned on the current state z.

   $p(x|z=j) = N(x|\mu_j, \sigma_j^2)$
   ↑ ↑ a Gaussian for each vehicle type.

Generative process: 1) sample a z according to $p(z)$
2) sample x given z, $p(x|z)$

Note: we never see z! only see x!

Likelihood of x:

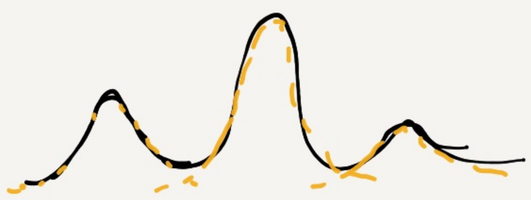
$$p(x) = \sum_j p(x, z=j) = \sum_j p(x|z=j)p(z=j)$$

$$p(x) = \sum_j \pi_j N(x|\mu_j, \sigma_j^2)$$

weighted sum of Gaussians.

π_j: component weight (prior)

$N(x|\mu_j, \sigma_j^2)$: mixture component



Clustering w/ GMMs

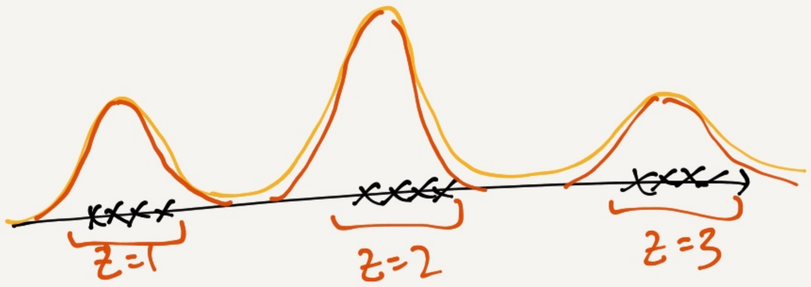
Given data $D = \{x_1, \ldots, x_N\}$

- assume there are K clusters
- assume each cluster is a Gaussian distribution.

Estimate GMM from D.

$\Rightarrow$ we obtain the following:

1) Cluster center and extent (μ_j, σ_j^2)
2) Frequency of each cluster (π_j)
3) the cluster assignment for each x_i (z_i)



Antoni's Hack

- treat z_i as a parameter to be estimated.
- let $z_i \in \{1, \ldots K\}$ be the cluster assignment for x_i
- Goal: maximize the joint log-likelihood of (x, z)

$$\hat{\theta} = \arg\max_\theta \sum_i \log p(x_i, z_i) = \arg\max_\theta \sum_i \log p(x_i | z_i) p(z_i)$$

- Indicator variable trick

let $z_{ij} = \begin{cases} 1, & z_i = j \quad (x_i \text{ is assigned to cluster } j) \\ 0, & \text{otherwise} \end{cases}$

Note $\sum_i z_{ij} = 1$

Then, $p(x_i | z_i) = \prod_{j=1}^K N(x_i | \mu_j, \sigma_j^2)^{z_{ij}}$

$p(z_i) = \prod_{j=1}^K \pi_j^{z_{ij}}$ ← selects the π_{z_i}

Substitute:

$$\hat{\theta} = \arg\max \sum_{i=1}^N \log \left[\prod_{j=1}^K N(x_i | \mu_j, \sigma_j^2)^{z_{ij}} \prod_{j=1}^K \pi_j^{z_{ij}} \right]$$

$$= \arg\max_{\{\pi_j, \mu_j, \sigma_j^2, z_{ij}\}} \sum_{i=1}^N \sum_{j=1}^K \left[z_{ij} \log N(x_i | \mu_j, \sigma_j^2) + z_{ij} \log \pi_j \right]$$

Variables depend on each other, try an alternating maximization scheme:

1) given $\{\pi_j, \mu_j, \sigma_j^2\}$ fixed, then solve for z_{ij}
   - each z_i is independent of others

$$z_{ij} = \underset{z_{ij}}{\text{argmax}} \sum_j z_{ij} \left(\log \pi_j N(x_i|\mu_j, \sigma_j^2)\right)$$

$z_{ij} \in \{0,1\}$; pick j w/ largest $\log \pi_j N(x_i|\mu_j, \sigma_j^2)$

$$z_i = \underset{j}{\text{argmax}} \log \pi_j N(x_i|\mu_j, \sigma_j^2)$$

2) given z_{ij} fixed, solve for $\{\pi_j, \mu_j, \sigma_j^2\}$

mean μ_j:

$$\ell(\mu_j) = \sum_i -\frac{1}{2}\frac{(x_i - \mu_j)^2}{\sigma_j^2} \cdot z_{ij}$$

($\log N(x_i|\mu_j, \sigma^2)$)

$$\frac{\partial \ell}{\partial \mu_j} = \sum_i z_{ij} \frac{1}{\sigma_j^2}\left(-\frac{1}{2}\right)(x_i - \mu_j) 2(-1) = 0$$

$$= \sum_i z_{ij} x_i - \sum_i z_{ij} \mu_j = 0$$

$$\Rightarrow \hat{\mu}_j = \frac{1}{\sum_i z_{ij}} \sum_i z_{ij} x_i$$

($\sum_i z_{ij}$: # of points assigned to cluster j; $\sum_i z_{ij} x_i$: sum of points x_i assigned to j)

mean of points assigned to cluster j.

Similarly

$$\hat{\sigma}_j^2 = \frac{\sum_i z_{ij}(x_i - \hat{\mu}_j)^2}{\sum_i z_{ij}}$$ — variance of points in j

$$\hat{\pi}_j = \frac{1}{N}\sum_i z_{ij}$$ — fraction of points assigned to j.

3) iterate (1) & (2) until convergence.

Note:

- this 2-step procedure always maximizes the original objective $\Rightarrow$ converges to a local maximum
- σ_j^2 = fixed value, $\pi_j = \frac{1}{K}$

  $\Rightarrow$ K-means clustering algorithm.
- need some initialization of (π, μ, σ^2) or (z_i) to start the algorithm.
  - different init. may give different answers.
- not maximizing the actual $p(x)$ (data likelihood), so it's not MLE.

[we are optimizing z, rather than marginalizing it out as in $p(x)$.]

Expectation Maximization (EM) algorithm

(Dempster, Laird, Rubin 1977 — 62K citations on google scholar in 2020.

- MLE when there are hidden variables

X = observation r.v.
Z = hidden r.v.
$p(x) = \sum_z p(x|z)p(z)$

Goal: maximize data LL:

$$\hat{\theta} = \arg\max_\theta \log p(x) = \arg\max_\theta \log \sum_z p(x|z)p(z)$$

Key observation:

- if we had the "complete" data $\{x, z\}$ observed, then the problem is easy.

$$\hat{\theta} = \arg\max_\theta \log p(x,z) = \arg\max_\theta \sum_i \log p(x_i|z_i) + \log p(z_i)$$

$\Rightarrow$ Step 2 in Antonis hack.

- so, guess the values in a probabilistic way.
  1) select expected values of z given previous model $\Rightarrow \hat{z}$
  2) maximize $\log p(x, \hat{z})$ to get new model.
  3) repeat (1)&(2) until convergence.

Formally

0) Select initial model $\hat{\theta}^{(old)}$

1) E-step: $Q(\theta; \hat{\theta}^{(old)}) = E_{z|x,\hat{\theta}^{(old)}}\left[\log p(x,z|\theta)\right]$

"Q" function; previous parameters; new parameters we will optimize; conditional expectation using the previous model $\hat{\theta}^{(old)}$; joint LL

2) M-step: $\hat{\theta}^{(new)} = \arg\max_\theta Q(\theta; \hat{\theta}^{(old)})$

3) $\hat{\theta}^{(old)} \leftarrow \hat{\theta}^{(new)}$, repeat (1) & (2) until convergence.

EM for GMMs

$$\log p(X, z) = \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij} \log \pi_j \, N(x_i \mid \mu_j, \sigma_j^2)$$

1) **E-step**:

$$Q(\theta; \hat{\theta}^{(old)}) = E_{z|X, \hat{\theta}^{(old)}} \left[\log p(X, z) \right]$$

$$= \sum_i \sum_j E_{z|X, \hat{\theta}^{(old)}} [z_{ij}] \log \pi_j \, N(x_i \mid \mu_j, \sigma_j^2)$$

$E_{z|X, \hat{\theta}^{(old)}}[z_{ij}] = \hat{z}_{ij}$

(same form as step (2) in my hack, but with z_{ij} replaced with $\hat{z}_{ij}$)

$$\hat{z}_{ij} = E_{z|X, \hat{\theta}^{(old)}} [z_{ij}]$$

↓ Expectation of indicator variable (P1.J)

$$= p(z_{ij} = 1 \mid X, \hat{\theta}^{(old)})$$

$= \vdots$ (tutorial)

$$= \frac{\hat{\pi}_j \, N(x_i \mid \hat{\mu}_j, \hat{\sigma}_j^2)}{\sum_k \hat{\pi}_k \, N(x_i \mid \hat{\mu}_k, \hat{\sigma}_k^2)}$$

} "soft assignment" to cluster j using the $\hat{\theta}^{(old)}$

$$= p(z_i = j \mid x_i, \hat{\theta}^{(old)})$$ ← posterior probability of $z_i \mid x_i$

Note: $0 \le \hat{z}_{ij} \le 1$ soft assignment $\quad z_{ij} \in \{0, 1\}$ hard assignment

M-step: same as before $z_{ij} \to \hat{z}_{ij}$

Summary EM-GMM

E-step: $\hat{z}_{ij} = p(z_i \mid x_i, \hat{\theta}^{(old)})$ ← soft assignment using $\hat{\theta}^{(old)}$

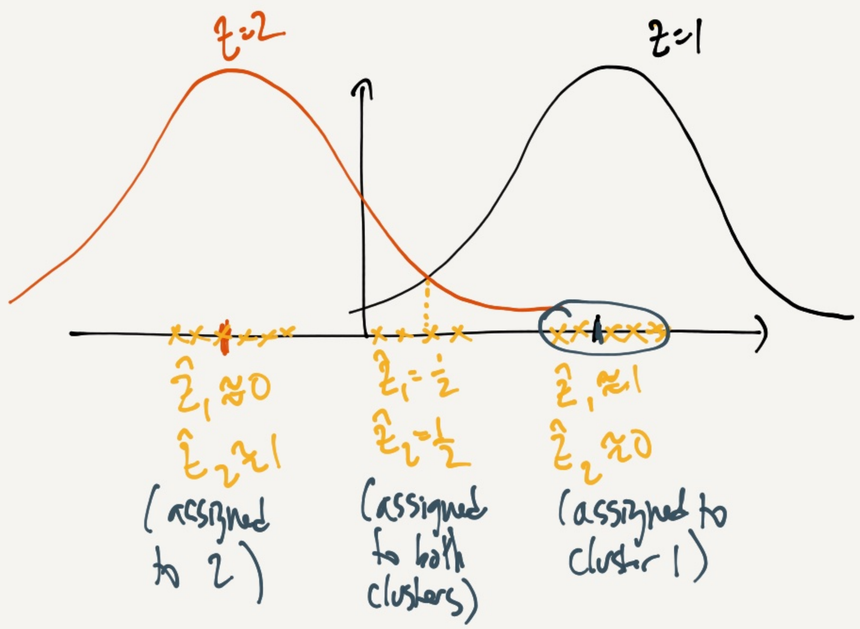
M-step:

$$\hat{\mu}_j = \frac{1}{\hat{N}_j} \sum_i \hat{z}_{ij} x_i$$ ← sample mean w/ weighted points by soft assignments

$$\hat{N}_j = \sum_i \hat{z}_{ij}$$ ← total # of points in cluster j (soft weighted)

$$\hat{\sigma}_j^2 = \frac{1}{\hat{N}_j} \sum_i \hat{z}_{ij} (x_i - \hat{\mu}_j)^2$$ ← "

$$\hat{\pi}_j = \frac{1}{N} \hat{N}_j$$ ← "



Notes:

1) General algorithm - EM is a general framework for MLE on models w/ hidden variables (not just mixture models)

2) convergence - after each EM iteration, the data LL increases → converges to local maximum. (could be arbitrarily slow)

3) initialization - different init give different estimates of θ. pick the θ w/ maximum $p(x|\theta)$.

Tutorial 4

4.12 - Lagrange Multipliers

in EM $\Rightarrow z_j = N_j$

$\sum_{j=1}^{K} \pi_j = 1$ constraint , $\pi_j \geq 0$

need to solve: $\pi_j = \underset{j}{\text{argmax}} \sum_{j=1}^{K} z_j \log \pi_j$

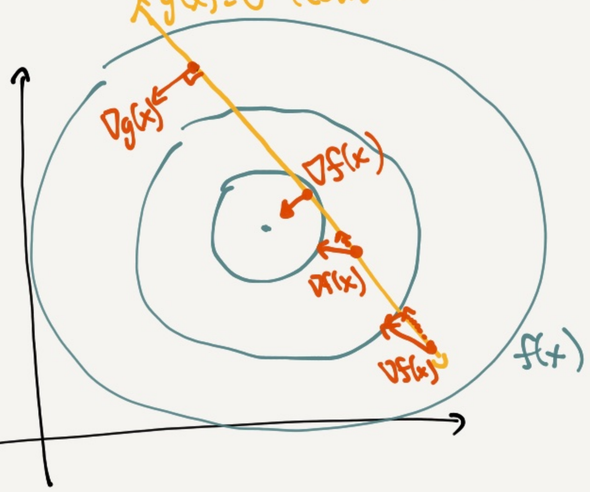
$\sum_j \pi_j = 1$ equality constraint

Look at the general problem

$x^* = \underset{x}{\text{argmax}}\ f(x)$ ⇐ objective function

s.t. $g(x) = 0$ ↖ constraint function

- $\nabla g(x)$ is $\perp$ to $g(x)=0$
- At the optimum, $\nabla f(x)$ is $\perp$ to $g(x)=0$. (otherwise we could move along $g(x)=0$ to increase $f(x)$.)



At the optimum, we have

$$\nabla f(x) + \lambda \nabla g(x) = 0 \quad , \quad \lambda \neq 0$$

Define Lagrangian function:

$$L(x, \lambda) = f(x) + \lambda g(x)$$

Find a stationary point (x, λ) of the Lagrangian.

$$\frac{\partial}{\partial x} L(x,\lambda) = 0 \quad , \quad \frac{\partial}{\partial \lambda} L(x,\lambda) = 0.$$

a) $g(\pi) = \sum_{j=1}^{K} \pi_j - 1 = 0$

$f(\pi) = \sum_j z_j \log \pi_j$

Lagrangian:

$$L(\pi, \lambda) = \sum_j z_j \log \pi_j + \lambda \left(\sum_j \pi_j - 1 \right)$$

$$\frac{\partial L}{\partial \lambda} = \sum_j \pi_j - 1 = 0 \Rightarrow \sum_j \pi_j = 1$$

$$\frac{\partial L}{\partial \pi_j} = \left(\frac{z_j}{\pi_j} + \lambda = 0 \right) \times \pi_j$$

$$\Rightarrow \left(z_j + \lambda \pi_j = 0 \right) \text{ sum over } j$$

$$\sum_j z_j + \lambda \sum_j \pi_j = 0$$

$$\Rightarrow \lambda = -\sum_j z_j$$

$$z_j + \left(-\sum_j z_j \right) \pi_j = 0$$

$$\Rightarrow \hat{\pi}_j = \frac{z_j}{\sum_j z_j}$$

4.6 - mixture of exponentials

$$p(x) = \sum_j \pi_j \lambda_j e^{-\lambda_j x}$$

π_j: comp. prob. $\lambda_j e^{-\lambda_j x}$: exponential density $p(x|j)$

Given data $D = \{x_i\}_{i=1}^N$

Define hidden variable z_i for each x_i.

Joint LL:

$$\log p(x,z) = \sum_{i=1}^N \sum_{j=1}^K z_{ij} \log \pi_j + z_{ij} \log p(x|j)$$

E-step:

$$Q(\hat{\theta}:\hat{\theta}^{(old)}) = E_{z|x,\hat{\theta}^{(old)}}\left[\log p(x,z|\theta)\right]$$

$$= E_{z|x,\hat{\theta}^{(old)}}\left[\sum_i \sum_j z_{ij} \log \pi_j + z_{ij} \log p(x|j)\right]$$

$$= \sum_i \sum_j \underbrace{E_{z|x,\hat{\theta}^{(old)}}[z_{ij}]}_{\hat{z}_{ij}} \log \pi_j + \underbrace{E_{z|x}[z_{ij}]}_{\hat{z}_{ij}} \log p(x|j)$$

$$\hat{z}_{ij} = E_{z|x,\hat{\theta}^{(old)}}[z_{ij}]$$

$$\hat{z}_{ij} = E_{z|x,\hat{\theta}^{(old)}}[z_{ij}] \quad (z_{ij}: \text{indicator variable})$$

$$= p(z_{ij}=1|x,\hat{\theta}^{(old)}) \cdot 1 + p(z_{ij}=0|x,\hat{\theta}^{(old)}) \cdot 0$$

$$= p(z_{ij}=1|x,\hat{\theta}^{(old)})$$

(expectation of indicator variable is the prob of the indicator = 1)

$$= p(z_i=j|x)$$

↓ Bayes' Rule

$$= \frac{p(x|z_i=j)\,p(z_i=j)}{p(x)}$$

$\Rightarrow x = \{x_i, \underbrace{x_1, x_2 \ldots}_{x_{\setminus i}}\}$

independence of x_i & $x_{\setminus i}$

$$= \frac{p(x_{\setminus i})\,p(x_i|z_i=j)\,p(z_i=j)}{p(x_{\setminus i})\,p(x_i)}$$

$x_i \perp x_{\setminus i}$

$\Rightarrow p(x) = p(x_i) \cdot p(x_{\setminus i})$

$$= \frac{p(x_i|z_i=j)\,p(z_i=j)}{\sum_k p(x_i|z_i=k)\,p(z_i=k)}$$

$\sum_k p(x_i|z_i=k)\,p(z_i=k) = p(x_i)$

$$\hat{z}_{ij} = \frac{\pi_j p(x_i|j)}{\sum_k \pi_k p(x_i|k)}$$

(evaluated w/ $\hat{\theta}^{(old)}$)

Note: $p(x|z_i=j) = p(x_{\setminus i}|z_i=j)\,p(x_i|z_i=j)$

doesn't affect $x_{\setminus i}$

$= p(x_{\setminus i})\,p(x_i|z_i=j)$

M-step

$$(\hat{\pi}, \hat{\lambda}) = \underset{\lambda,\pi}{\operatorname{argmax}} \; Q(\theta; \hat{\theta}^{(old)})$$

$$= \underset{\lambda,\pi}{\operatorname{argmax}} \sum_i \sum_j \hat{z}_{ij} (\log \pi_j + \hat{z}_{ij} \log p(x|j))$$

Look at π first

$$\hat{\pi} = \underset{\pi_j}{\operatorname{argmax}} \sum_i \sum_j \hat{z}_{ij} \log \pi_j$$

$$\sum_j \pi_j = 1$$

$$\sum_j \Big(\sum_i \hat{z}_{ij}\Big) \log \pi_j, \quad \hat{N}_j$$

$$= \underset{\text{s.t.} \sum_j \pi_j = 1}{\operatorname{argmax}} \sum_j \hat{N}_j \log \pi_j$$

Use 4.12 a)

$$\Rightarrow \quad \hat{\pi}_j = \frac{\hat{N}_j}{\sum_K \hat{N}_K} = \frac{\hat{N}_j}{N}$$

Look at λ_j

$$\lambda_j = \underset{\lambda_j}{\operatorname{argmax}} \sum_i \hat{z}_{ij} \log p(x|j)$$

$$= \underset{\lambda_j}{\operatorname{argmax}} \sum_i \hat{z}_{ij} (\log \lambda_j - \lambda_j x_i)$$

$$\Rightarrow \frac{\partial}{\partial \lambda_j} = \sum_i \hat{z}_{ij} \Big(\frac{1}{\lambda_j} - x_i\Big) = 0$$

$$= \frac{1}{\lambda_j} \hat{N}_j - \sum_i \hat{z}_{ij} x_i = 0$$

$\hat{N}_j = \sum_i \hat{z}_{ij}$

$$\Rightarrow \quad \frac{1}{\lambda_j} = \frac{1}{\hat{N}_j} \sum_i \hat{z}_{ij} x_i$$

z_i = hidden assignment variable $\in \{1, \dots, k\}$

z_{ij} = indicator that $z_i = j = \begin{cases} 1, & z_i = j \\ 0, & \text{otherwise} \end{cases}$

"one-hot-encoding"

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow z_i = 1$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow z_i = 2$$

$\hat{z}_{ij} = p(z_i = j \mid x_i)$, computed in E-step

, soft assignment of x_i to j