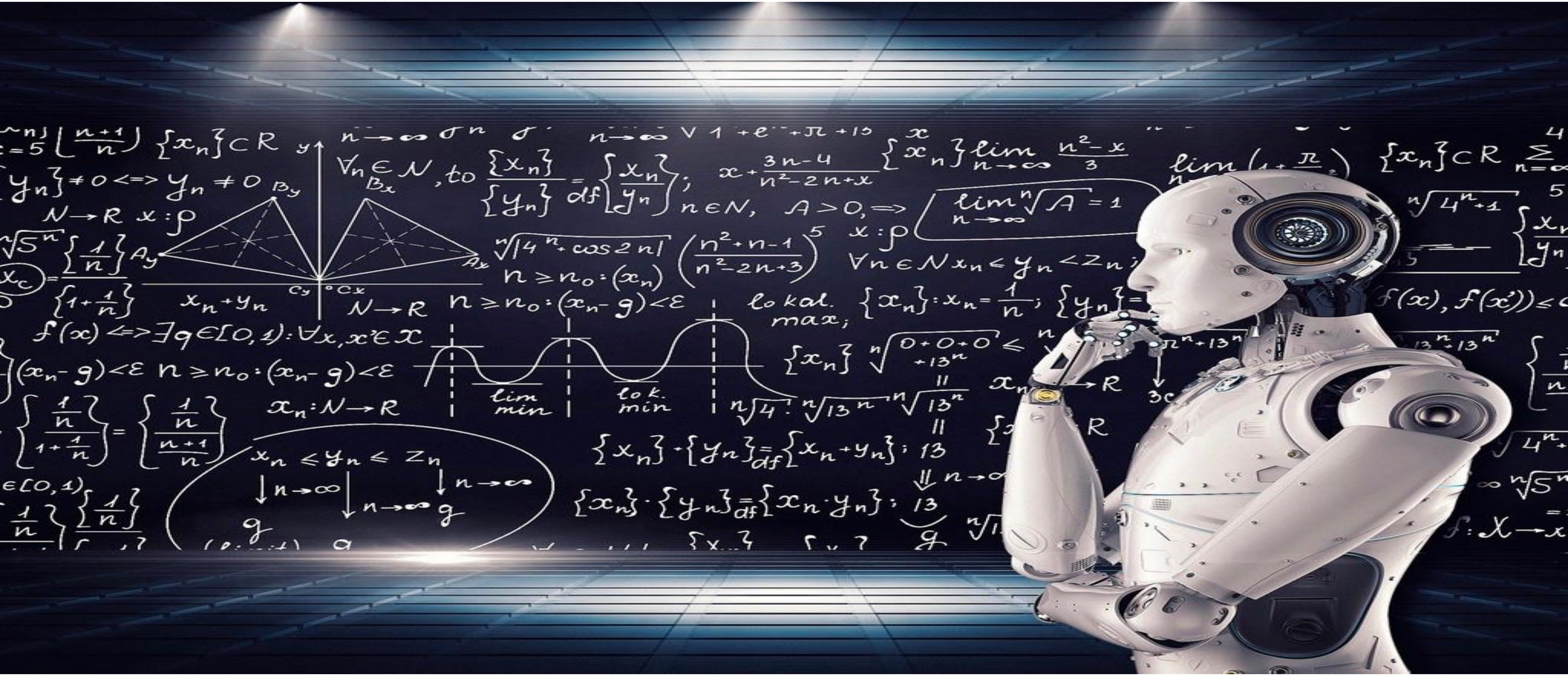
Density Estimation



Bayesian Learning Recap

- Prior classifier: $P(A) > P(B)$
- ML classifier: $P(x|A) > P(x|B)$ – assuming $P(A) = P(B)$
- MAP classifier:

$$P(A|x) = P(x|A)P(A) > P(x|B)P(B) = P(B|x)$$

* Dropping $P(x)$ from the denominator

- And we said we can use log probability – helps in the calculations

How to calculate the probability

- Parametric models
  - If we know \ can guess the distribution type we can estimate the parameters of the distribution
  - Gaussian Naïve Bayes
- Non parametric models
  - Histogram (=count…)
  - Discrete Naïve Bayes

Parametric

- For each class we will estimate the distribution parameter according to the train dataset
- If we're talking about normal distribution parameters, we need to estimate the mean and the variance:

$$\mu = \frac{1}{m}\sum_{k=1}^{m} x_k$$

$$\sigma^2 = \frac{1}{m}\sum_{k=1}^{m} (x_k - \mu)^2$$

Parametric

- Now, we can estimate the parameter for each likelihood probability, for each class:

$$\mu_i = \frac{1}{|A_i|}\sum_{x\in A_i} x$$

$$\sigma_i{}^2 = \frac{1}{|A_i|}\sum_{x\in A_i}(x-\mu_i)^2$$

- And then classify according to the largest probability given by the normal distribution formula:

$$P(x|A_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2}$$

Parametric

- But, this was good only for 1 attribute
- What if we have more than 1?
- In this case each likelihood probability will be estimated according to multivariate normal distribution
- For this we will need mean vector (each dimension will be the mean for some attribute) and the covariance matrix

The covariance metrix

Variance

$$\mathbf{S} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

$\left|\mathbf{S}\right|$ - is the determinant of the covariance matrix

$\mathbf{S}^{-1}$ - is the inverse matrix of the covariance matrix

Parametric

- For each attribute we will find the mean and the variance as before and we will create the mean vector and the covariance matrix
- We will classify according to the multivariate normal distribution:

$$P(\bar{x}|A_i) = \frac{1}{\sqrt{(2\pi)^d |S|^2}} e^{\left[-\frac{1}{2}(\bar{x}-\overline{\mu_i})^T S^{-1}(\bar{x}-\overline{\mu_i})\right]}$$

Non Parametric

- If we don't know the type of distribution?
- We need another way to estimate the probabilities $P(x|A_i)$ and $P(A_i)$
- The prior probability $P(A_i)$ can be estimated from the classes frequency in the training set
- But what with the likelihood?

Non Parametric

- In order to estimate the likelihood for a given instance we need a huge dataset

- If we have d attributes the number of possible terms in the likelihood $P(x_1, x_2, \dots, x_d | A_i)$ is $k \cdot |V_1| \cdot |V_2| \cdots |V_d|$
  Where k is the number of classes.

- We need a way \ assumption to overcome this problem

Naïve Bayes

- If we assume that all attributes are independent **given the class**, we will get:

$$P(x_1, x_2, \dots, x_d | A_i) = \prod_{j=1}^{d} P(x_j | A_i)$$

- And now we can find the MAP:

$$V_{NB} = \underset{i}{\operatorname{argmax}} P(A_i) \prod_{j=1}^{d} P(x_j | A_i)$$

- In this assumption we lower the number of possible terms in the likelihood :

$$k \sum_{j=1}^{d} |V_j|$$

Naïve Bayes

- In practice this algorithm works pretty well
- But, why Naïve Bayes works, although the approximation for the likelihood is bad $\left(P(\bar{x}|A_i) \approx \prod_{j=1}^{d} P\left(x_j|A_i\right)\right)$?
- The approximation is not what we are looking for… we looking to compare the posterior
- So , what we need is:

$$\underset{i}{\operatorname{argmax}} P(A_i) \prod_{j=1}^{d} P\left(x_j|A_i\right) = \underset{i}{\operatorname{argmax}} P(A_i) P(x_1, x_2, \dots, x_d|A_i)$$

Discrete Naïve Bayes

- Estimate the probability according to this formula:

$$P(x_j|A_i) = \frac{n_{ij}}{n_i}$$

- Where:
  - n_{ij} - is the number of training instances with the class A_i and the value x_j in the relevant attribute
  - n_i - is the number of training instances with the class A_i

Discrete Naïve Bayes

- We still have one problem to solve
- Some of the estimations can be zero according to the training set $P\left(x_j|A_i\right) = 0$
- This will make the likelihood probability to be zero due to the multiplications
- In order to solve that we will use Laplace estimation

Laplace estimation

- We will estimate the probability according to this formula:

$$P\left(x_j|A_i\right) = \frac{n_{ij} + 1}{n_i + |V_j|}$$

- Where:
  - n_{ij} - is the number of training instances with the class A_i and the value x_j in the relevant attribute
  - n_i - is the number of training instances with the class A_i
  - $|V_j|$ - is the number of possible values of the relevant attribute

Discrete Naïve Bayes - example

- We want to classify the best treatment for some disease A or B
- We have history data that contains:
    - Gender, Blood Pressure, Age and the Treatment that the patient received
- We want to classify new patient treatment with Naïve Bayes

Discrete Naïve Bayes - example

Gender	Blood Pressure	Age	Treatment
Male	Normal	Young	A
Male	High	Old	A
Male	High	Old	A
Female	High	Young	A
Female	Normal	Young	A
Female	High	Old	A
Male	Low	Young	B
Male	Low	Old	B
Male	Normal	Old	B
Female	Low	Young	B
Female	Normal	Old	B
Female	Normal	Old	B

Discrete Naïve Bayes - example

- In order to build the table for the classifier we will use the Laplace formula:

$$P\left(x_j|A_i\right) = \frac{n_{ij} + 1}{n_i + |V_j|}$$

Discrete Naïve Bayes - example

$$P(A) = \frac{6}{12} = \frac{1}{2}$$

$$P(male|A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$$

$$P(female|A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$$

$$P(high|A) = \frac{4+1}{6+3} = \frac{5}{9}$$

$$P(normal|A) = \frac{2+1}{6+3} = \frac{3}{9}$$

$$P(low|A) = \frac{0+1}{6+3} = \frac{1}{9}$$

$$P(young|A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$$

$$P(old|A) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$$

$$P(B) = \frac{6}{12} = \frac{1}{2}$$

$$P(male|B) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$$

$$P(female|B) = \frac{3+1}{6+2} = \frac{4}{8} = \frac{1}{2}$$

$$P(high|B) = \frac{0+1}{6+3} = \frac{1}{9}$$

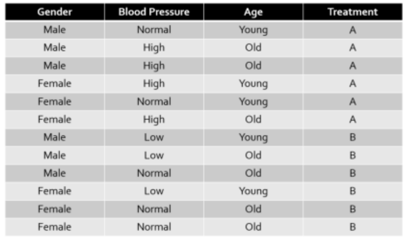
$$P(normal|B) = \frac{3+1}{6+3} = \frac{4}{9}$$

$$P(low|B) = \frac{3+1}{6+3} = \frac{4}{9}$$

$$P(young|B) = \frac{2+1}{6+2} = \frac{3}{8}$$

$$P(old|B) = \frac{4+1}{6+2} = \frac{5}{8}$$

$$P\left(x_j|A_i\right) = \frac{n_{ij}+1}{n_i + |V_j|}$$



Gender	Blood Pressure	Age	Treatment
Male	Normal	Young	A
Male	High	Old	A
Male	High	Old	A
Female	High	Young	A
Female	Normal	Young	A
Female	High	Old	A
Male	Low	Young	B
Male	Low	Old	B
Male	Normal	Old	B
Female	Low	Young	B
Female	Normal	Old	B
Female	Normal	Old	B

Discrete Naïve Bayes - example

- Classify the following new instances:
  - $male, young, high$
    - $P(A|male, young, high) = P(A) \cdot P(male|A) \cdot P(young|A) \cdot P(high|A) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{5}{9} = \frac{5}{54}$
    - $P(B|male, young, high) = P(B) \cdot P(male|B) \cdot P(young|B) \cdot P(high|B) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{8} \cdot \frac{1}{9} = \frac{3}{288}$

    After normalization:

    - $P(A|male, young, high) = \frac{\frac{5}{54}}{\frac{5}{54} + \frac{3}{288}} = 0.9$ $\quad P(B|male, young, high) = \frac{\frac{3}{288}}{\frac{5}{54} + \frac{3}{288}} = 0.1$
    - The instance classification is A
  - $female, old, low$
    - $P(A|female, old, low) = P(A) \cdot P(female|A) \cdot P(old|A) \cdot P(low|A) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{4}{8} \cdot \frac{1}{9} = \frac{4}{288}$
    - $P(B|female, old, low) = P(B) \cdot P(female|B) \cdot P(old|B) \cdot P(low|B) = \frac{1}{2} \ast \frac{1}{2} \ast \frac{5}{8} \ast \frac{4}{9} = \frac{20}{288}$

    After normalization:

    - $P(A|female, old, low) = \frac{\frac{4}{288}}{\frac{4}{288} + \frac{20}{288}} = 0.167$ $\quad P(B|female, old, low) = \frac{\frac{20}{288}}{\frac{4}{288} + \frac{20}{288}} = 0.833$
    - The instance classification is B

Expectation Maximization (EM) Algorithm

- Iterative method for parameter estimation where layers of data are missing from the observation
- Dempster, Laird, Rubin, J of the Royal Stat Soc, 1977
- Many variations followed. Research into methodology and applications is very active
- Has two steps:
  - Expectation (E) and Maximization (M)
- Applicable to a wide range of machine learning and inference tasks

Basic setting in EM

- D is a set of data points: **observed** data
- Θ is a parameter vector.
- EM is an iterative method for finding θ_{ML}
- It’s mostly useful when
  - Calculating P(x | θ) directly is hard.
  - Calculating P(x,z|θ) is simpler, where z is some “hidden” data (or “missing” data)
  - The hidden data is assumed to be determined by some other rv Z, which is part of the model.
  - Note: the model, under θ, controls both X and Z but in D we only see the values of X.

Randomly selecting one of two coins

- There are two coins, with p_A and p_B
- One of the coins is selected, with w_A and w_B probabilities
- Then it is tossed 10 times
- We observe the results of many repeats of this exercise
- If we know which coin is tossed in each set then we can do MLE and get both the ps and the ws
- But we don’t …
- Lets see what EM can do here

🟠	HHHHTHHHHH
🟠	THHHHHHHTH
🟠	HHHHHHHTHH
🟢	HHTHTTHHTT
🟠	HHTHHHHHTH
🟢	HTTHTHHHTT
🟢	HTTHTHHHHT
🟠	HTHHTHHHHT

The EM algorithm

- Consider a set of starting parameters

- Use these to "estimate" the <u>values</u> of the missing data, per observed data point

- Use the "complete" data to update all parameters

- Repeat until convergence

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2
🟠	THHHHHHHTH		
🟠	HHHHHHHTHH		
🟢	HHTHTTHHTT		
🟠	HHTHHHHHTH		
🟢	HTTHTHHHTT		
🟢	HTTHTHHHHT		
🟠	HTHHTHHHHT		

Coin A responsibilities

Coin B responsibilities

1

Init $p_A = 0.6$
$p_B = 0.5$
ws are 0.5

2

Compute responsibilities

$$P_A(x_1) = w_A \binom{10}{9} 0.6^9 0.4^1 = 0.04$$

$$P_B(x_1) = w_B \binom{10}{9} 0.5^9 0.5^1 = 0.01$$

$$r(x_1, A) = \frac{0.04}{0.05} = 0.8$$

$$r(x_1, B) = \frac{0.01}{0.05} = 0.2$$

Note: use aposteriori estimates

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2
🟠	THHHHHHHTH	0.76	0.24
🟠	HHHHHHHTHH		
🟢	HHTHTTHHTT		
🟠	HHTHHHHHTH		
🟢	HTTHTHHHTT		
🟢	HTTHTHHHHT		
🟠	HTHHTHHHHT		

Coin A responsibilities

Coin B responsibilities

$$P_A(x_2) = w_A \binom{10}{8} 0.6^8 0.4^2 = 0.12$$

$$P_B(x_2) = w_B \binom{10}{8} 0.5^8 0.5^2 = 0.044$$

$$r(x_2, A) = \frac{0.12}{0.164} = 0.76$$

$$r(x_2, B) = \frac{0.044}{0.164} = 0.24$$

①

Init $p_A = 0.6$
$p_B = 0.5$
ws are 0.5

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2
🟠	THHHHHHHTH	0.76	0.24
🟠	HHHHHHHTHH	0.8	0.2
🟢	HHTHTTTHHTT		
🟠	HHTHHHHHTH	0.76	0.24
🟢	HTTHTHHHTT		
🟢	HTTHTHHHHT		
🟠	HTHHTHHHHT		

Coin A responsibilities

Coin B responsibilities

①

Init $p_A = 0.6$
$p_B = 0.5$
ws are 0.5

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2
🟠	THHHHHHHTH	0.76	0.24
🟠	HHHHHHHTHH	0.8	0.2
🟢	HHTHTTHHTT	0.45	0.55
🟠	HHTHHHHHTH	0.76	0.24
🟢	HTTHTHHHTT	0.45	0.55
🟢	HTTHTHHHHT		
🟠	HTHHTHHHHT		

Coin A responsibilities

Coin B responsibilities

①

Init $p_A = 0.6$
$p_B = 0.5$
ws are 0.5

$$P_A(x_4) = w_A \binom{10}{5} 0.6^5 0.4^5$$

$$P_B(x_4) = w_B \binom{10}{5} 0.5^5 0.5^5$$

$$r(x_4, A) = 0.45$$

$$r(x_4, B) = 0.55$$

© Ben Galili

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2
🟠	THHHHHHHTH	0.76	0.24
🟠	HHHHHHHTHH	0.8	0.2
🟢	HHTHTTHHTT	0.45	0.55
🟠	HHTHHHHHTH	0.76	0.24
🟢	HTTHTHHHTT	0.45	0.55
🟢	HTTHTHHHHT	0.55	0.45
🟠	HTHHTHHHHT	0.64	0.36

Coin A responsibilities

Coin B responsibilities

1

Init $p_A = 0.6$
$p_B = 0.5$
ws are 0.5

2

Compute responsibilities

3

Compute new assignments:

$$New\ w_A = \frac{1}{N}\sum_{i=1}^{N} r(x_i, A)$$

$$New\ w_B = \frac{1}{N}\sum_{i=1}^{N} r(x_i, B)$$

$$New\ w_A = \frac{1}{8}\sum_{i=1}^{8} r(x_i, A) = \frac{5.2}{8} = 0.65$$

$$New\ w_B = \frac{1}{8}\sum_{i=1}^{8} r(x_i, B) = \frac{2.8}{8} = 0.35$$

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2	0.9
🟠	THHHHHHHTH	0.76	0.24	0.8
🟠	HHHHHHHTHH	0.8	0.2	0.9
🟢	HHTHTTTHHTT	0.45	0.55	0.5
🟠	HHTHHHHHTH	0.76	0.24	0.8
🟢	HTTHTHHHTT	0.45	0.55	0.5
🟢	HTTHTHHHHT	0.55	0.45	0.6
🟠	HTHHTHHHHT	0.64	0.36	0.7

① Init $p_A = 0.6$
$p_B = 0.5$
ws are 0.5

② Compute responsibilities

$r(x_i,A)$

$r(x_i,B)$

Value observed at i: $v(i)$

3+4 Compute MLEs for the model parameters:

$$p_A = \frac{1}{(New\ w_A)N}\sum_{i=1}^{N} r(x_i,A)v(i)$$

$$p_B = \frac{1}{(New\ w_B)N}\sum_{i=1}^{N} r(x_i,B)v(i)$$

3+4

$$p_A = \frac{1}{5.2}\sum_{i=1}^{8} r(x_i,A)v(i) = 0.745$$

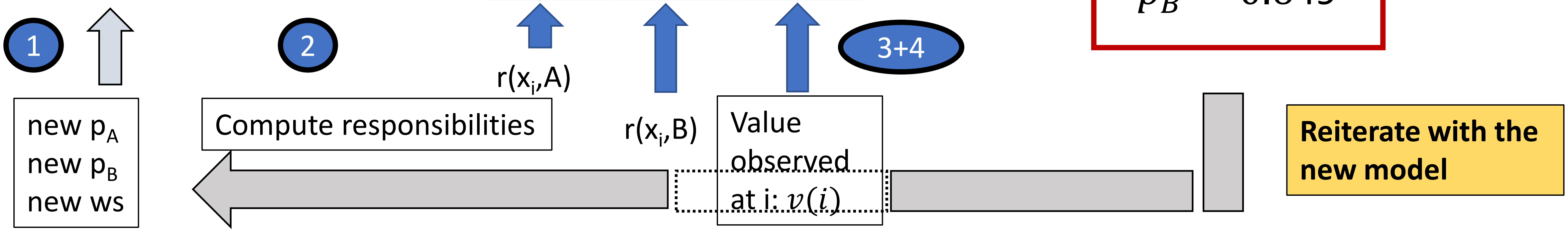
$$p_B = \frac{1}{2.8}\sum_{i=1}^{8} r(x_i,B)v(i) = 0.649$$

EM: uncovering the coins …

🟠	HHHHTHHHHH	0.8	0.2	0.9
🟠	THHHHHHHTH	0.76	0.24	0.8
🟠	HHHHHHHTHH	0.8	0.2	0.9
🟢	HHTHTTHHTT	0.45	0.55	0.5
🟠	HHTHHHHHTH	0.76	0.24	0.8
🟢	HTTHTHHHTT	0.45	0.55	0.5
🟢	HTTHTHHHHT	0.55	0.45	0.6
🟠	HTHHTHHHHT	0.64	0.36	0.7

3+4

$$w_A = 0.65$$
$$w_B = 0.35$$
$$p_A = 0.745$$
$$p_B = 0.649$$



$r(x_i,A)$

$r(x_i,B)$

Value observed at i: $v(i)$

new p_A
new p_B
new ws

Compute responsibilities

Reiterate with the new model

The EM algorithm for two coins

- Consider a set of starting parameters, including the parameters of Z
- Use these to “estimate” the values of the missing data, per observed data point
  - Compute responsibilities using MAP
- Use the “complete” data to update all parameters (of both Z and X|Z)

$$New\ w_A = \frac{1}{N}\sum_{i=1}^{N} r(x_i, A)$$

$$p_A = \frac{1}{(New\ w_A)N}\sum_{i=1}^{N} r(x_i, A)v(i)$$

$$New\ w_B = \frac{1}{N}\sum_{i=1}^{N} r(x_i, B)$$

$$p_B = \frac{1}{(New\ w_B)N}\sum_{i=1}^{N} r(x_i, B)v(i)$$

- Repeat until convergence

EM for GMMs

- Step 1: Expectation (E-step)
  - Evaluate the “responsibilities” of each data point to each Gaussian using the current parameters
- Step 2: Maximization (M-step)
  - Re-estimate parameters (ws, μs and σs) using the existing “responsibilities”
  - That is – every data point, x, contributes to each Gaussian component, G_i, in proportion to its responsibility: $r(x,G_i)$

Gaussian mixtures equations

- Responsibilities:

$$r(x,k) = \frac{w_k N(x|\mu_k\ ,\sigma_k)}{\sum_{j=1}^{K} w_j N(x|\mu_j\ ,\sigma_j)}$$

- Weights:

$$New\ w_j = \frac{1}{N}\sum_{i=1}^{N} r(x_i, j)$$

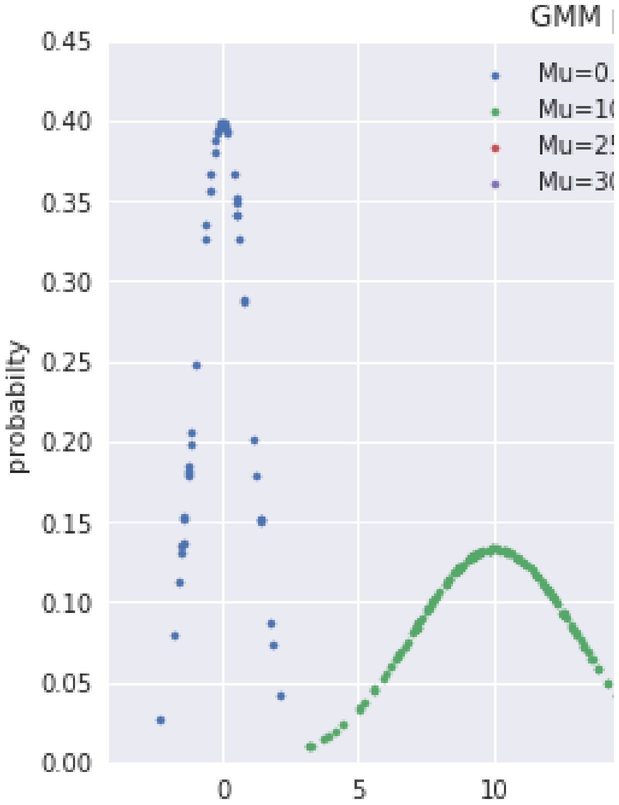
- Mean:

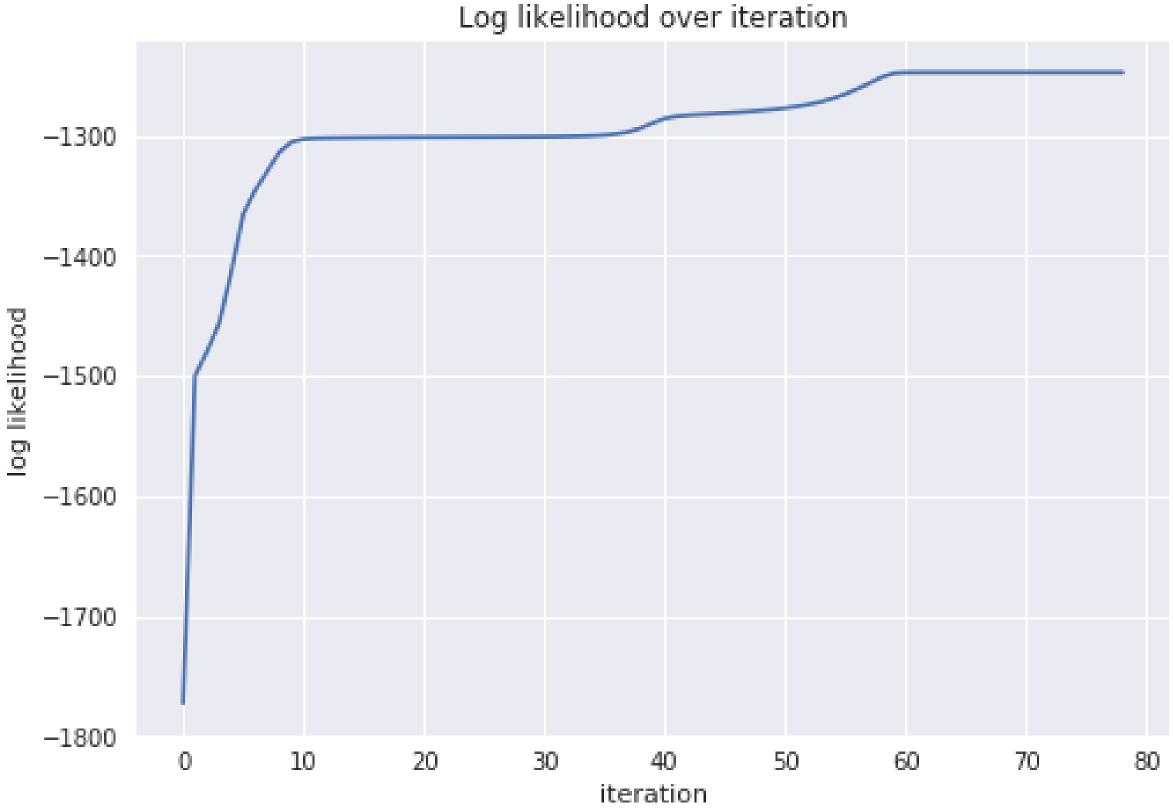
$$New\ \mu_j = \frac{1}{(New\ w_j)N}\sum_{i=1}^{N} r(x_i, j)\ x_i$$

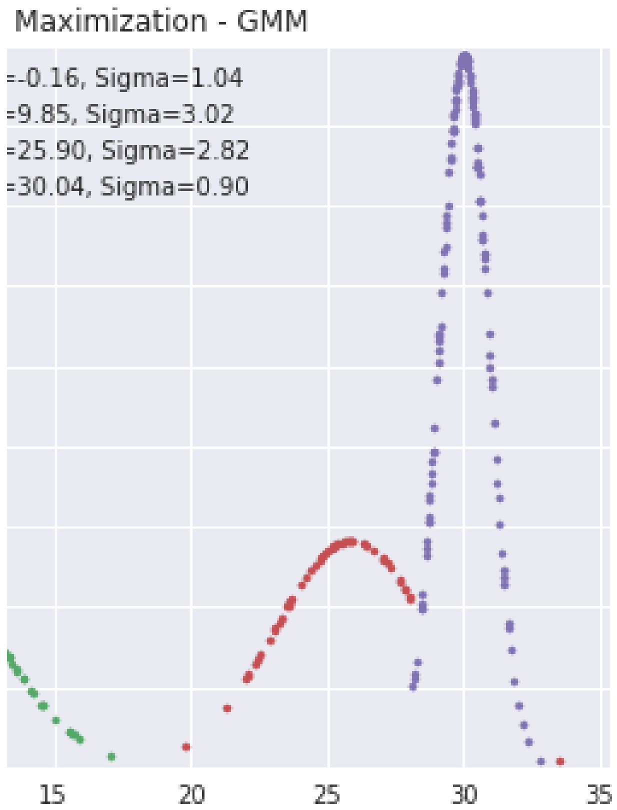
- Variance:

$$(New\ \sigma_j)^2 = \frac{1}{(New\ w_j)N}\sum_{i=1}^{N} r(x_i, j)\left(x_i - New\ \mu_j\right)^2$$

Running example







Questions

?