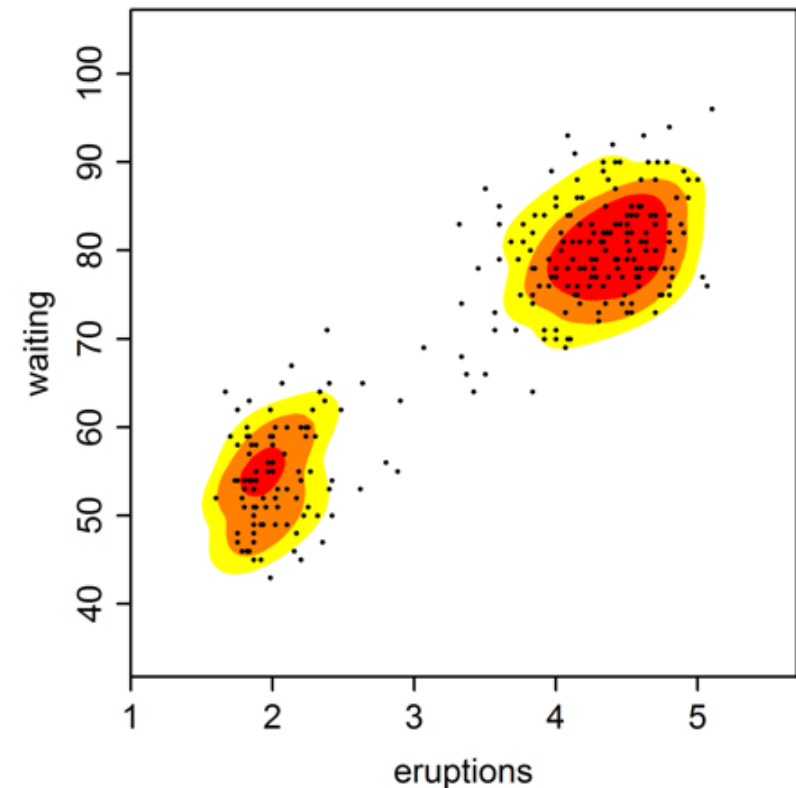


# Multivariate Normal Distributions, GMMs, EM



# Outline and some reminders

$$\operatorname{argmax}_i \{P(x|A_i)P(A_i)\}$$



Need to estimate  $P(x|A_i)$  and  $P(A_i)$

- Estimating Probabilities and Densities:  
parametric vs. non-parametric
- Estimating in 1D vs. High-Dim
- Statistical dependence and conditional independence
- Naïve Bayes classifiers
- Today: EM algorithm

# Types of Learning Tasks

- Regression
  - Given  $\{x_i, y_i\}$  find  $f$  such that  $y = f(x)$
- Classification
  - Given  $\{x_i, y_i\}$  where  $y_i \in \{0, 1\}$  for training, determine for a new  $x$  if  $x \in C_0$  or  $x \in C_1$
- **Density Estimation**
  - **Given  $\{x_i\}$  find PDF that best explains the data**
- Clustering
  - Given  $\{x_i\}$  find a partition to  $k$  subsets under some constraints

Density estimation is a useful step for classification, regression and clustering

# Recall – MAP classification

Classify an instance with observed properties  $\vec{x}$  as

$$\operatorname{argmax}_i P(\vec{x}|A_i)P(A_i)$$

# Fitting a Statistical Model – the parametric approach

- Instead of storing the histogram itself and/or using rules we fit a model (Gaussians) to the histogram and store the model parameters.
- For each class  $A_i$  we estimate:

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

- Classification: given a new instance  $x$  – we calculate the class conditional probabilities, based on the model, and look for the class that maximizes the aposteriori probabilities, based on:

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

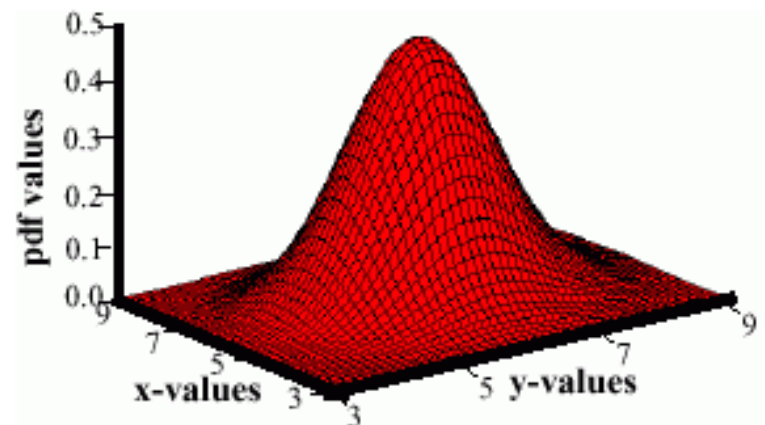
# Multivariate Normal Distributions

- A multivariate normal distribution is defined by its (multi D) pdf:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \cdot (\det(\Sigma))^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot \langle \bar{\mathbf{x}} - \bar{\boldsymbol{\mu}}, \Sigma^{-1} \cdot (\bar{\mathbf{x}} - \bar{\boldsymbol{\mu}}) \rangle\right)$$

where  $\bar{\boldsymbol{\mu}}$  represents the mean (vector) and  $\Sigma$  represents the covariance matrix.

- The covariance is always symmetric and positive semidefinite.
- How does the shape vary as a function of the covariance?
- Will be further discussed next week

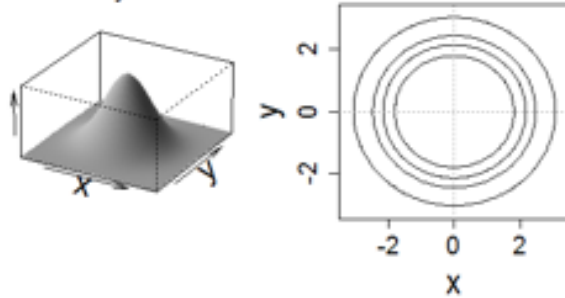


# Simple case – diagonal $\Sigma$

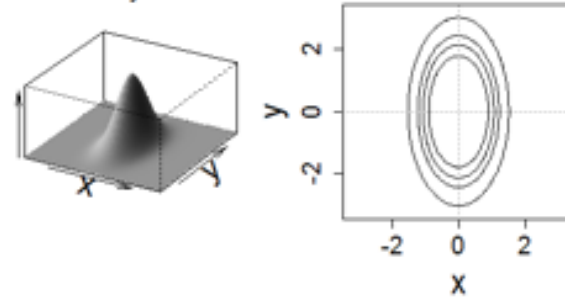
$$\begin{aligned} p(x; \mu, \Sigma) &= \frac{1}{2\pi \begin{vmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{vmatrix}^{1/2}} \exp \left( -\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \right) \\ &= \frac{1}{2\pi (\sigma_1^2 \cdot \sigma_2^2 - 0 \cdot 0)^{1/2}} \exp \left( -\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \right) \\ &= \frac{1}{2\pi \sigma_1 \sigma_2} \exp \left( -\frac{1}{2\sigma_1^2} (x_1 - \mu_1)^2 - \frac{1}{2\sigma_2^2} (x_2 - \mu_2)^2 \right) \\ &= \frac{1}{\sqrt{2\pi} \sigma_1} \exp \left( -\frac{1}{2\sigma_1^2} (x_1 - \mu_1)^2 \right) \cdot \frac{1}{\sqrt{2\pi} \sigma_2} \exp \left( -\frac{1}{2\sigma_2^2} (x_2 - \mu_2)^2 \right). \end{aligned}$$

# 2D joint Gaussians

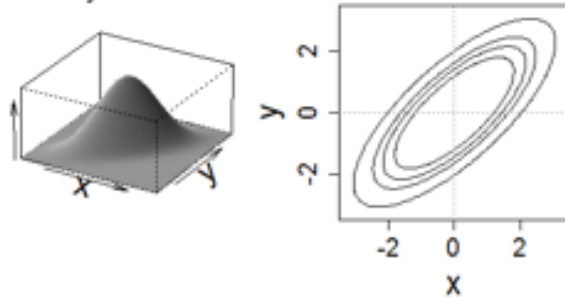
$$\sigma_x = \sigma_y, \rho = 0$$



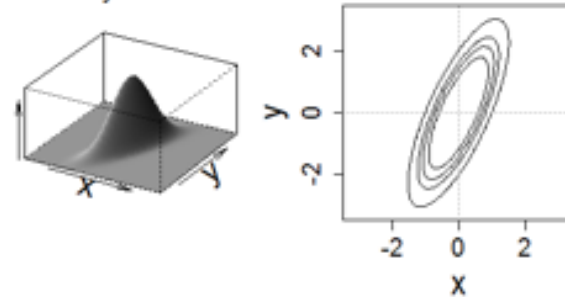
$$2\sigma_x = \sigma_y, \rho = 0$$



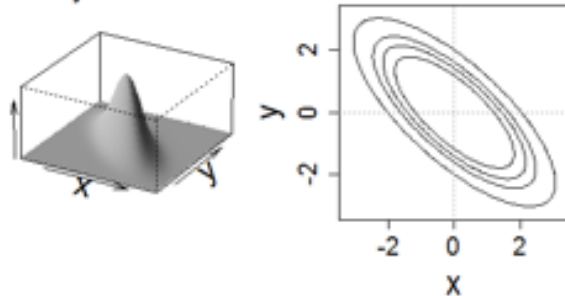
$$\sigma_x = \sigma_y, \rho = 0.75$$



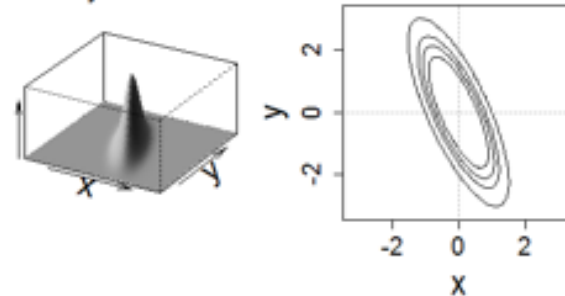
$$2\sigma_x = \sigma_y, \rho = 0.75$$



$$\sigma_x = \sigma_y, \rho = -0.75$$

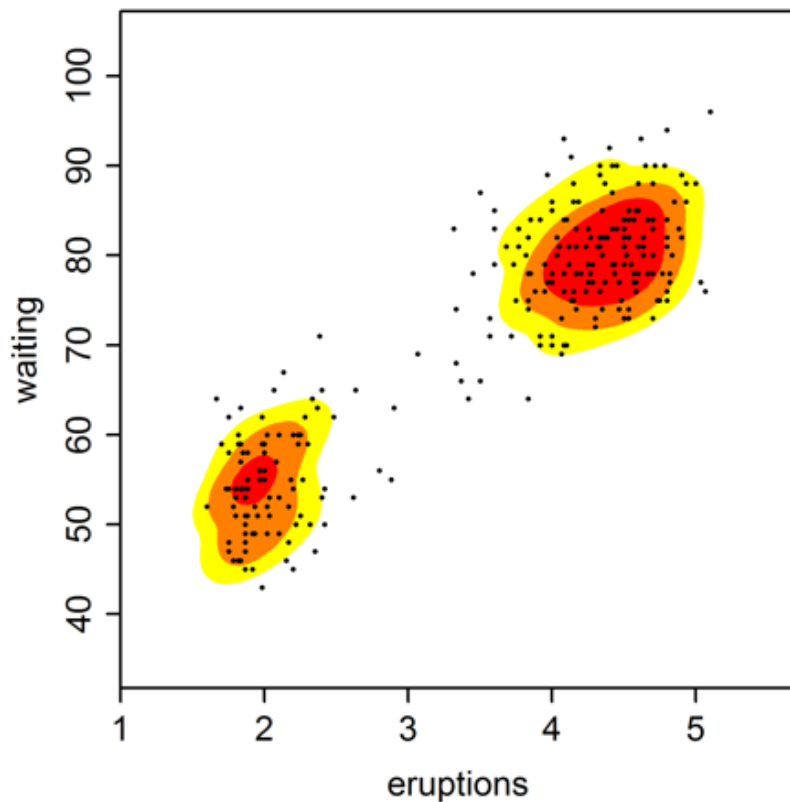


$$2\sigma_x = \sigma_y, \rho = -0.75$$



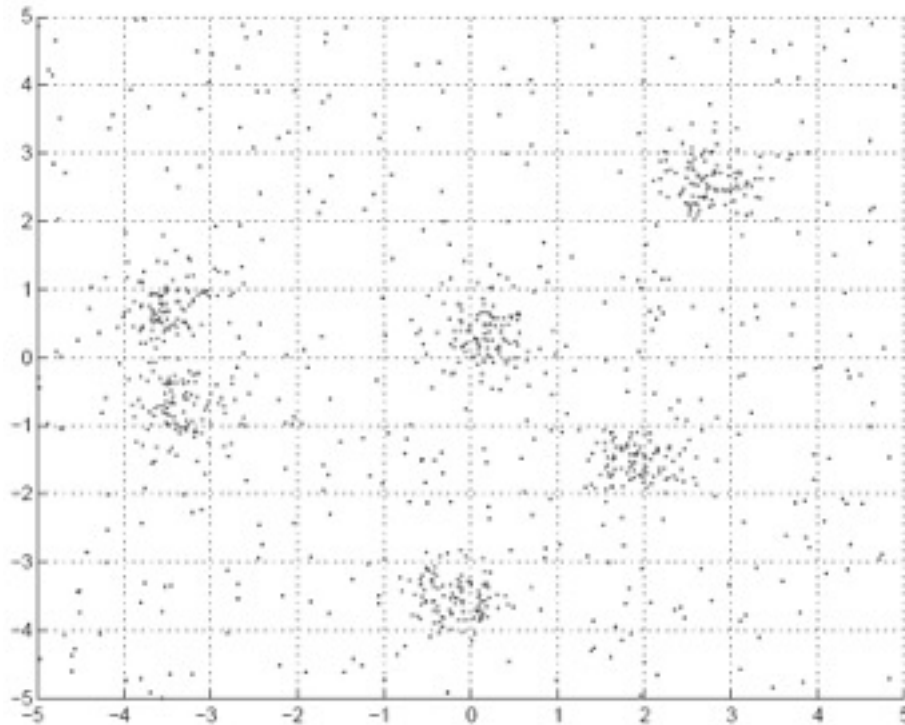


# Old Faithful Wyoming



- What can we observe?
- What would we like to know in order to understand the entire system (and make predictions)?
- Gaussian Mixtures

# Several underlying distributions



# Gaussian mixtures

X is a Gaussian Mixture RV if the density function for X's distribution is:

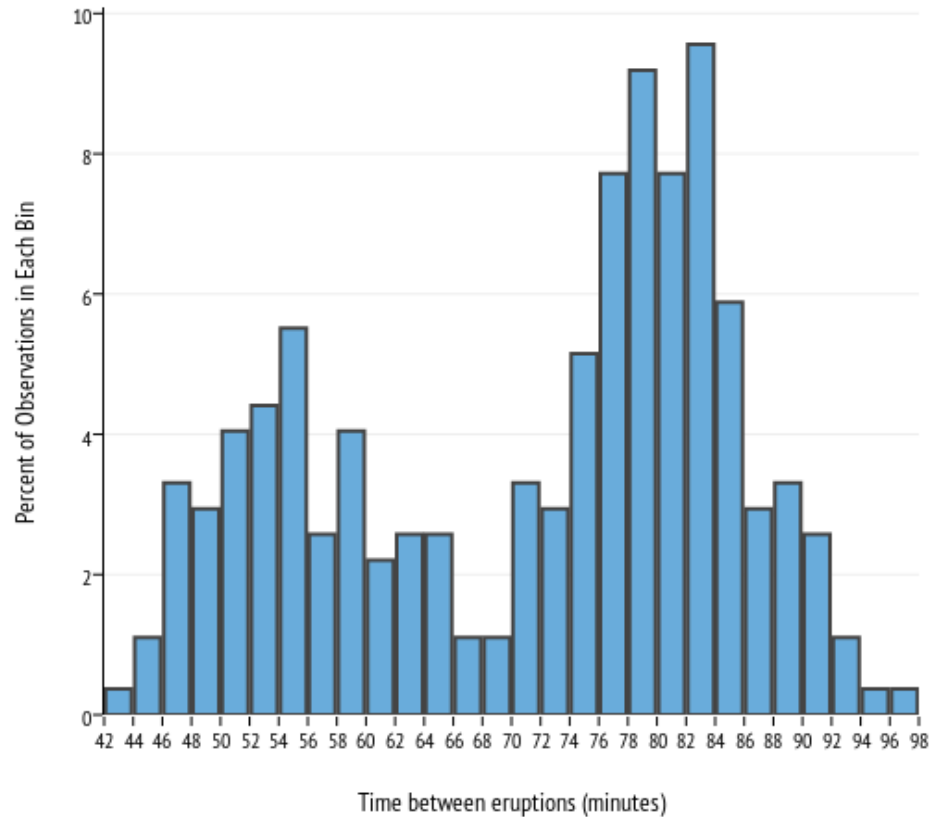
$$f(x) = \sum_{i=1}^k w_i f_i(x)$$

Where each one of the  $f_i$  density functions is a Gaussian density:

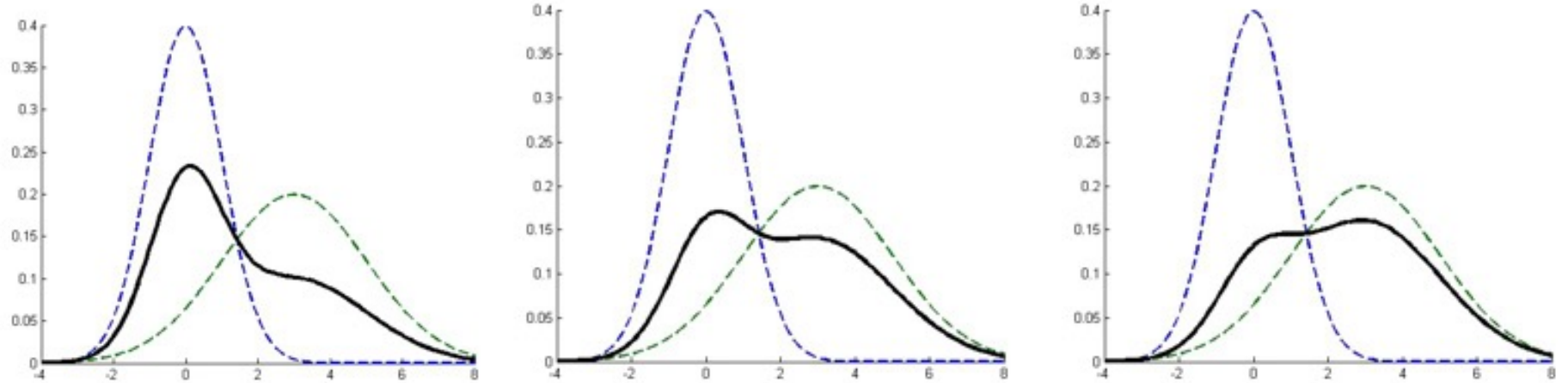
$$f_i(x) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-(x-\mu_i)^2 / 2\sigma^2}$$

# Old faithful eruptions

Old (Not So?) Faithful: A Bimodal Distribution

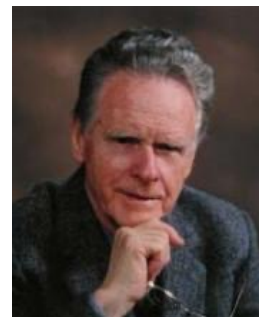


# GMD dependence on weights



# Expectation Maximization (EM) Algorithm

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



Harvard Univ, Dept of Statistics

# Basic setting in EM

- $D$  is a set of data points: **observed** data
- $\Theta$  is a parameter vector.
- EM is an iterative method for finding  $\theta_{ML}$
- It's mostly useful when
  - Calculating  $P(x \mid \theta)$  directly is hard.
  - Calculating  $P(x, z \mid \theta)$  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  $\theta$ , controls both  $X$  and  $Z$  but in  $D$  we only see the values of  $X$ ; the values of  $Z$  are hidden

# The basic EM setting - summarized

$$C = (X, Z)$$

- + C: complete data (“augmented data”)
  - + X: observable data (“incomplete” data)
  - + Z: hidden data (“missing” data)
- 
- + D: the actual observed X values (from a sample)



# 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  $p$ s and the  $w$ s.

But we don't ...

Lets see what EM can do here.

|   |            |
|---|------------|
| ● | HHHHTHHHHH |
| ● | THHHHHHHTH |
| ● | HHHHHHHTHH |
| ● | HHTHTTHHTT |
| ● | HHTHHHHHTH |
| ● | HTTHTHHHTT |
| ● | HTTHTHHHHT |
| ● | HTHHTHHHHT |

# 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  $p$ s and the  $w$ s.

But we don't ...

Lets see what EM can do here.









|   |            |
|---|------------|
|    | HHHHTHHHHH |
|    | THHHHHHHTH |
|    | HHHHHHHTHH |
|    | HHTHTTHHTT |
|    | HHTHHHHHTH |
|    | HTTHTHHHTT |
|  | HTTHTHHHHT |
|  | HTHHTHHHHT |

# Who is who?

$$C = (X, Z)$$

- + C: complete data (“augmented data”)
- + X: observable data (“incomplete” data)
- + Z: hidden data (“missing” data)

- + D: the actual observed X values (from a sample)

|   |             |
|---|-------------|
|   | HHHHTHHHHH  |
|  | THHHHHHHTH  |
|  | HHHHHHHTHH  |
|  | HHTHTTHHTT  |
|  | HHTHHHHHTH  |
|  | HTTHTHHHTT  |
|  | HTTHTHHHHT  |
|  | HTHHHTHHHHT |

# The EM algorithm

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

# EM: uncovering the coins ...

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

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

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

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

Note: we use aposteriori estimates

|   |             |     |     |
|---|-------------|-----|-----|
| ● | HHHHTHHHHH  | 0.8 | 0.2 |
| ● | THHHHHHHHTH |     |     |
| ● | HHHHHHHTHH  |     |     |
| ● | HHTHTTHHTT  |     |     |
| ● | HHTHHHHHHTH |     |     |
| ● | HTTHTHHHTT  |     |     |
| ● | HTTHTHHHHT  |     |     |
| ● | HTHHHTHHHHT |     |     |

Coin A responsibilities

Coin B responsibilities

Compute responsibilities

1

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

2

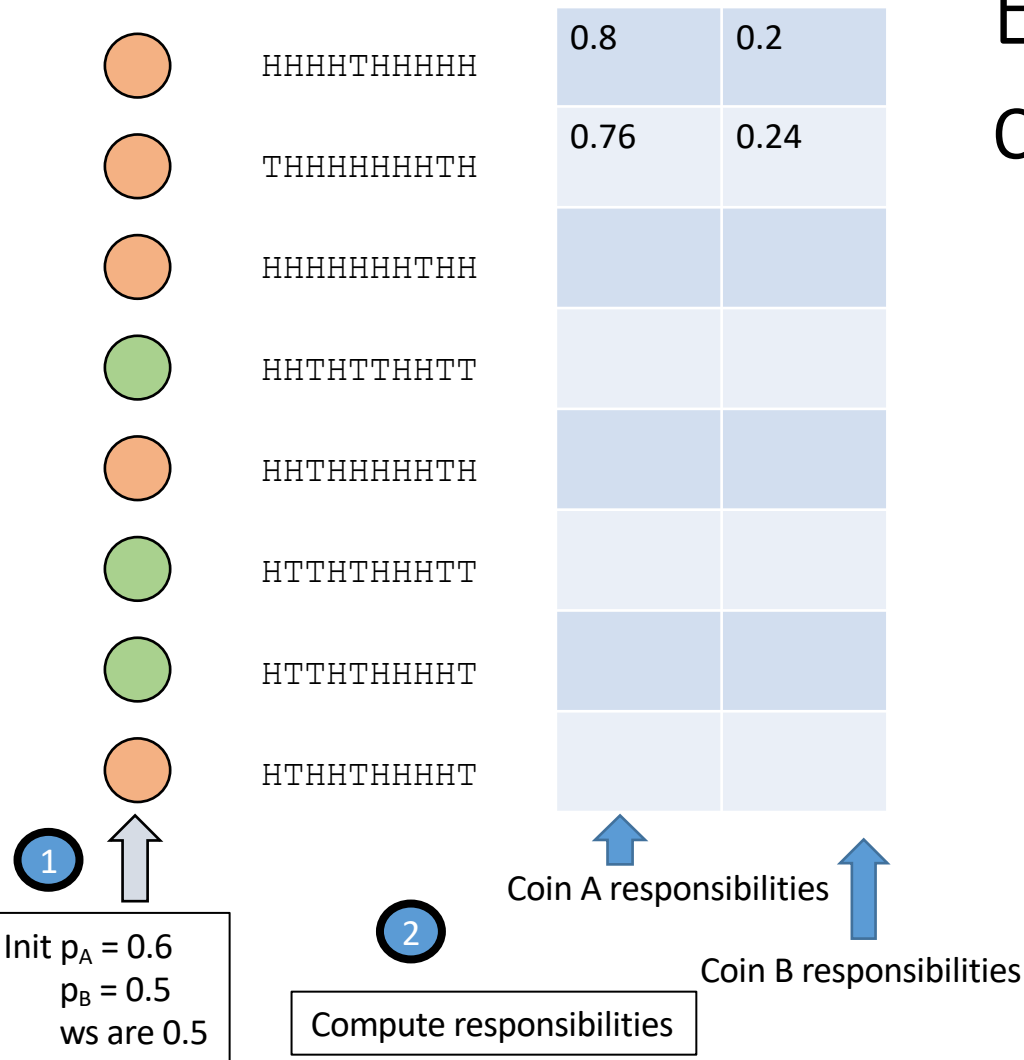
# EM: uncovering the coins ...

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

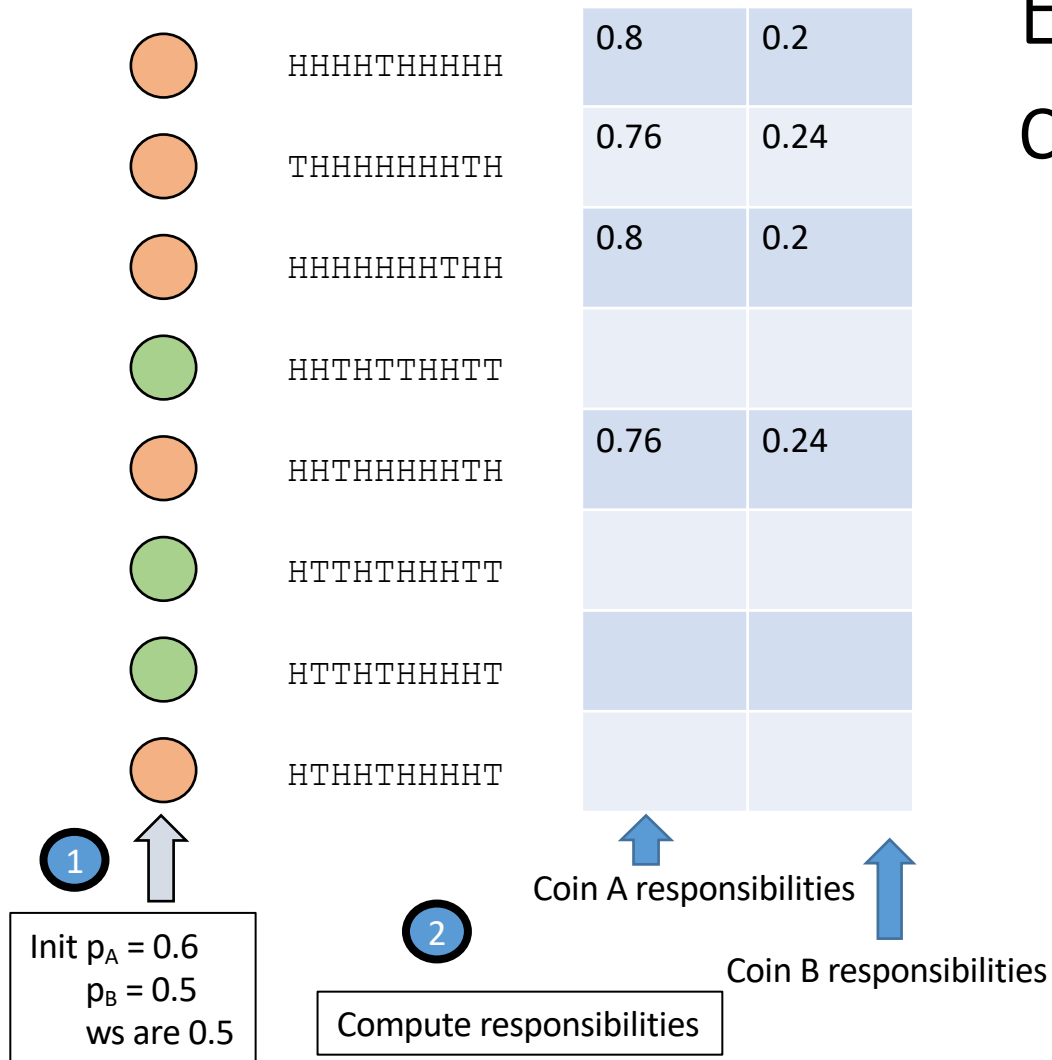
$$P_B(x_2) = w_B \binom{10}{8} 0.5^8 0.5^2 = 0.044$$

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

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



# EM: uncovering the coins ...



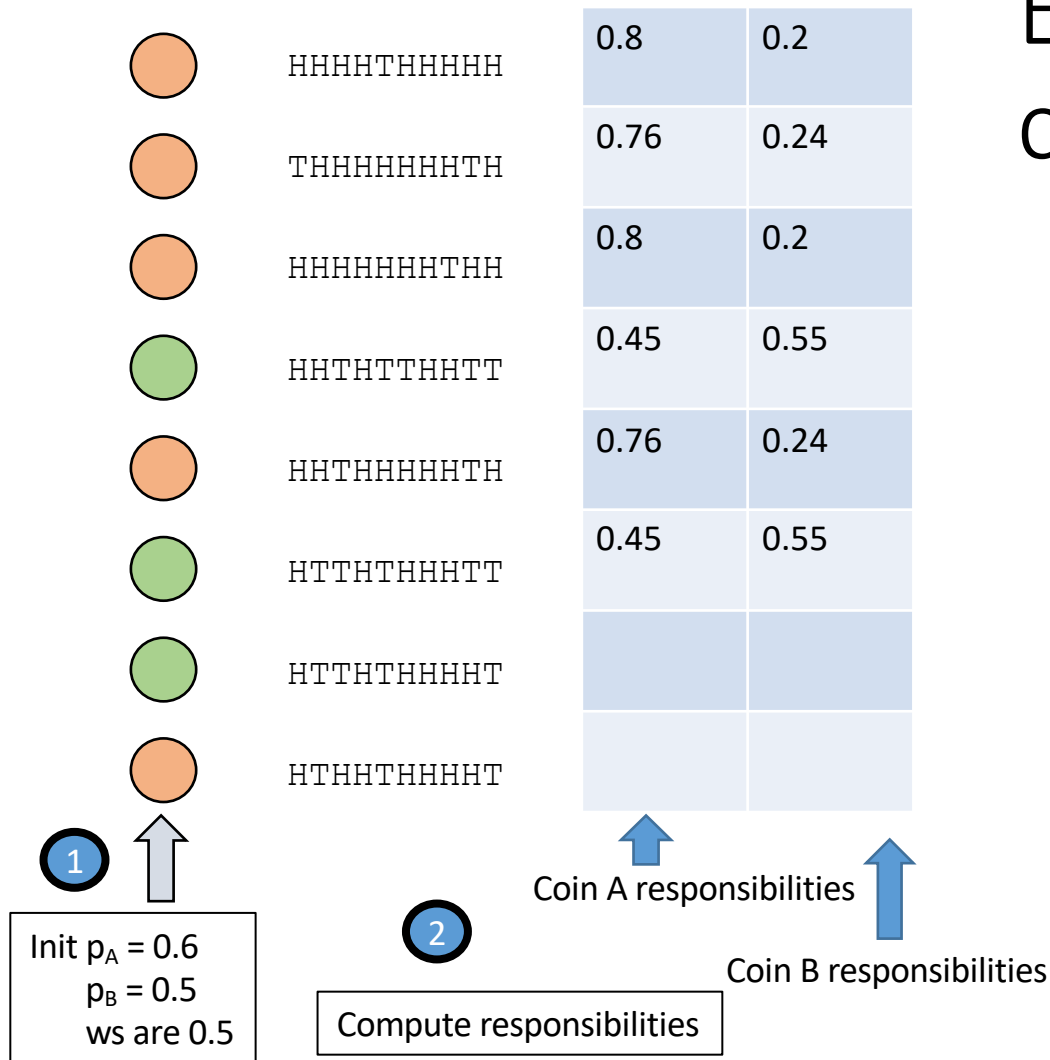
# EM: uncovering the coins ...

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

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

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

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





# EM: uncovering the coins ...

Use the responsibilities to compute new assignments:

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

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

3

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

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



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



HTHHHTHHHT

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

# EM: uncovering the coins ...



|      |      |     |
|------|------|-----|
| 0.8  | 0.2  | 0.9 |
| 0.76 | 0.24 | 0.8 |
| 0.8  | 0.2  | 0.9 |
| 0.45 | 0.55 | 0.5 |
| 0.76 | 0.24 | 0.8 |
| 0.45 | 0.55 | 0.5 |
| 0.55 | 0.45 | 0.6 |
| 0.64 | 0.36 | 0.7 |

3+4

Compute MLEs for the model parameters:

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

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

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

$$p_B = \frac{1}{3.8} \sum_{i=1}^8 r(x_i, B)v(i) = 0.48$$

1

2

3+4

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

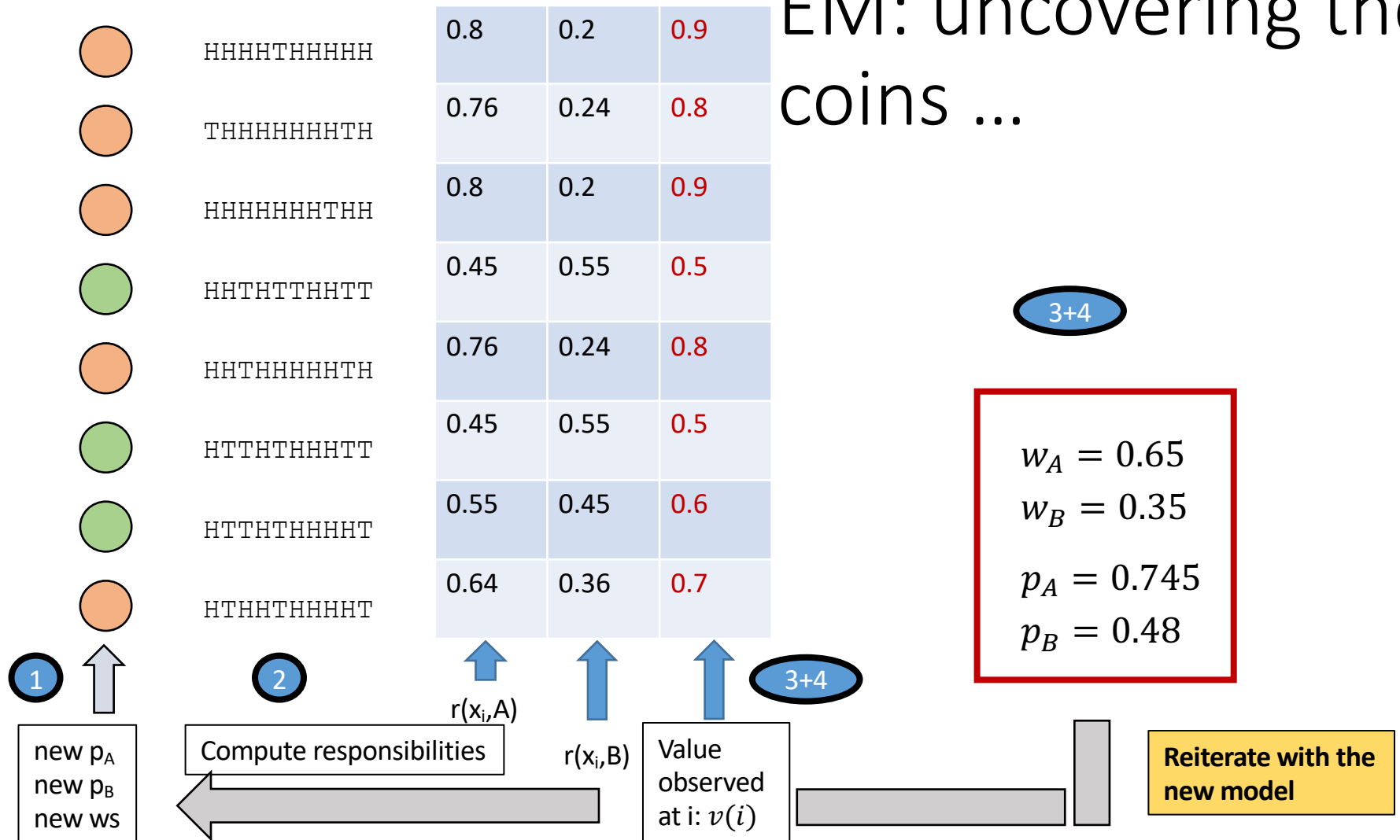
Compute responsibilities

$r(x_i, A)$

$r(x_i, B)$

Value  
 observed  
 at  $i$ :  $v(i)$

# EM: uncovering the coins ...



# 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 (using the current ws as priors)
- Use the “complete” data to update all parameters (of both Z and X|Z)

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

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

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

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

- Repeat until convergence

# EM algorithm

- A general algorithm/framework for inference where observations are dependent on a hidden intermediate
- Requires “specialization” to any given task or configuration.

# EM for GMMs

- Step 1: Expectation (E-step)

Evaluate the “responsibilities” of each data point,  $x_i$ , to each Gaussian ( $k$ ) using the current parameters (what is the posterior probability of each of the Gaussians given that data point)

- Step 2: Maximization (M-step)

Re-estimate parameters ( $w$ s,  $\mu$ s and  $\sigma$ s) using the existing “responsibilities”.

That is – every data point,  $x$ , contributes to the parameters of each Gaussian component,  $G_k$ , in proportion to its responsibility:  $r(x, G_k)$ .

# Responsibilities for Gaussian mixtures

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

# Parameter updates for Gaussian mixtures

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

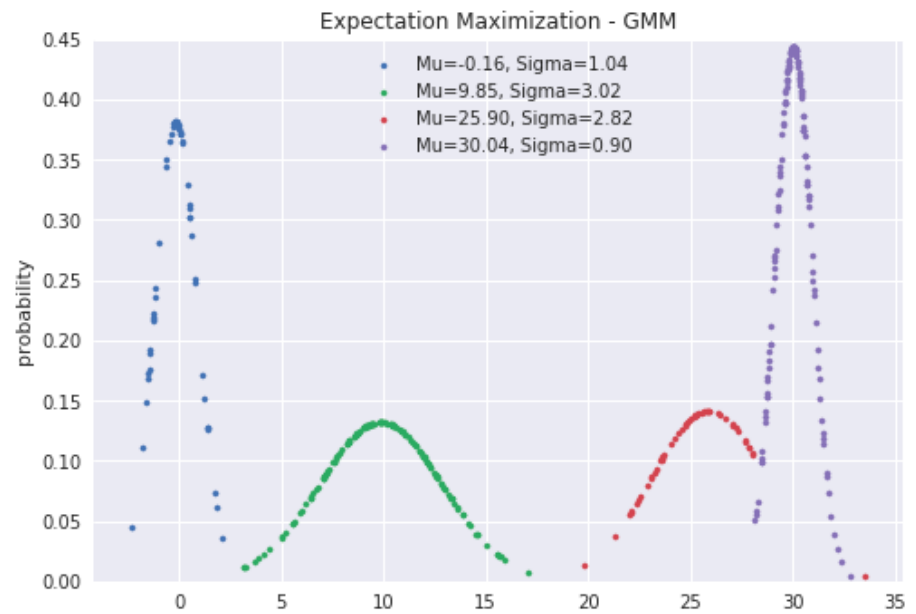
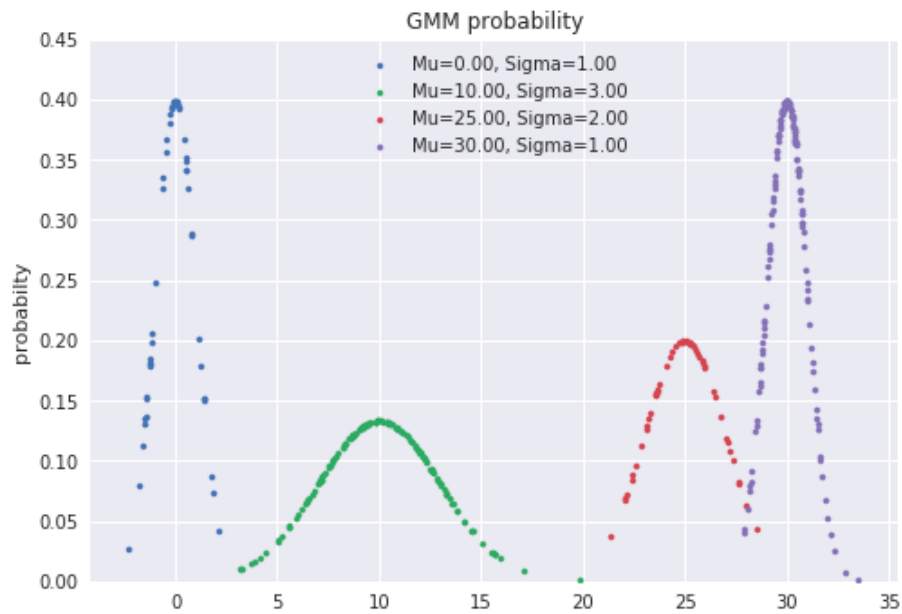


# Parameter updates for Gaussian mixtures

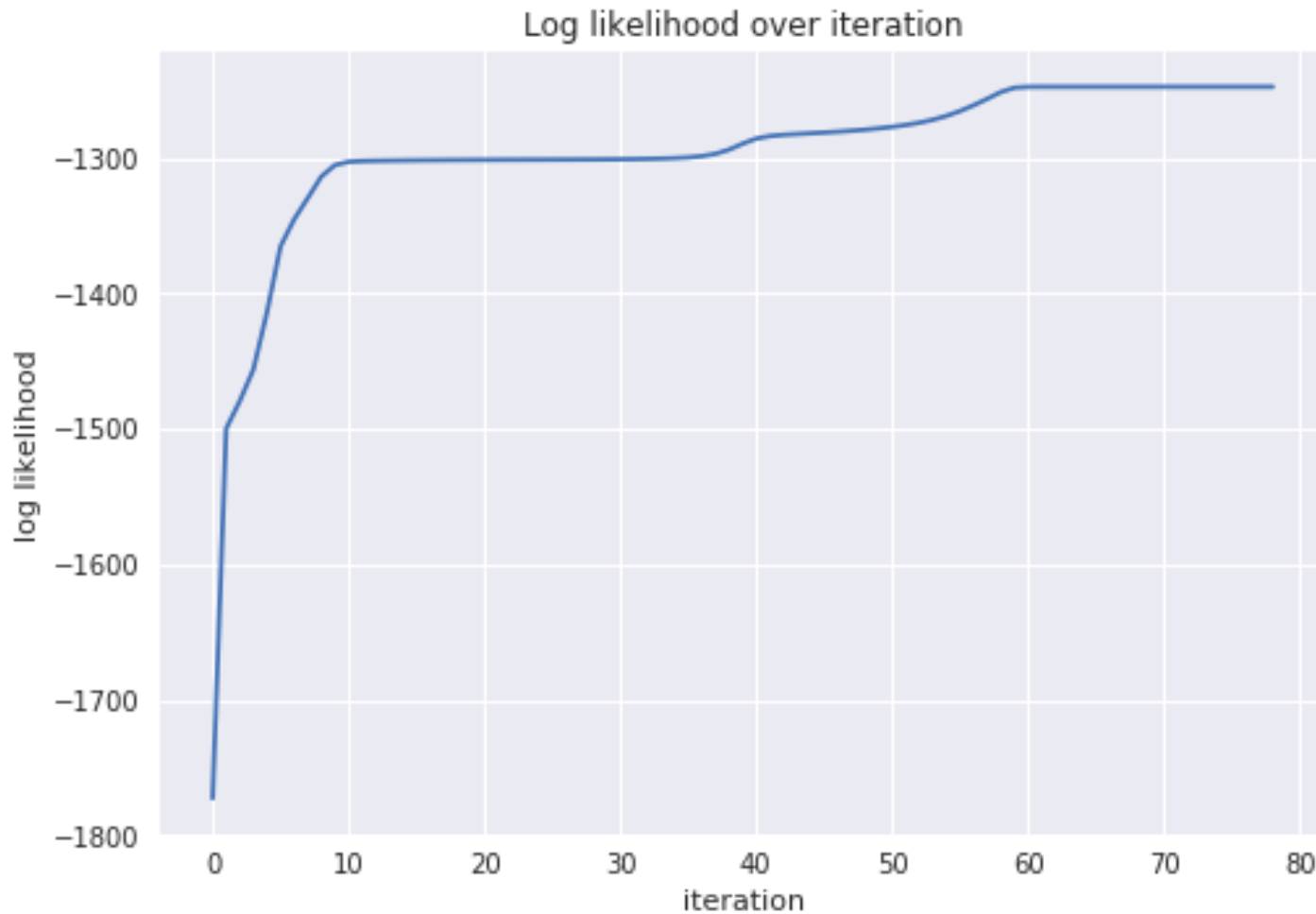
$$New \mu_k = \frac{1}{(New w_k)N} \sum_{i=1}^N r(x_i, k) x_i$$

$$(New \sigma_k)^2 = \frac{1}{(New w_k)N} \sum_{i=1}^N r(x_i, k) (x_i - New \mu_k)^2$$

# Running example



# Convergence



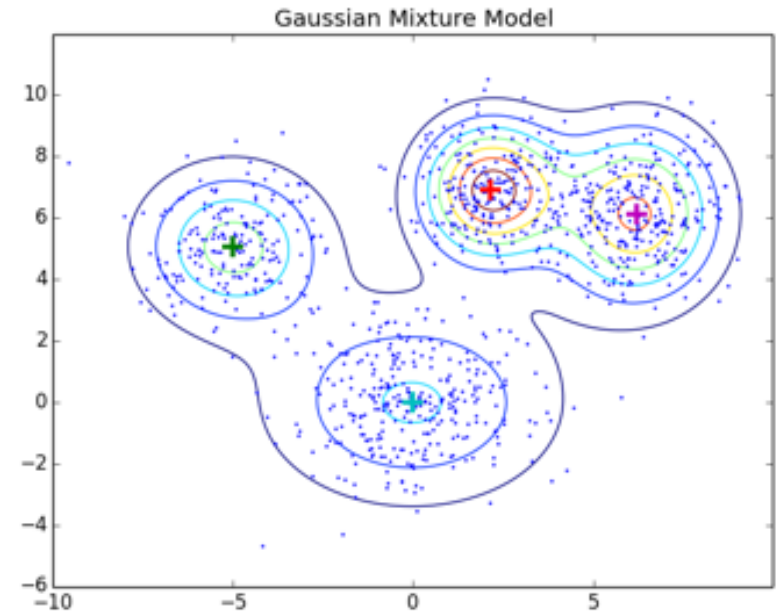
# Multidimensional GMM EM

The PDF of a d dimensional Gaussian mixture distribution has the form:

$$f(\vec{x}) = \sum_{i=1}^k w_i f_i(\vec{x})$$

where each  $f_i$  is a d dimensional Gaussian density

- How many parameters do we need to find?
  - k weights – w (actually k-1 ... )
  - d means per Gaussian
  - $\binom{d+1}{2}$  matrix entries:  
d variances +  $\binom{d}{2}$  covariances per Gaussian

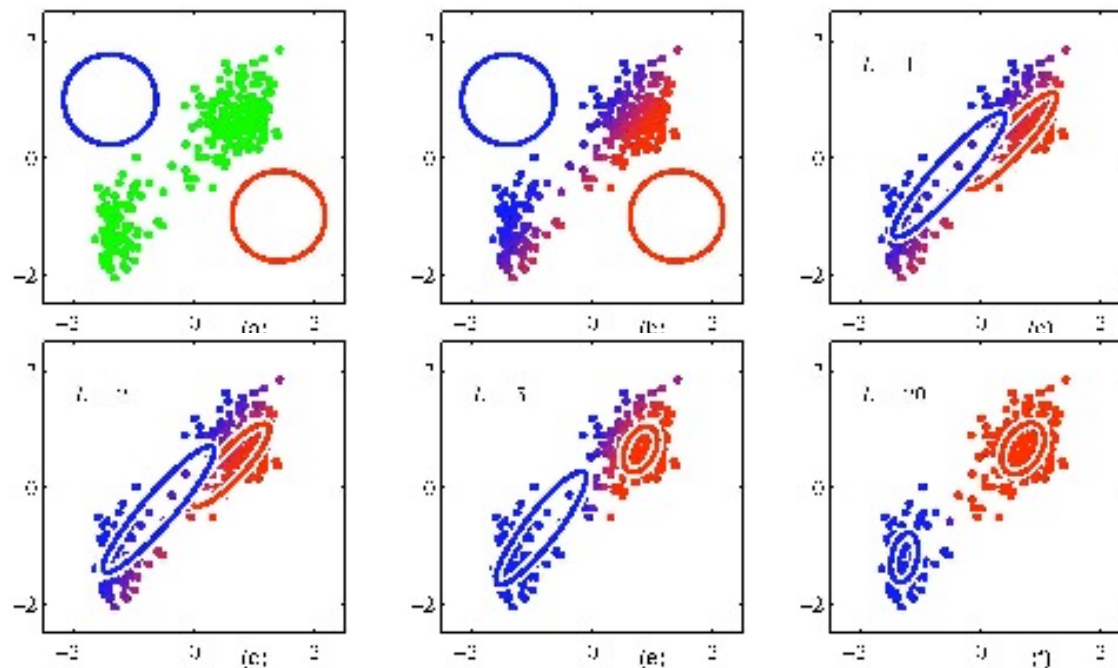


# EM algorithm

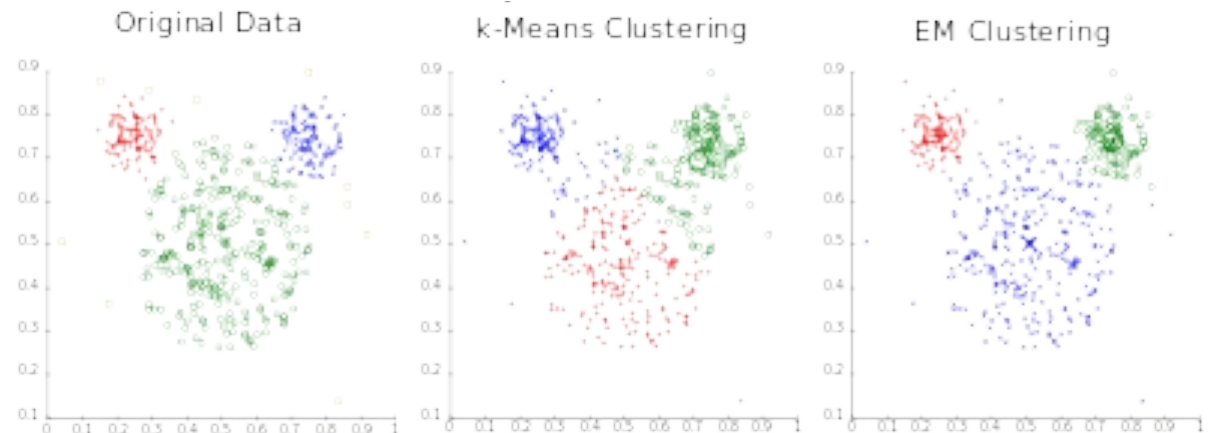
- A general algorithm/framework for inference where observations are dependent on a hidden intermediate
- Requires “specialization” to any given task or configuration.

- Consider a set of starting parameters, including the parameters of  $Z$
- Use these to compute responsibilities
- Use the “complete” data to update all parameters (of both  $Z$  and  $X|Z$ )
- Repeat until convergence

# Visual example of EM for 2D Gauss mixtures



# EM application examples



Mickey data

- Clustering  
(we will come back to this)
- Modelling  
(including evolution and other processes)
- Prediction/classification/regression
- Outlier detection  
(predictive maintenance)

# General form of EM

- Given a joint distribution over observed and latent variables:  $p(X, Z|\theta)$
- Want to find  $\theta$  that maximizes:

$$p(X|\theta)$$

1. Initialize parameters  $\theta^{old}$

2. E Step: Evaluate

$$p(Z|X, \theta^{old})$$

3. M-Step: Re-estimate parameters (based on expectation of complete-data log likelihood)

$$\theta^{new} = \operatorname{argmax}_{\theta} \sum_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta)$$

4. Check for convergence of parameters or likelihood

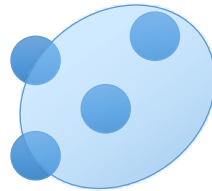


# What EM will NOT do

Determine structure of the model

# components

graph structure



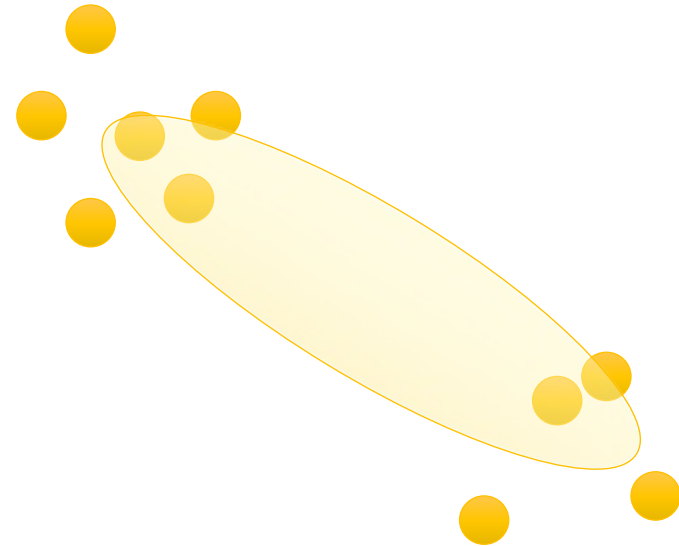
Guarantee global maximum

Always have nice closed-form update equations

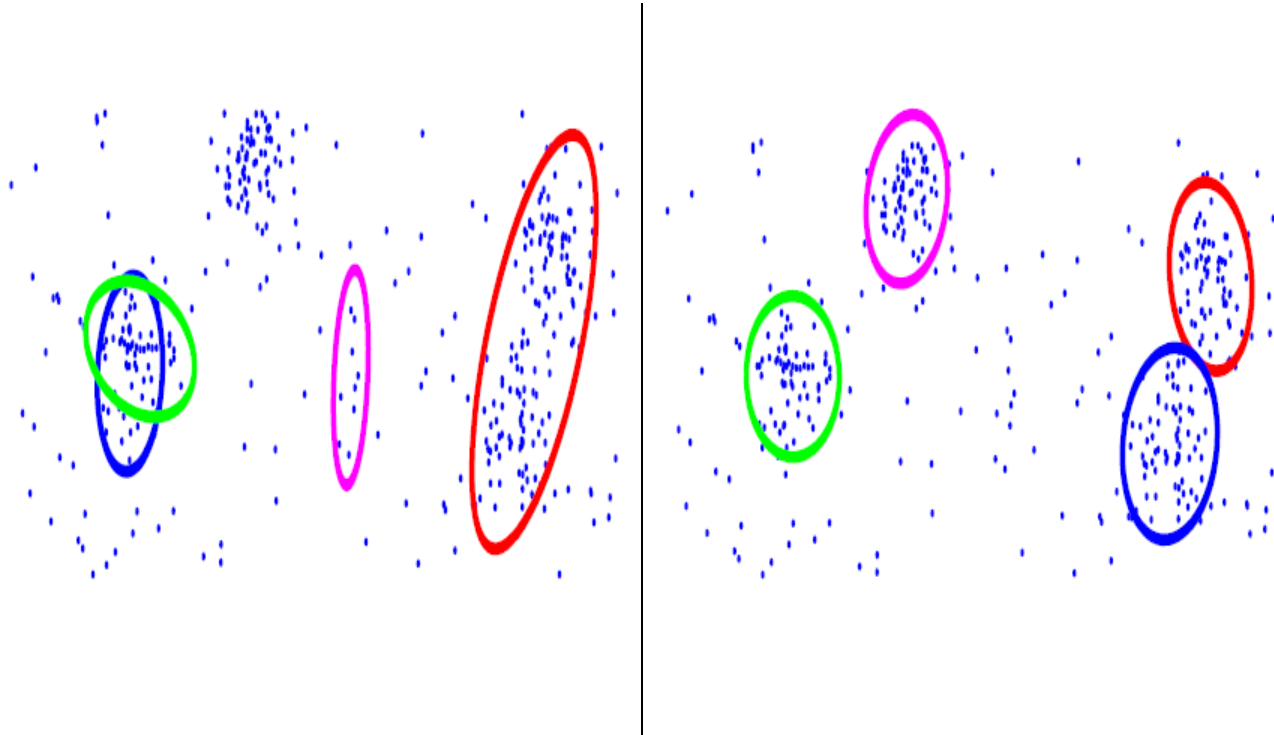
need to develop per case

Avoid computational problems

we may want to use sampling  
may be parallelizable



# Local minima



# EM pros and cons

- Convergence: in every iteration of the EM algorithm the attained likelihood is improved over the previous round.
  - EM is appropriate for (almost) any family of models and for any number of parameters.
- Convergence can be very slow on some instances and is intimately related to the amount of missing information.
  - Like any learning approach, we work on the training data. It is important to control against overfitting. (e.g: number of values of  $Z$ )
  - Model assumptions are user determined. Not principally determined.
  - No global optimum guarantee  
(it could get stuck at local maxima, saddle points, etc)
    - ➔ The initial values are important and several sets should be used

# Summary

- EM: an MLE approach to learning more complicated probability model structures.
- Specifically – structures that involve a hidden layer
- The EM process for two coins
- GMMs
- The EM process for GMMs