

# Learning in Graphical Models

- Problem Dimensions
  - Model
    - Bayes Nets
    - Markov Nets
  - Structure
    - Known
    - Unknown (structure learning)
  - Data
    - Complete
    - **Incomplete (missing values or hidden variables)**

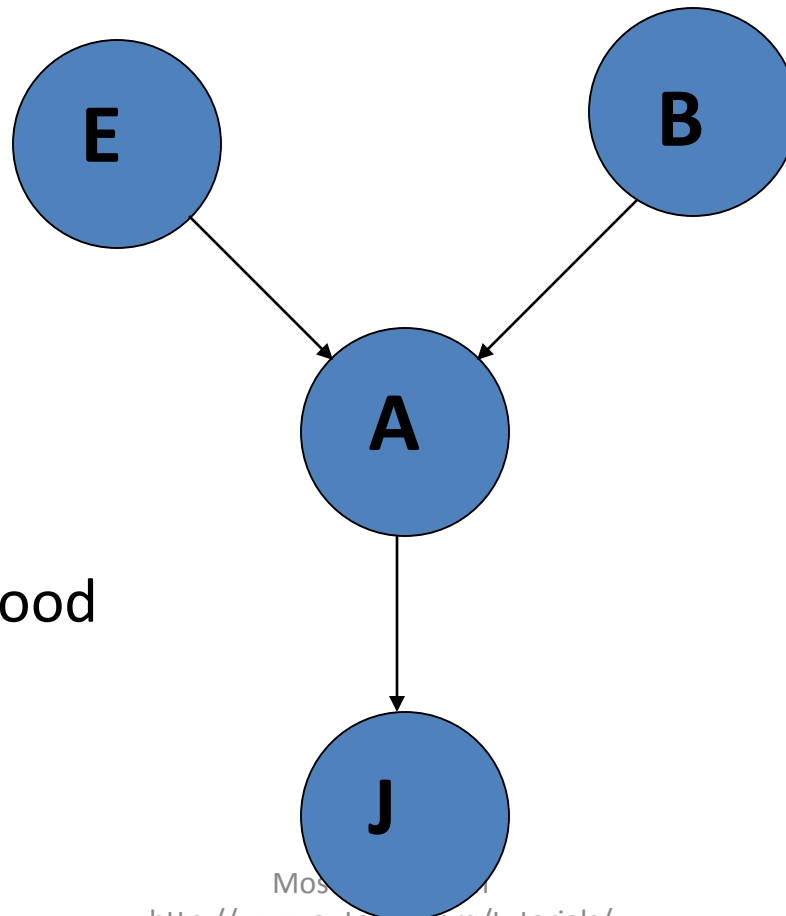
# Outline

- Objective
- Simple example
- Complex example

# Objective

- Learning with missing/unobservable data

E	B	A	J
1	1	1	1
1	0	1	1
0	0	0	0
...			



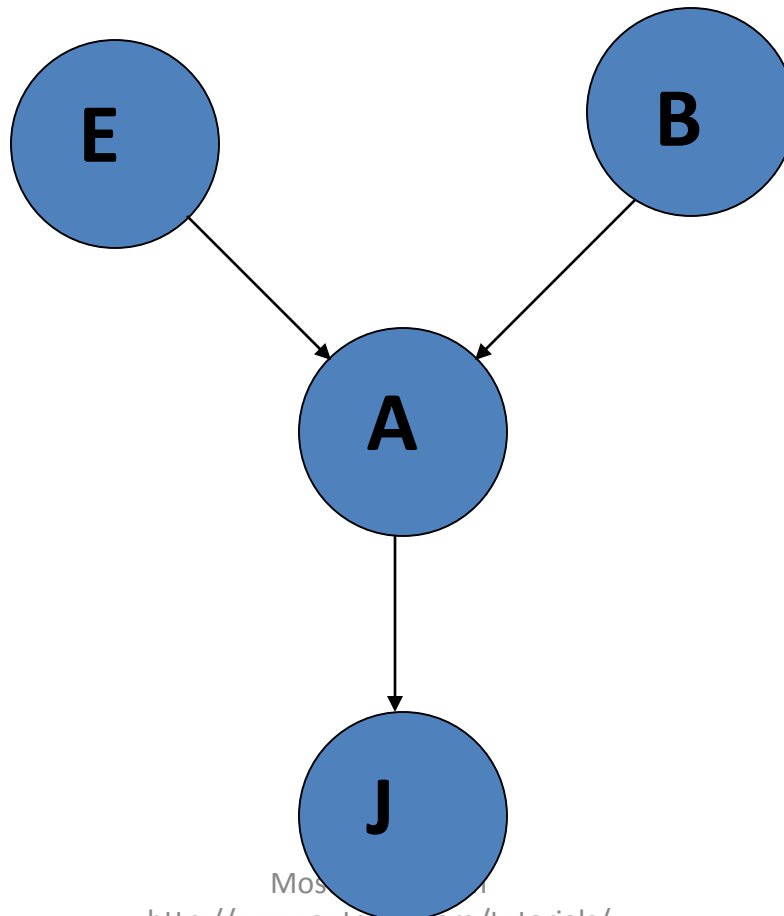
Maximum likelihood

# Objective

- Learning with missing/unobservable data

E	B	A	J
1	1	?	1
1	0	?	1
0	0	?	0
...			

Optimize what?



# Outline

- Objective
- **Simple example**
- Complex example

# Simple example

Let events be "grades in a class"

$$w_1 = \text{Gets an A} \quad P(A) = \frac{1}{2}$$

$$w_2 = \text{Gets a B} \quad P(B) = \mu$$

$$w_3 = \text{Gets a C} \quad P(C) = 2\mu$$

$$w_4 = \text{Gets a D} \quad P(D) = \frac{1}{2} - 3\mu$$

(Note  $0 \leq \mu \leq 1/6$ )

Assume we want to estimate  $\mu$  from data. In a given class there were

a A's  
b B's  
c C's  
d D's

A	B	C	D
14	6	9	10

What's the maximum likelihood estimate of  $\mu$  given  $a, b, c, d$  ?

# Maximize likelihood

$$P(A) = 1/2 \quad P(B) = \mu \quad P(C) = 2\mu \quad P(D) = 1/2 - 3\mu$$

$$P(a, b, c, d | \mu) = K(1/2)^a(\mu)^b(2\mu)^c(1/2 - 3\mu)^d$$

$$\log P(a, b, c, d | \mu) = \log K + a \log 1/2 + b \log \mu + c \log 2\mu + d \log (1/2 - 3\mu)$$

$$\text{FOR MAX LIKE } \mu, \text{ SET } \frac{\partial \text{LogP}}{\partial \mu} = 0$$

$$\frac{\partial \text{LogP}}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0$$

$$\text{Gives max like } \mu = \frac{b + c}{6(b + c + d)}$$

A	B	C	D
14	6	9	10

# Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) =  $h$

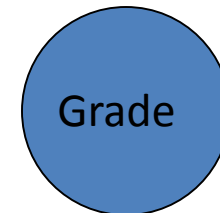
Number of C's =  $c$

Number of D's =  $d$

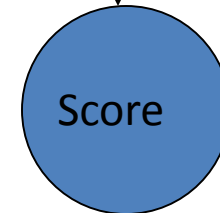
What is the max. like estimate of  $\mu$  now?

REMEMBER
$P(A) = 1/2$
$P(B) = \mu$
$P(C) = 2\mu$
$P(D) = 1/2 - 3\mu$

Hidden



Observable





# Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) =  $h$

Number of C's =  $c$

Number of D's =  $d$

What is the max. like estimate of  $\mu$  now?

We can answer this question circularly:

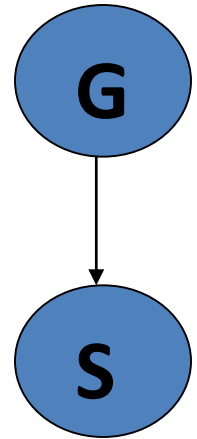
REMEMBER

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

$$P(B) = \mu$$

$$P(C) = 2\mu$$

$$P(D) = \frac{1}{2} - 3\mu$$



## MAXIMIZATION

If we know the            values of  $a$  and  $b$  we could compute the maximum likelihood value of  $\mu$

$$\mu = \frac{b + c}{6(b + c + d)}$$

# Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) =  $h$

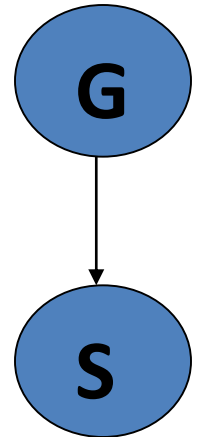
Number of C's =  $c$

Number of D's =  $d$

What is the max. like estimate of  $\mu$  now?

We can answer this question circularly:

REMEMBER  
 $P(A) = 1/2$   
 $P(B) = \mu$   
 $P(C) = 2\mu$   
 $P(D) = 1/2 - 3\mu$



## EXPECTATION

If we know the value of  $\mu$  we could compute the expected value of  $a$  and  $b$

Since the ratio  $a:b$  should be the same as the ratio  $1/2 : \mu$

$$a = \frac{1/2}{1/2 + \mu} h \quad b = \frac{\mu}{1/2 + \mu} h$$

## MAXIMIZATION

If we know the   values of  $a$  and  $b$  we could compute the maximum likelihood value of  $\mu$  **under those expected values**

$$\mu = \frac{b + c}{6(b + c + d)}$$

# EM for our example

REMEMBER

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

$$P(B) = \mu$$

$$P(C) = 2\mu$$

$$P(D) = \frac{1}{2} - 3\mu$$

We begin with a guess for  $\mu$

We iterate between EXPECTATION and MAXIMALIZATION to improve our estimates of  $\mu$  and  $a$  and  $b$ .

Define  $\mu(t)$  the estimate of  $\mu$  on the  $t$ 'th iteration


$b(t)$  the estimate of  $b$  on  $t$ 'th iteration

$\mu(0) =$  initial guess


$$b(t) = \frac{\mu(t)h}{\frac{1}{2} + \mu(t)} = E[b \mid \mu(t)]$$

$$\mu(t+1) = \frac{b(t) + c}{6(b(t) + c + d)}$$

$=$  max like est of  $\mu$  given  $b(t)$



**E-step**



**M-step**

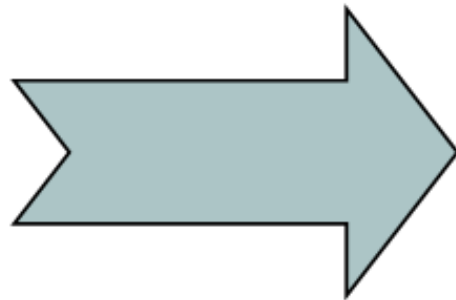
# EM Convergence

- Convergence proof based on fact that  $\text{Prob}(\text{data} \mid \mu)$  must increase or remain same between each iteration [NOT OBVIOUS]
  - But it can never exceed 1 [OBVIOUS]
- So it must therefore converge [OBVIOUS]

---

In our example,  
suppose we had

$h = 20$   
 $c = 10$   
 $d = 10$   
 $\mu(0) = 0$



t	$\mu(t)$	b(t)
0	0	0
1	0.0833	2.857
2	0.0937	3.158
3	0.0947	3.185
4	0.0948	3.187
5	0.0948	3.187
6	0.0948	3.187

# Generalization

- $X$ : observable data (score = {h, c, d})
- $z$ : missing data (grade = {a, b})
- $\theta$ : model parameters to estimate ( $\mu$ )
  
- E: given  $\theta$ , compute the expectation of counts of  $z$
- M: use  $z$   $c\mathcal{P}(X, z|\theta)$  in E step, maximize the likelihood with respect to  $\theta$

# Outline

- Objective
- Simple example
- **Complex example**

# Gaussian Mixtures

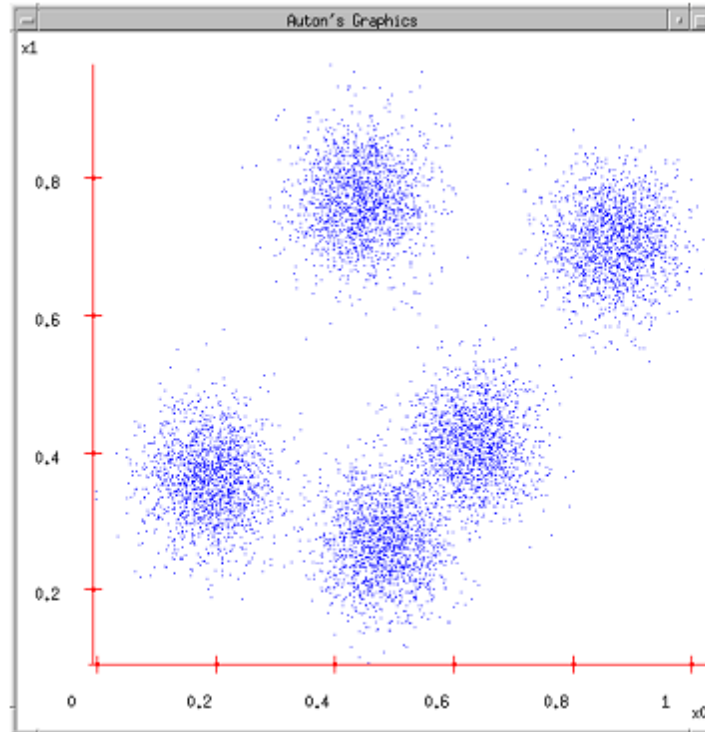
"I've got data from  $k$  classes. Each class produces observations with a normal distribution and variance  $\sigma^2 I$ . Standard simple multivariate gaussian assumptions. I can tell you all the  $P(w_i)$ 's."

"I need a maximum likelihood estimate of the  $\mu_i$ 's."

"There's just one thing. None of the data are labeled. I have datapoints, but I don't know what class they're from (any of them!)"

# Gaussian Mixtures

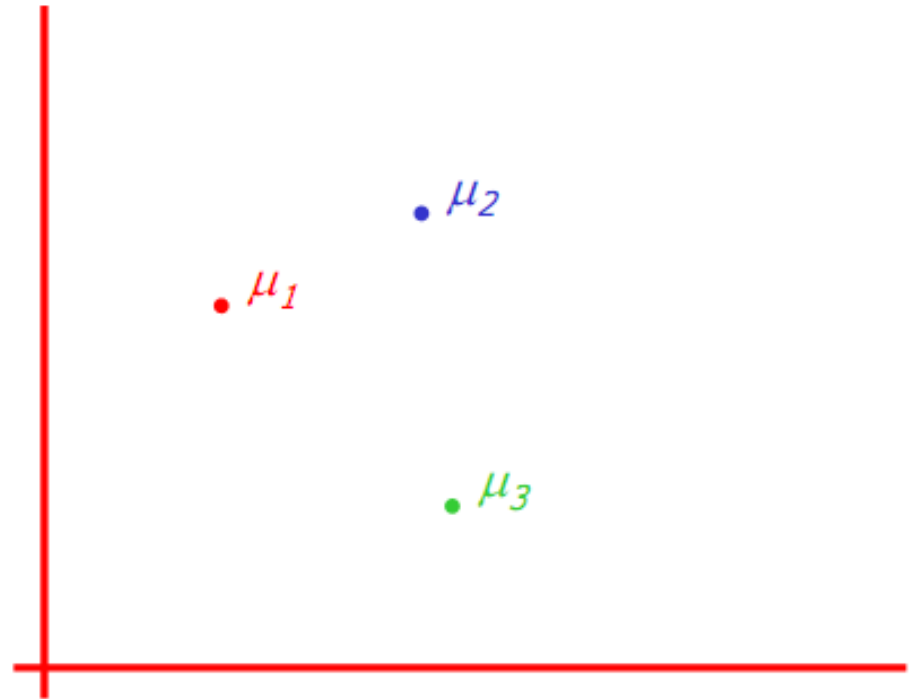
- Know
  - Data
  - $\sigma^2 \mathbf{I}$
  - $P(w_i)$
- Don't know
  - Data label
- Objective
  - estimate of the  $\mu_i$ 's





# The GMM assumption

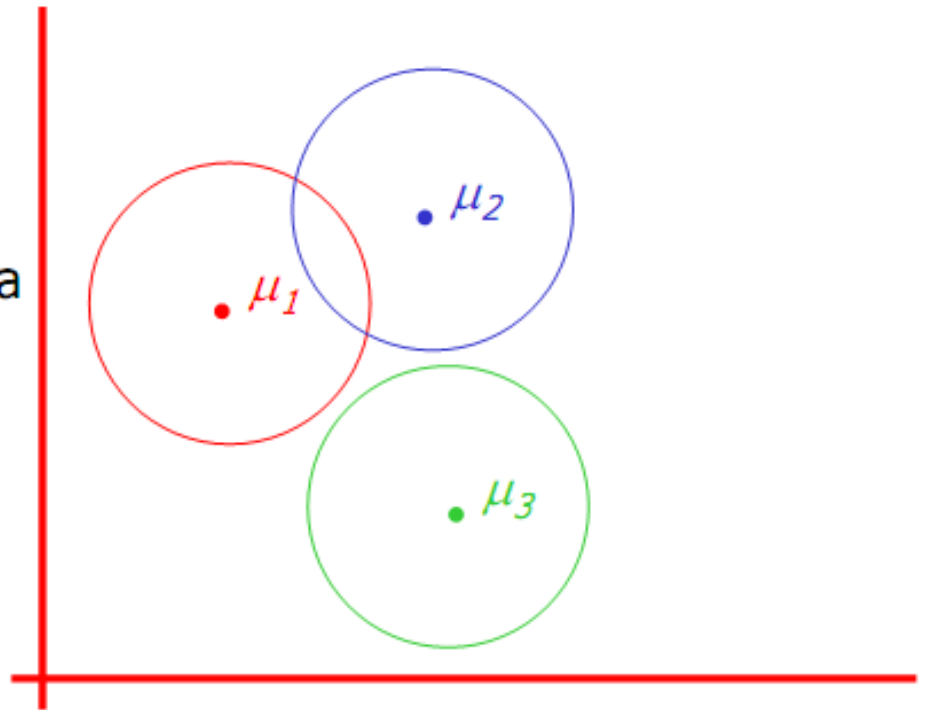
- There are  $k$  components. The  $i$ 'th component is called  $\omega_i$
- Component  $\omega_i$  has an associated mean vector  $\mu_i$



# The GMM assumption

- There are  $k$  components. The  $i$ 'th component is called  $\omega_i$
- Component  $\omega_i$  has an associated mean vector  $\mu_i$
- Each component generates data from a Gaussian with mean  $\mu_i$  and covariance matrix  $\sigma^2 \mathbf{I}$

Assume that each datapoint is generated according to the following recipe:

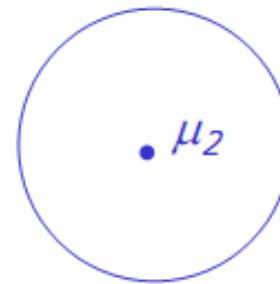


# The GMM assumption

- There are  $k$  components. The  $i$ 'th component is called  $\omega_i$
- Component  $\omega_i$  has an associated mean vector  $\mu_i$
- Each component generates data from a Gaussian with mean  $\mu_i$  and covariance matrix  $\sigma^2 \mathbf{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random. Choose component  $i$  with probability  $P(\omega_i)$ .

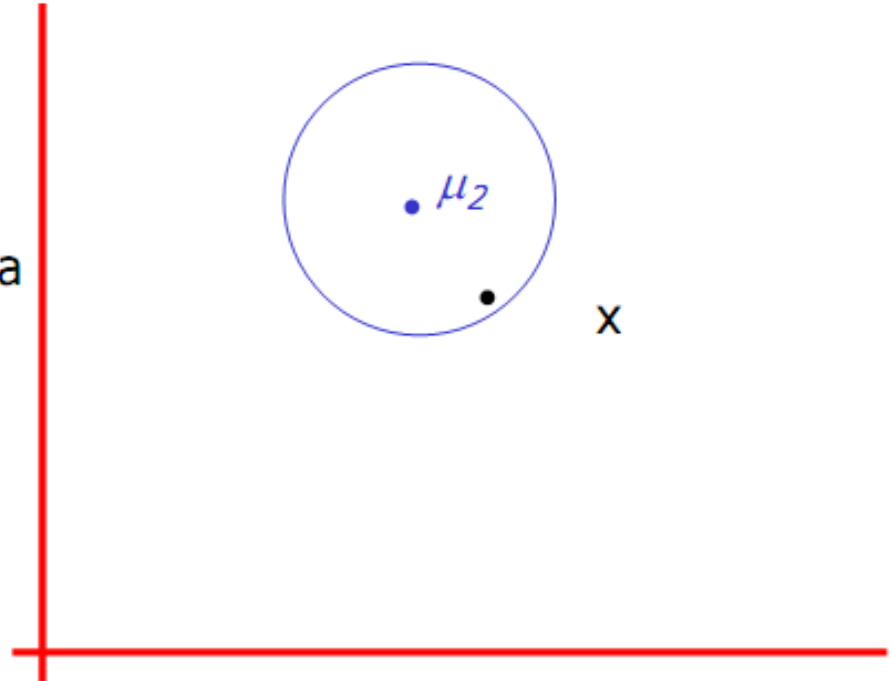


# The GMM assumption

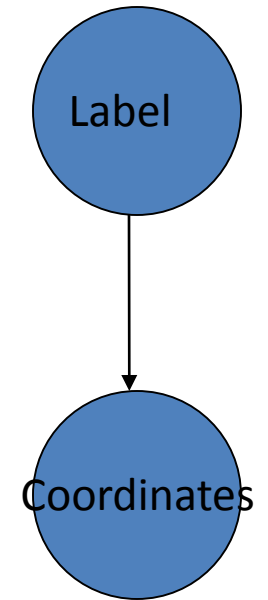
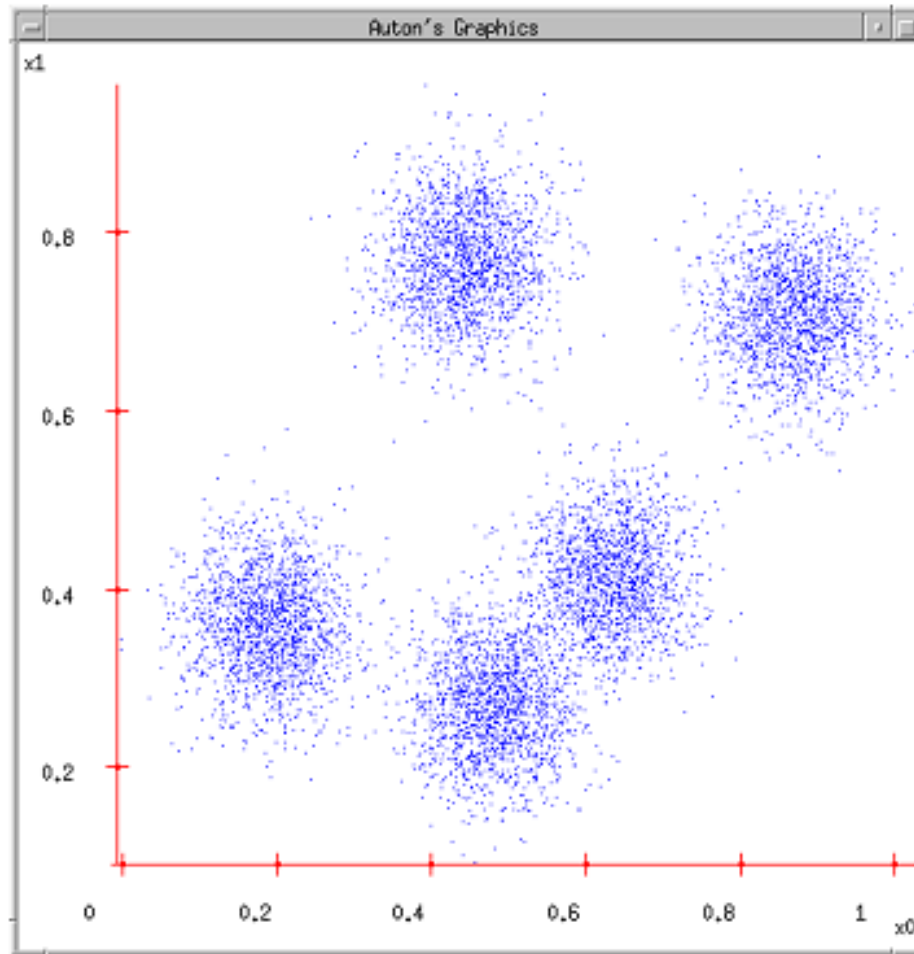
- There are  $k$  components. The  $i$ 'th component is called  $\omega_i$
- Component  $\omega_i$  has an associated mean vector  $\mu_i$
- Each component generates data from a Gaussian with mean  $\mu_i$  and covariance matrix  $\sigma^2 \mathbf{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random. Choose component  $i$  with probability  $P(\omega_i)$ .
2. Datapoint  $\sim N(\mu_i, \sigma^2 \mathbf{I})$



# The data generated



# Computing the likelihood

Remember:

We have unlabeled data  $x_1 x_2 \dots x_R$

We know there are  $k$  classes

We know  $P(w_1) P(w_2) P(w_3) \dots P(w_k)$

We don't know  $\mu_1 \mu_2 \dots \mu_k$

We can write  $P(\text{data} \mid \mu_1 \dots \mu_k)$

$$= p(x_1 \dots x_R \mid \mu_1 \dots \mu_k)$$

$$= \prod_{i=1}^R p(x_i \mid \mu_1 \dots \mu_k)$$

$$= \prod_{i=1}^R \sum_{j=1}^k p(x_i \mid w_j, \mu_1 \dots \mu_k) P(w_j)$$

$$= \prod_{i=1}^R \sum_{j=1}^k K \exp\left(-\frac{1}{2\sigma^2} (x_i - \mu_j)^2\right) P(w_j)$$

Most slides from

<http://www.autonlab.org/tutorials/>

# EM for GMMs

For Max likelihood we know  $\frac{\partial}{\partial \mu_i} \log \text{Pr ob}(\text{data} | \mu_1 \dots \mu_k) = 0$

Some wild' n' crazy algebra turns this into : "For Max likelihood, for each j,

$$\mu_j = \frac{\sum_{i=1}^R P(w_j | x_i, \mu_1 \dots \mu_k) x_i}{\sum_{i=1}^R P(w_j | x_i, \mu_1 \dots \mu_k)}$$

This is n nonlinear equations in  $\mu_j$ 's."

# EM for GMMs

For Max likelihood we know  $\frac{\partial}{\partial \mu_i} \log \text{Pr ob}(\text{data} | \mu_1 \dots \mu_k) = 0$

Some wild 'n' crazy algebra turns this into : "For Max likelihood, for each j,

$$\mu_j = \frac{\sum_{i=1}^R P(w_j | x_i, \mu_1 \dots \mu_k) x_i}{\sum_{i=1}^R P(w_j | x_i, \mu_1 \dots \mu_k)}$$

This is n nonlinear equations in  $\mu_j$ 's."

If, for each  $x_i$  we knew that for each  $w_j$  the prob that  $x_i$  was in class  $w_j$  is  $P(w_j | x_i, \mu_1 \dots \mu_k)$  Then... we would easily compute  $\mu_j$ .

If we knew each  $\mu_j$  then we could easily compute  $P(w_j | x_i, \mu_1 \dots \mu_j)$  for each  $w_j$  and  $x_i$ .



# EM for GMMs

Iterate. On the  $t$ 'th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \dots \mu_c(t) \}$$

$p_i(t)$  is shorthand for estimate of  $P(\omega_i)$  on  $t$ 'th iteration

E-step

Compute "expected" classes of all datapoints for each class

$$P(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t) P(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t), \sigma^2 \mathbf{I}) p_i(t)}{\sum_{j=1}^c p(x_k | w_j, \mu_j(t), \sigma^2 \mathbf{I}) p_j(t)}$$

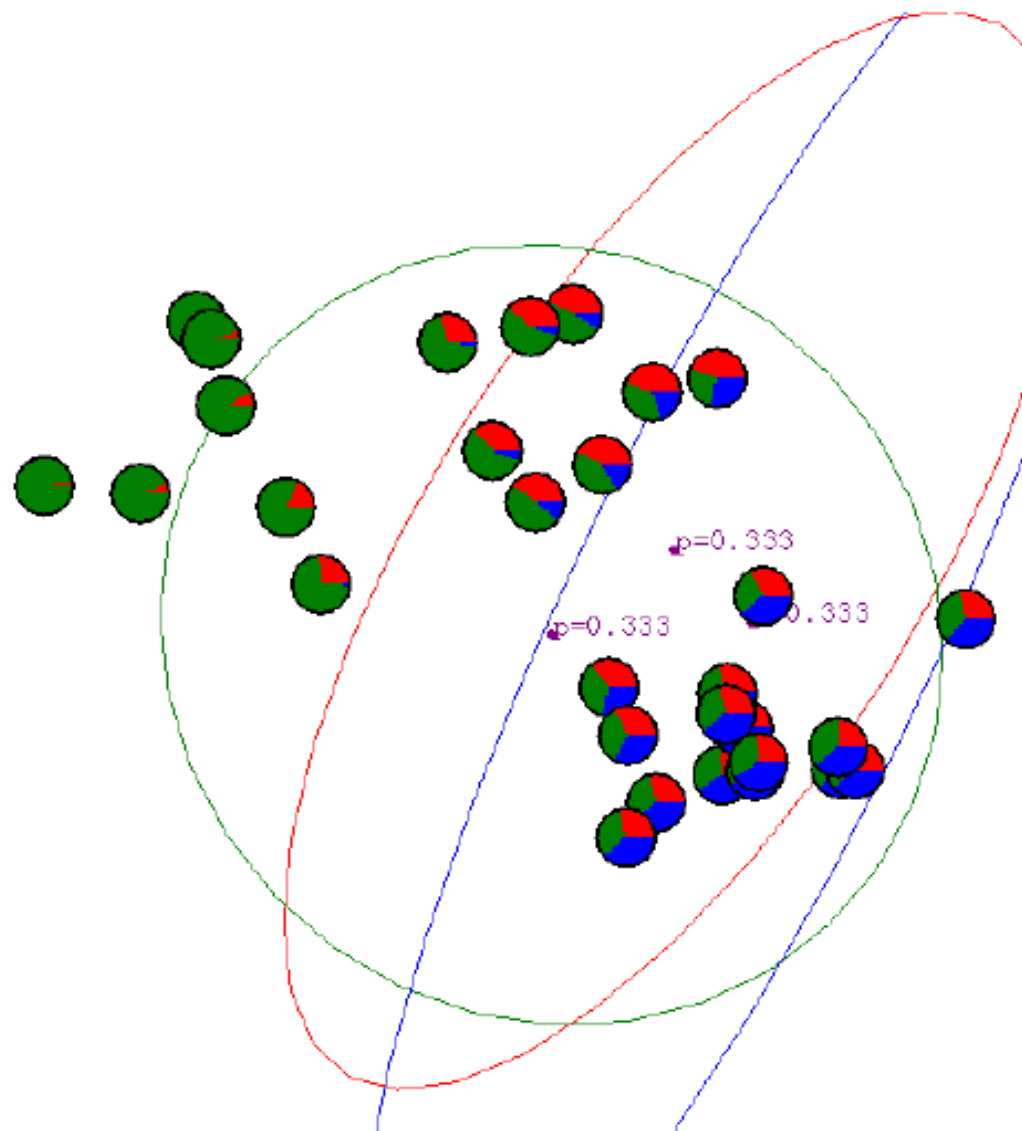
Just evaluate a Gaussian at  $x_k$

M-step.

Compute Max. like  $\mu$  given our data's class membership distributions

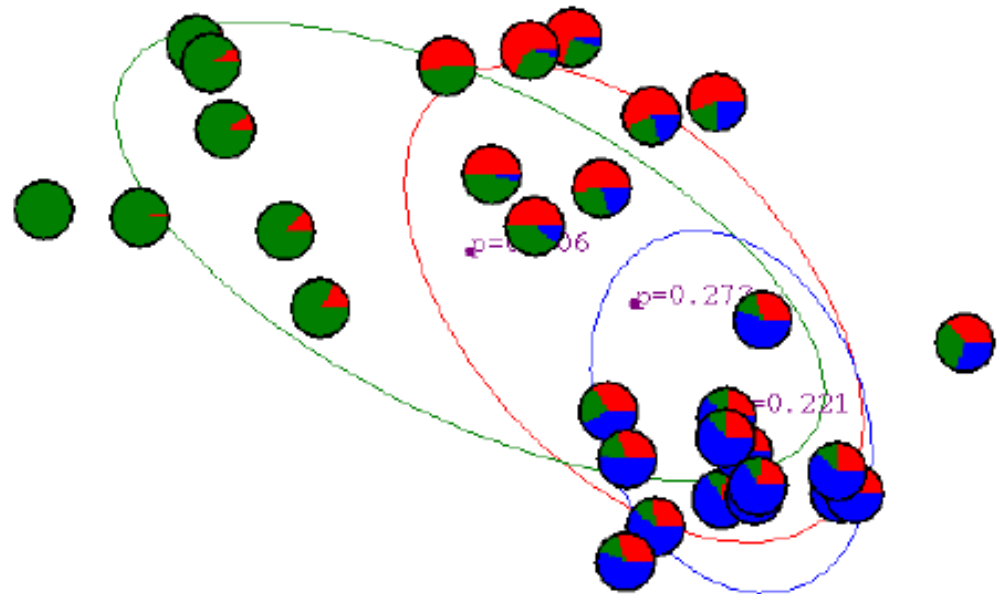
$$\mu_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) x_k}{\sum_k P(w_i | x_k, \lambda_t)}$$

# Gaussian Mixture Example: Start

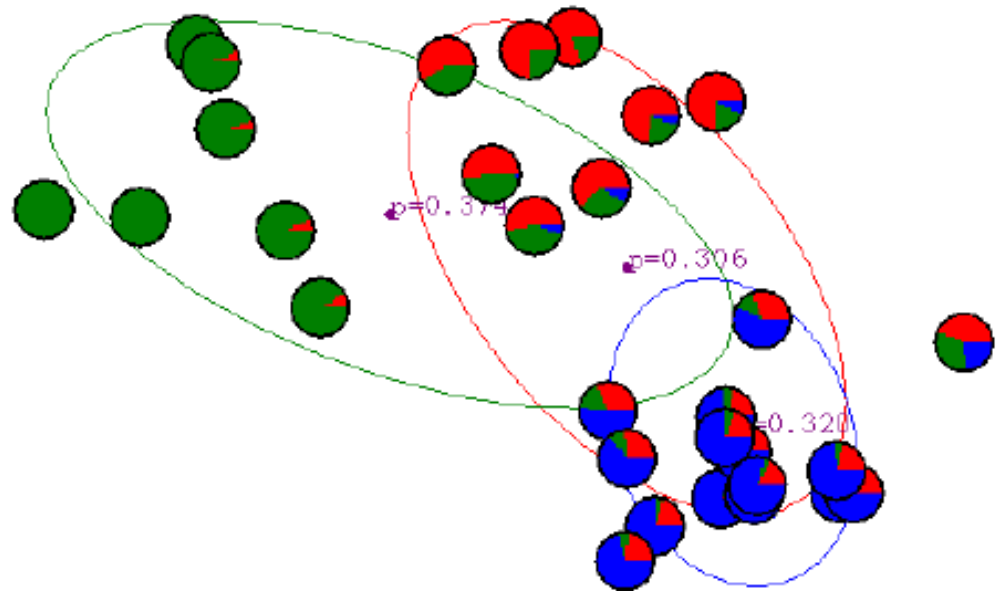


*Advance apologies: in Black and White this example will be incomprehensible*

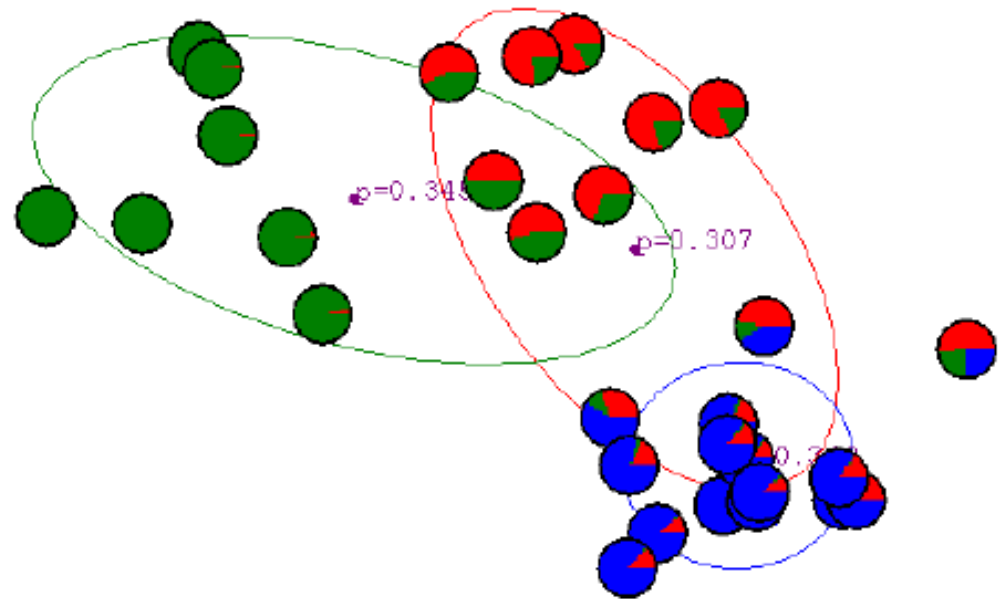
# After first iteration



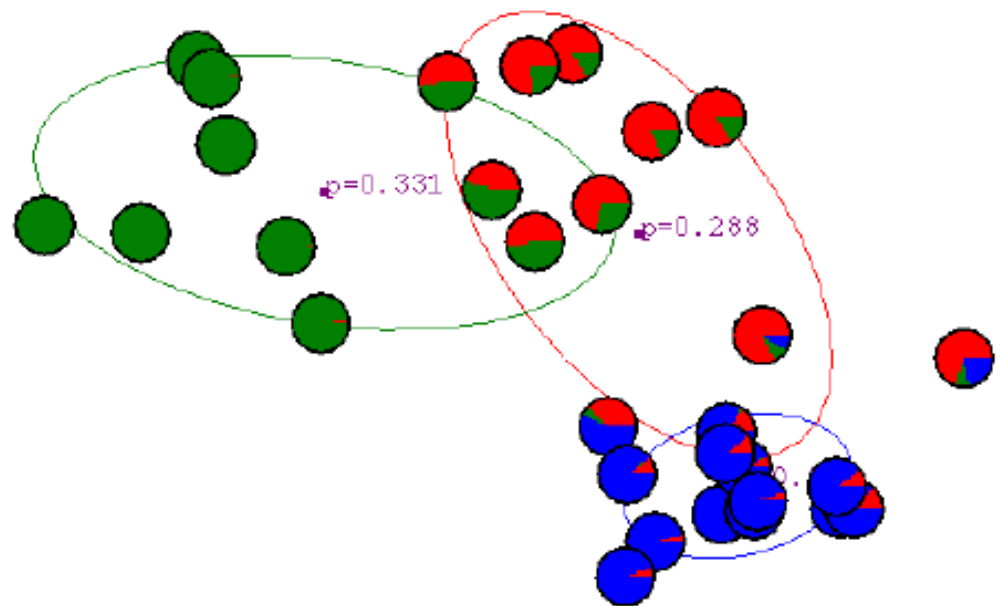
# After 2nd iteration



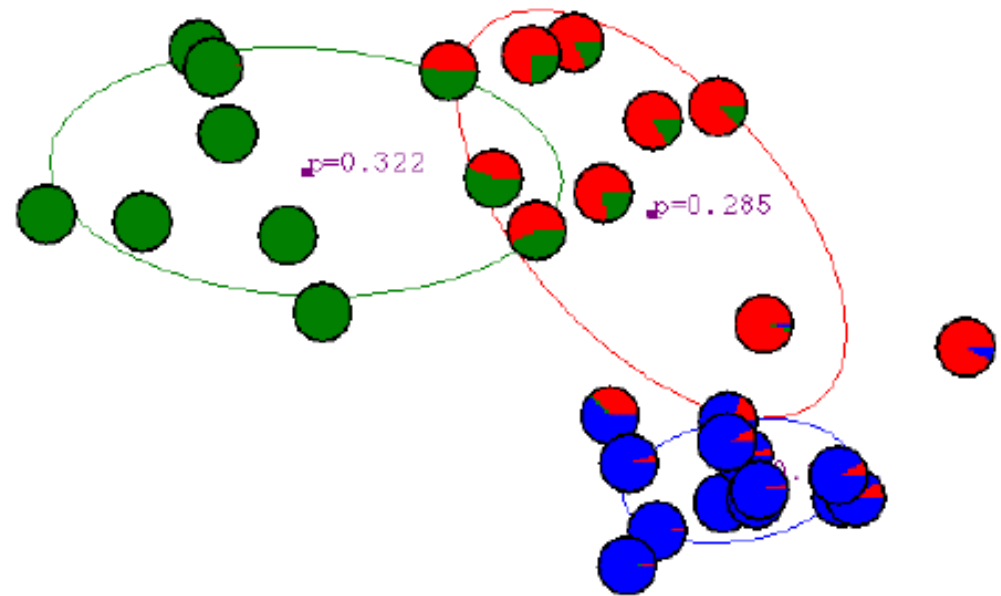
After 3rd  
iteration



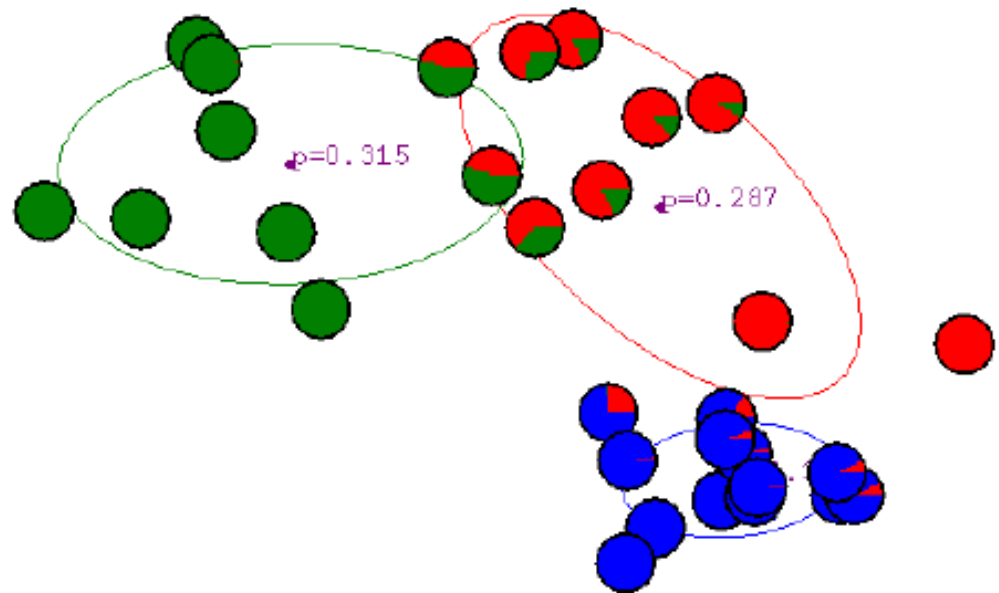
After 4th iteration



After 5th iteration

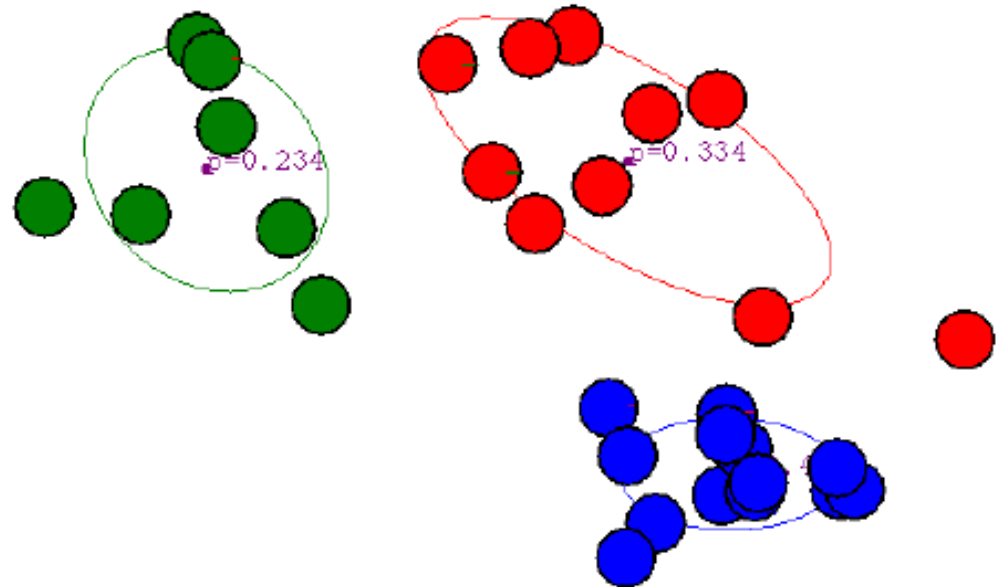


After 6th iteration





After 20th  
iteration



# Final comments

- Deal with missing data/latent variables
- Maximize expected log likelihood
- Local maxima

# Expectation-Maximization

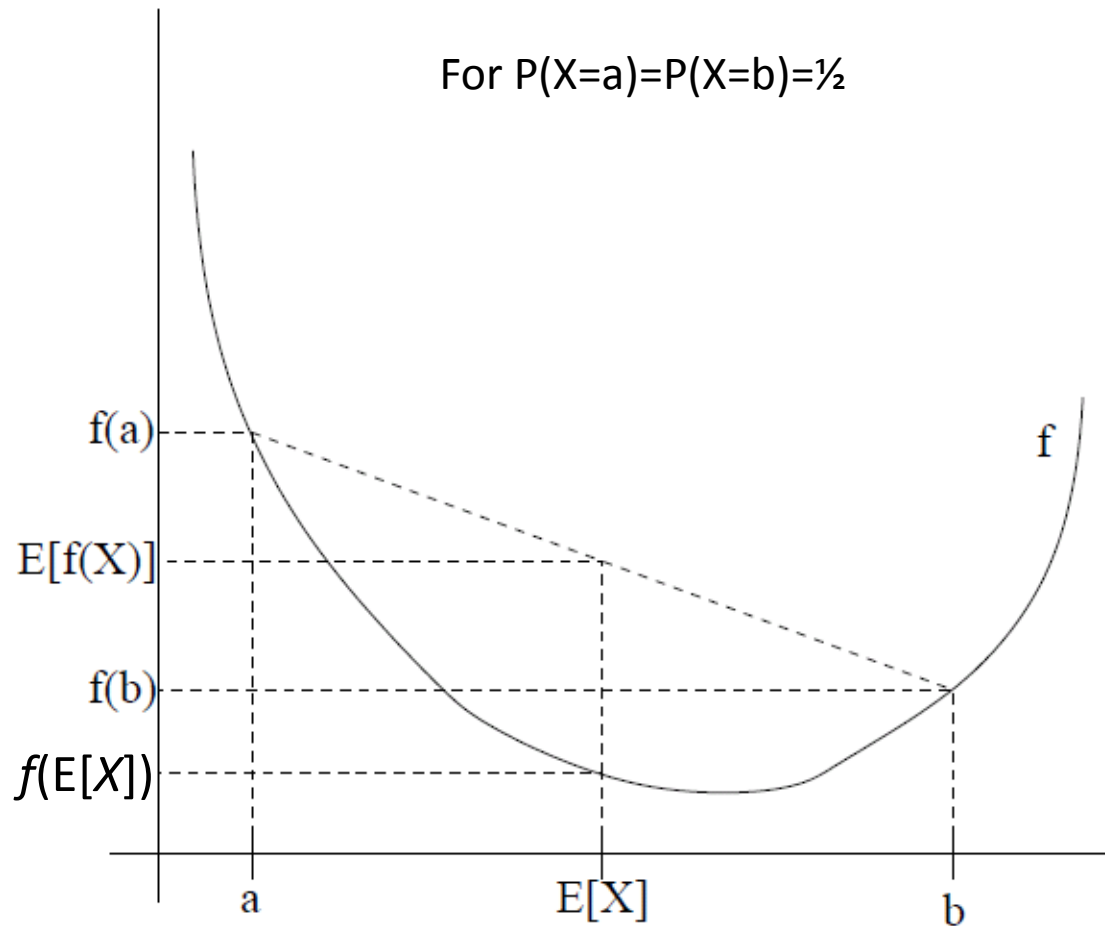
- Previously
  - Basics of EM
  - Learning a mixture of Gaussians (k-means)
- Next:
  - Short story justifying EM
    - Slides based on [lecture notes from Andrew Ng](#)

# 10,000 foot level EM

- Guess some parameters, then
  - Use your parameters to get a distribution over hidden variables
  - Re-estimate the parameters as if your distribution over hidden variables is correct
- Seems magical. When/why does this work?

# Jensen's Inequality

- For  $f$  convex,  $E[f(X)] \geq f(E[X])$



# Jensen's Inequality

- For  $f$  convex,  $E[f(X)] \geq f(E[X])$
- (on board)

# Maximizing likelihood

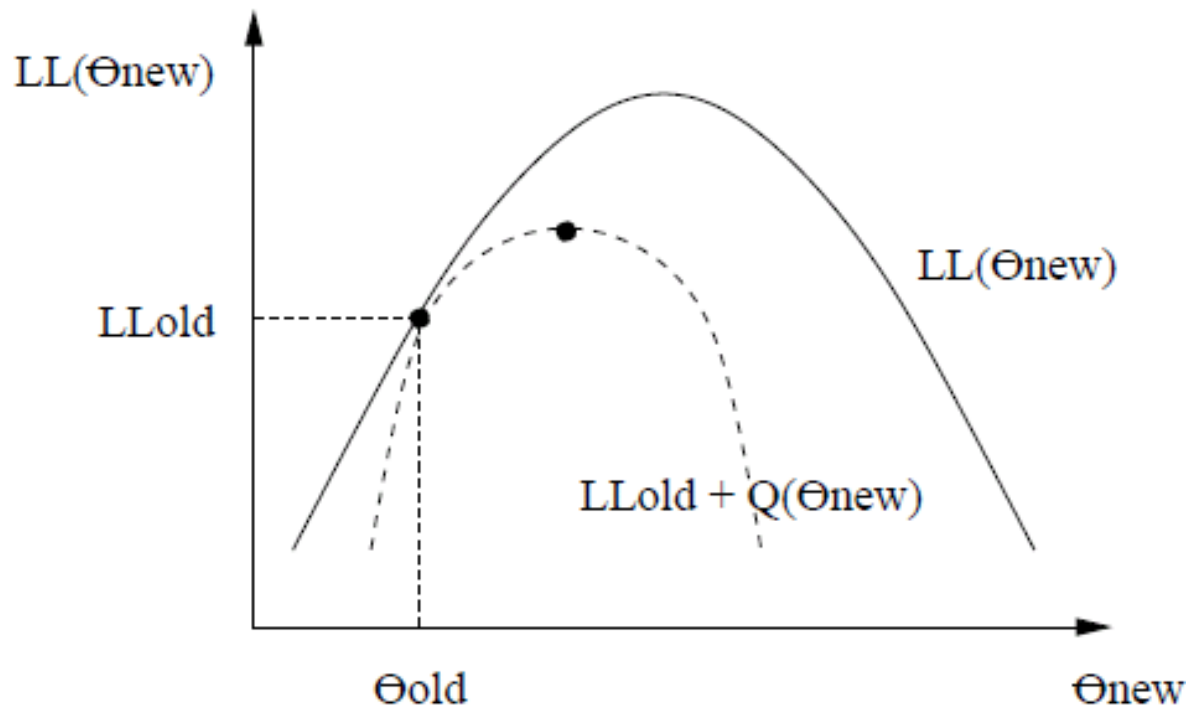
- $x^{(i)}$  = data,  $z^{(i)}$  = hidden vars,  $\theta$  = parameters

$$\begin{aligned}\sum_i \log p(x^{(i)}; \theta) &= \sum_i \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \\ &= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \\ &\geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}\end{aligned}$$

- This lower bound is easier to maximize, but
  - What is Q? What good is maximizing a lower bound?

# What do we use for $Q$ ?

- EM: Given a guess  $\theta_{\text{old}}$  for  $\theta$ , improve it
- Idea: choose  $Q$  such that our lower bound equals the true log likelihood at  $\theta_{\text{old}}$ :





Ensure the bound is tight at  $\theta_{\text{old}}$

- When does Jensen's inequality hold exactly?

# Ensure the bound is tight at $\theta_{\text{old}}$

- When does Jensen's inequality hold exactly?
- Sufficient that

$$\log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

be constant with respect to  $z^{(i)}$

- Thus, choose  $Q(z^{(i)}) = p(z^{(i)} \mid x^{(i)}; \theta_{\text{old}})$

# Putting it together

(E-step) For each  $i$ , set

$$Q_i(z^{(i)}) := p(z^{(i)} | x^{(i)}; \theta).$$

Old  $\theta$

(M-step) Set

$$\theta := \arg \max_{\theta} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

# For exponential family

- *E* step:
  - Use  $\theta_n$  to estimate **expected** sufficient statistics over **complete** data
- *M* step
  - Set  $\theta_{n+1}$  = ML parameters given sufficient statistics
    - (Or MAP parameters)

# EM in practice

- Local maxima
  - Random re-starts, simulated annealing...
- Variants
  - Hard EM: set  $Z$  to most likely value (e.g. k-means)
  - Generalized EM: increase (not nec. maximize) lower bound in each step
  - Approximate E-step (e.g. sampling)