

Tools & Models for Data Science

Optimization–Expectation Maximization

Chris Jermaine & Risa Myers

Rice University



- First, a few words...
 - EM is a very widely-used MLE algorithm for dealing with missing data
 - Perhaps the most intense thing we'll discuss this semester?
 - But definitely understandable to a Rice UG/MCS/PhD
 - So pay attention carefully!

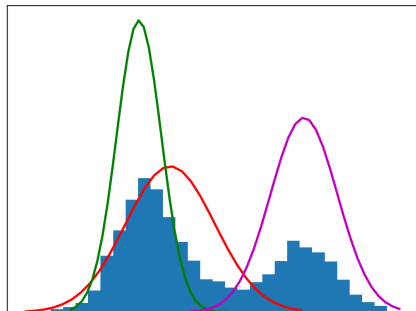
- Often, one has an optimization problem that would be easy...
 - Except that some of the data are missing
- Why might data be missing?
 - They were never recorded
 - Wrong values recorded
 - They are imaginary
 - ? When might data be imaginary?

Imaginary or Hidden data

- Complex models
- Imagine the process for generating the data
- Common models with hidden parameters
 - Gaussian Mixture Model
 - Hidden Markov Model
- intermediate steps & parameters are needed in the data generation process
- The values of these parameters are hidden from the observer
- For example
 - 1 Roll a die to choose a Gaussian
 - 2 Use that Gaussian to produce the data
- Common first steps are rolling a die or flipping a coin

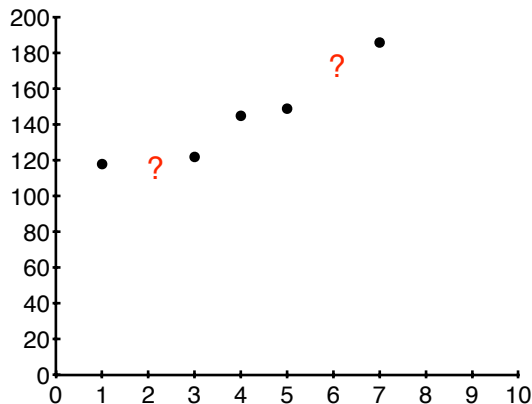
Gaussian Mixture Model

- Often used when we have a complex multi-variate distribution
- With weird shapes and many modes
- Hierarchical model
- K different Gaussians in our mixture
- Builds the distribution by mixing together K Gaussian distributions



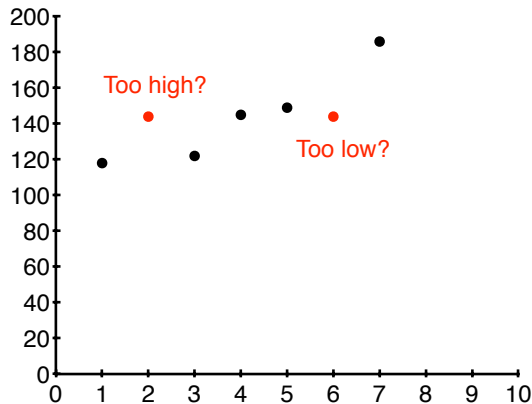
Why Can't We Do Something Simple?

- Like replace with the mean?
 - Back to the regression example
 - Want a line to fit points
 $\langle 118, ?, 122, 145, 149, ?, 186 \rangle$
 - Mean of observed data is 144



Why Can't We Do Something Simple?

- Like replace with the mean?
 - Back to the regression example
 - Want a line to fit points
 $\langle 118, ?, 122, 145, 149, ?, 186 \rangle$
 - Mean of observed data is 144
 - Does
 $\langle 118, 144, 122, 145, 149, 144, 186 \rangle$
make sense?
 - No: given our regression model,
we expect values in-keeping with
model



- EM lets us learn the model & integrate over all possible values in a fancy way

Can't We Just Drop the Data?

- In our example, why not just learn from $\langle 118, 122, 145, 149, 186 \rangle$?
 - Might make sense here...
 - But not in general
 - We can bias our data
 - Or discard a lot of useful information if just one bit is missing

- In data science, often impossible to drop missing data
- Because the data are missing (or hidden) by design
- Happens with “hierarchical models”

Hierarchical Model Example

- Have a bag with two coins
- First has probability p_1 of heads
- Second has probability p_2 of heads
- I repeatedly reach in, pull out a coin
- Identity is $z_i \in \{1, 2\}$
- Flip it 10 times and observe x_i heads
 - How do we compute $\Theta = \{p_1, p_2\}$?
 - Each z_i is missing: we don't know identity of coin
 - ? How could we just drop missing data in this case?
-
-
-
- This is a trial over a RV
- Represents coin selected at trial i
- Full dataset is $\{x_i, z_i\}$ pairs

- Formally: we want to compute an MLE for $L(\Theta|x_1, x_2, \dots, z_1, z_2, \dots)$
 - x_1, x_2, \dots are observed data
 - z_1, z_2, \dots are missing data
 - $\Theta = \{p_1, p_2\}$, the probability of heads for each coin
- Recall
 - This is just like computing the PDF
 - The Likelihood function flips the parameters
 - Measures how likely the parameters are given the data

- Recall that:

- $L(\Theta|x_1, x_2, \dots, z_1, z_2, \dots) = f(x_1, x_2, \dots, z_1, z_2, \dots|\Theta)$

- When the z 's are missing, choose Θ to max

$$\int_{\langle z_1, z_2, \dots \rangle} f(x_1, x_2, \dots, z_1, z_2, \dots|\Theta) d\langle z_1, z_2, \dots \rangle$$

“Integrating out” a Variable

$$\int_{\langle z_1, z_2, \dots \rangle} f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta) d\langle z_1, z_2, \dots \rangle$$

- Sum over all possible values of the variable you are integrating out
- Example: say we have (height, weight) pairs
- Probabilities are:

$\langle (\text{short}, \text{light}), 0.3 \rangle, \langle (\text{short}, \text{heavy}), 0.1 \rangle, \langle (\text{tall}, \text{light}), 0.2 \rangle, \langle (\text{tall}, \text{heavy}), 0.4 \rangle$

Probability they are “tall” is?

“Integrating out” a Variable

$$\int_{\langle z_1, z_2, \dots \rangle} f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta) d\langle z_1, z_2, \dots \rangle$$

- Sum over all possible values of the variable you are integrating out
- Example: say we have (height, weight) pairs
- Probabilities are:

$$\langle (\text{short}, \text{light}), 0.3 \rangle, \langle (\text{short}, \text{heavy}), 0.1 \rangle, \langle (\text{tall}, \text{light}), 0.2 \rangle, \langle (\text{tall}, \text{heavy}), 0.4 \rangle$$

- Probability they are “tall” is $0.6 = \sum_{\text{weight } w} Pr[(\text{tall}, w)]$
- Easy here, but difficult in the general case!

Expectation Maximization

- Is an iterative algorithm for difficult missing-data MLEs
- Basic idea...
 - Have an estimate Θ^{iter} for each iteration
 - Repeatedly update Θ^{iter} until convergence
 - Looks a lot like gradient descent, right?
- But EM is unique in how it deals with missing data points
 - The famous “ Q function”

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = E[\log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}}) | x_1, x_2, \dots, \Theta^{\text{iter}-1}]$$

- How to interpret the Q function?

- Q function is:

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = E \left[\log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}}) | x_1, x_2, \dots, \Theta^{\text{iter}-1} \right]$$

- Treat z_1, z_2, \dots as random variables
 - With distribution $f(z_1, z_2, \dots | x_1, x_2, \dots, \Theta^{\text{iter}-1})$
 - Kind of like Bayesian approach, with a prior and posterior distribution
 - We're going to get something that looks like a posterior distribution over the z s
 - The Q -function is the expected value of the LLH wrt this distribution

- What is expected value?

- Recall: expected value of $g(z)$ when z has distribution (PDF) $f(z)$ is $\sum_z f(z)g(z)$ or $\int_z f(z)g(z)dz$
- When z is discrete, $f(z)$ is a probability
- Example: If we sample (A,B) from

$$\langle(1,2),.3\rangle,\langle(3,5),0.1\rangle,\langle(2,6),0.2\rangle,\langle(-3,6),0.4\rangle$$

- $E[A+B] = 0.3 \times (1+2) + 0.1 \times (3+5) + 0.2 \times (2+6) + 0.4 \times (-3+6)$

- Continuous version is:

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = E [\log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}}) | x_1, x_2, \dots, \Theta^{\text{iter}-1}]$$

- If z s are discrete (usually are):

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = \sum_{\langle z_1, z_2, \dots \rangle} f(z_1, z_2, \dots | x_1, x_2, \dots, \Theta^{\text{iter}-1}) \log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}})$$

What are the Values of Z ?

- The z s are often useful
 - In a GMM, they tell you which cluster each data point belongs to
- Running EM doesn't actually tell you the values of the z s
- But, once you have Θ it's often easy to get the values
- E.g. Use Bayes rule to find the posterior probability for each cluster

Basic EM Algorithm

- It is an iterative algorithm
 - Start with a reasonable guess for Θ , call this Θ^0
- In each iteration:
 - Choose Θ^{iter} to maximize the expected value of the log-likelihood
- Stop when you can't improve any more

Back to the Example

- Best to return to our example...
 - Bag with two coins
 - $\Theta = \{p_1, p_2\}$: probability of each coin being heads
 - $z_i \in \{1, 2\}$: identity of coin flip the i th time I reach in the bag
 - $x_i \in \{0, \dots, 10\}$: number of heads for the i th trial
- So where do we start?

Back to the Example

- Best to return to our example...
 - Bag with two coins
 - $\Theta = \{p_1, p_2\}$: probability of each coin being heads
 - $z_i \in \{1, 2\}$: identity of coin flip the i th time I reach in the bag
 - $x_i \in \{0, \dots, 10\}$: number of heads for the i th trial
- So where do we start?
- With the likelihood function!

$$\begin{aligned} L(\Theta | x_1, x_2, \dots, z_1, z_2, \dots) &= f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta) \\ &= \prod_i f(x_i, z_i | \Theta) \\ &= \prod_i \frac{1}{2} \text{Binomial}(x_i | p_{z_i}, 10) \end{aligned}$$

- Assume each action is independent
- $f()$ is the density function given our parameter set
- x_i is the i th # of heads and z_i is the identity of the i th coin selected

Unpack the LLH Equation

$$= \prod_i \frac{1}{2} \text{Binomial}(x_i | p_{z_i}, 10)$$

- $\frac{1}{2}$ is the 50/50 chance of pulling out each coin
- Binomial because it models coin flips
- 10 coin flips
- p_{z_i} = Probability of Heads for each z_i (Recall: $z_i = \{1, 2\}$)
- We're using the identity of the coin to choose the right probability

What About the Posterior for the Missing Data

- Recall

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = \sum_{\langle z_1, z_2, \dots \rangle} f(z_1, z_2, \dots | x_1, x_2, \dots, \Theta^{\text{iter}-1}) \log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}})$$

- Note $f(z_1, z_2, \dots | x_1, x_2, \dots, \Theta^{\text{iter}-1}) = \prod_i f(z_i | x_i, \Theta^{\text{iter}-1})$

- Can take product because we assume independence

- Use Bayes' rule!

$$\begin{aligned} f(z_i | x_i, \Theta^{\text{iter}-1}) &= \frac{f(x_i, z_i | \Theta^{\text{iter}-1})}{f(x_i | \Theta^{\text{iter}-1})} \\ &= \frac{\frac{1}{2} \text{Binomial}(x_i | p_{z_i}^{\text{iter}-1}, 10)}{f(x_i | \Theta^{\text{iter}-1})} \end{aligned}$$

- Joint distribution of x_i, z_i given the parameters

- Divided by a normalizing constant

- What is $f(x_i | \Theta^{\text{iter}-1})$?

- What is $f(x_i|\Theta^{\text{iter}-1})$?

$$f(x_i|\Theta^{\text{iter}-1}) = \frac{1}{2}\text{Binomial}(x_i|p_1^{\text{iter}-1}, 10) + \frac{1}{2}\text{Binomial}(x_i|p_2^{\text{iter}-1}, 10)$$

- So, plug in these values to the equation on the previous slide to get:

$$f(z_i|x_i, \Theta^{\text{iter}-1}) = \frac{\frac{1}{2}\text{Binomial}(x_i|p_{z_i}^{\text{iter}-1}, 10)}{\frac{1}{2}\text{Binomial}(x_i|p_1^{\text{iter}-1}, 10) + \frac{1}{2}\text{Binomial}(x_i|p_2^{\text{iter}-1}, 10)}$$

- Cannot discard the denominator in this case, because we need the values to sum to 1
- Computing $f(z_i|x_i, \Theta^{\text{iter}-1})$ requires a pass over the data: called “E-Step”

OK, got the E-Step, Now What?

Now we need the M-Step, where we **maximize** Q wrt. Θ^{iter}

- Back to the Q function:

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = \sum_{\langle z_1, z_2, \dots \rangle} f(z_1, z_2, \dots | x_1, x_2, \dots, \Theta^{\text{iter}-1}) \log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}})$$

- Let $c_{i,j}$ denote $f(z_i = j | x_i, \Theta^{\text{iter}-1})$

$$Q(\Theta^{\text{iter}}, \Theta^{\text{iter}-1}) = \sum_{z_1} \sum_{z_2} \sum_{z_3} \dots \left(\prod_i c_{i,z_i} \right) \log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}})$$

OK, got the E-Step, Now What?

$$\sum_{z_1} \sum_{z_2} \sum_{z_3} \dots \left(\prod_i c_{i,z_i} \right) \log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}})$$

■ Where $\log f(x_1, x_2, \dots, z_1, z_2, \dots | \Theta^{\text{iter}})$ is:

$$\begin{aligned} \log \prod_i \text{Binomial}(x_i | p_{z_i}^{\text{iter}}) &= \sum_i \log \text{Binomial}(x_i | p_{z_i}^{\text{iter}}) \\ &\propto \sum_i \log \left((p_{z_i}^{\text{iter}})^{x_i} \times (1 - p_{z_i}^{\text{iter}})^{10-x_i} \right) \\ &= \sum_i x_i \log(p_{z_i}^{\text{iter}}) + (10 - x_i) \log(1 - p_{z_i}^{\text{iter}}) \end{aligned}$$

- log of products = sum of logs
- Binomial PMF $\binom{n}{k} p^k (1-p)^{n-k}$ has a combinatorial term, drop and switch to \propto
- Distribute the log function

OK, got the E-Step, Now What?

- Dropping the “iter” on $p_{z_i}^{\text{iter}}$ (since both terms have it), the Q function becomes:

$$\sum_{z_1} \sum_{z_2} \sum_{z_3} \dots \left(\prod_i c_{i,z_i} \right) \sum_i x_i \log(p_{z_i}) + (10 - x_i) \log(1 - p_{z_i})$$

- Note: We are summing over an exponential number of items

The M-Step

- Now, we need to maximize:

$$\sum_{z_1} \sum_{z_2} \sum_{z_3} \dots \left(\prod_i c_{i,z_i} \right) \sum_i x_i \log(p_{z_i}) + (10 - x_i) \log(1 - p_{z_i})$$

- Ugly! Or is it? Consider just one variable, z_1 . Write as:

$$\begin{aligned} & \sum_{z_1} \sum_{\langle z_2, z_3, \dots \rangle} c_{1,z_1} a(\langle z_2, z_3, \dots \rangle) \left(x_1 \log(p_{z_1}) + (10 - x_1) \log(1 - p_{z_1}) + \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \right) \\ &= \sum_{\langle z_2, z_3, \dots \rangle} c_{1,1} a(\langle z_2, z_3, \dots \rangle) \left(x_1 \log(p_1) + (10 - x_1) \log(1 - p_1) + \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \right) \\ &+ \sum_{\langle z_2, z_3, \dots \rangle} c_{1,2} a(\langle z_2, z_3, \dots \rangle) \left(x_1 \log(p_2) + (10 - x_1) \log(1 - p_2) + \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \right) \end{aligned}$$

- where a is the terms in $\prod_i c_{i,z_i}$ except for the current coin flip
- where b is the terms in the sum over the x_i except for the current coin flip

$$c_{1,1} \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \left(x_1 \log(p_1) + (10 - x_1) \log(1 - p_1) + \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \right) \\ + c_{1,2} \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \left(x_1 \log(p_2) + (10 - x_1) \log(1 - p_2) + \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \right)$$

- Continuing, split into the actual two terms, one for each coin, then distribute the c terms:

$$= c_{1,1} (x_1 \log(p_1) + (10 - x_1) \log(1 - p_1)) \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \\ + c_{1,1} \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \\ + c_{1,2} (x_1 \log(p_2) + (10 - x_1) \log(1 - p_2)) \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \\ + c_{1,2} \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle)$$

- Continuing, recombine in a simpler way & drop the $\sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle)$ from the first term:

$$\begin{aligned}
 & c_{1,1} (x_1 \log(p_1) + (10 - x_1) \log(1 - p_1)) \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \\
 & + c_{1,1} \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \\
 & + c_{1,2} (x_1 \log(p_2) + (10 - x_1) \log(1 - p_2)) \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \\
 & + c_{1,2} \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \\
 & = c_{1,1} (x_1 \log(p_1) + (10 - x_1) \log(1 - p_1)) + \\
 & + c_{1,2} (x_1 \log(p_2) + (10 - x_1) \log(1 - p_2)) + \\
 & + \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle)
 \end{aligned}$$

- Question: why can we drop $\sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle)$ from:

$$c_{1,1} (x_1 \log(p_1) + (10 - x_1) \log(1 - p_1)) \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle)$$

- Answer: $a(\langle z_2, z_3, \dots \rangle)$ is the posterior probability of flip sequence 2 coming from coin z_2 and flip sequence 3 coming from coin z_3 and flip sequence 4 coming from coin z_4 and so on.
- Since we sum over all possible $\langle z_2, z_3, \dots \rangle$, we are summing the probability all possible identities for coins 2, 3, 4...
- This has to equal 1!
- Why? By definition, when you sum the probability of all possibilities, you get 1

- Note the last term in

$$\begin{aligned} & c_{1,1} (x_1 \log(p_1) + (10 - x_1) \log(1 - p_1)) + \\ & + c_{1,2} (x_1 \log(p_2) + (10 - x_1) \log(1 - p_2)) + \\ & + \sum_{\langle z_2, z_3, \dots \rangle} a(\langle z_2, z_3, \dots \rangle) \sum_{i=2}^n b(\langle z_2, z_3, \dots \rangle) \end{aligned}$$

- This is like a “littler” Q -function, or what we get after removing the first coin flip (z_1). So we have shown the Q -function is just:

$$\begin{aligned} & c_{1,1} (x_1 \log(p_1) + (10 - x_1) \log(1 - p_1)) + \\ & c_{1,2} (x_1 \log(p_2) + (10 - x_1) \log(1 - p_2)) + \\ & \sum_{z_2} \sum_{z_3} \dots \left(\prod_{i \geq 2} c_{i, z_i} \right) \sum_{i \geq 2} x_i \log(p_{z_i}) + (10 - x_i) \log(1 - p_{z_i}) \end{aligned}$$

- So we've shown we can remove z_1 from the nasty summation $\sum_{z_1} \sum_{z_2} \sum_{z_3} \dots$
- We can use the same algebraic manipulations to get rid of z_2 , then z_3 , etc., giving us:

$$\sum_i c_{i,1} (x_i \log(p_1) + (10 - x_i) \log(1 - p_1)) + c_{i,2} (x_i \log(p_2) + (10 - x_i) \log(1 - p_2))$$

The M-Step

- Now we maximize wrt p_1, p_2
- Partial derivative wrt p_1 :

$$\sum_i c_{i,1} x_i \frac{1}{p_1} - \sum_i c_{i,1} (10 - x_i) \frac{1}{1 - p_1}$$

- Set to zero:

$$\begin{aligned}\sum_i c_{i,1} x_i \frac{1}{p_1} - \sum_i c_{i,1} (10 - x_i) \frac{1}{1 - p_1} &= 0 \\ \frac{1}{p_1} \sum_i c_{i,1} x_i - \frac{1}{1 - p_1} \sum_i 10 c_{i,1} + \frac{1}{1 - p_1} \sum_i c_{i,1} x_i &= 0 \\ (1 - p_1) \sum_i c_{i,1} x_i - p_1 \sum_i 10 c_{i,1} + p_1 \sum_i c_{i,1} x_i &= 0 \\ \sum_i c_{i,1} x_i - p_1 \sum_i 10 c_{i,1} &= 0 \\ p_1 &= \frac{\sum_i c_{i,1} x_i}{\sum_i 10 c_{i,1}}\end{aligned}$$

- Recall: $c_{i,j}$ denotes $f(z_i = j | x_i, \Theta^{\text{iter} - 1})$
- Let's consider

$$p_1 = \frac{\sum_i c_{i,1} x_i}{\sum_i 10 c_{i,1}}$$

- We are taking a weighted sum that looks at how many times, out of 10, we got heads
- Taking into account a weighting factor that each data point was effected by c_1
- We can repeat for c_2

The M-Step

- So, $p_2 = \frac{\sum_i c_{i,2} x_i}{\sum_i 10 c_{i,2}}$
- Very simple!!

```
set  $p_1 = 0.8, p_2 = 0.2$   
while ( $p_1, p_2$  still change) do  
  compute  $c_{i,1}, c_{i,2}$  for each  $i$   
  set  $p_1 = \frac{\sum_i c_{i,1} x_i}{\sum_i 10 c_{i,1}}$   
  set  $p_2 = \frac{\sum_i c_{i,2} x_i}{\sum_i 10 c_{i,2}}$   
end while
```

- But a long way to get there

A Quick Review

- Goal: Find the model parameters, p_1 and p_2
- Given: Observations of the number of heads in 10 coin tosses, over a number of trials
- It's (relatively) easy if we know which coin was selected during each trial

$$p_1 = \frac{\text{\# of heads using coin 1}}{\text{total \# of flips using coin 1}}$$

$$p_2 = \frac{\text{\# of heads using coin 2}}{\text{total \# of flips using coin 2}}$$

A Quick Review

- If we don't know which coin was selected for each trial, we need to estimate it
- We could do this by
 - 1 Computing the most likely coin selected for each observation, given an estimate of p_1 and p_2
 - 2 Then use these assignments to compute a revised MLE estimate of the parameters
 - 3 Repeat until p_1 and p_2 converge

- Alternatively, we could
 - Compute the probability for each possible combination of the selected coins
 - Use these probabilities to build a weighted function of the data to determine the probabilities
 - And re-estimate the parameters, using the weighted functions in an MLE
- Alternate between guessing the distribution of the missing data (the E-step)
- And re-estimating the parameter values (the M-step)

Thoughts about EM

- EM requires a lot of thinking and a lot of math
- It's very efficient (great for big data)
- However, if you have a lot of missing data
 - Use Markov Chain Monte Carlo (MCMC) / Bayesian methods
 - Easier than EM
 - Not as efficient

Questions?

- What do we know now that we didn't know before?
- How can we use what we learned today?