Learning with missing labels

Machine Learning

THE UNIVERSITY OF UTAH

So far in the class

We have focused *supervised learning*

Every example in the training set is *labeled* by an oracle, perhaps a noisy one

Training data: S = $\{(\mathbf{x}_i, y_i)\}$

We have seen various learning algorithms

And different ways to analyze learning

What if: The labels are missing

$\{\mathbf{x}_i\}$

Training data: S = ~~$\{(\mathbf{x}_i, y_i)\}$~~

Or alternatively:
We have a very small number of labeled examples.
And a large number of unlabeled examples

Semi-supervised learning: Few labeled examples, many unlabeled examples

Unsupervised learning: No labeled examples at all

This lecture

- Semi-supervised/Unsupervised learning
- Expectation-Maximization
- Variants of EM
    - K-Means

This lecture

- Semi-supervised/Unsupervised learning
- Expectation-Maximization
- Variants of EM
  - K-Means

Labeled data is a scarce resource

Expensive and time consuming to obtain

- Sometimes requires specialized expertise

Some of you are already facing this in your projects!

Some examples:

- Biology: If you want labeled genome data, you might not be able to get it without expensive lab work
- Language: Annotating semantics requires many linguists many days/years
- Computer vision: Annotating videos/images is time-consuming and expensive

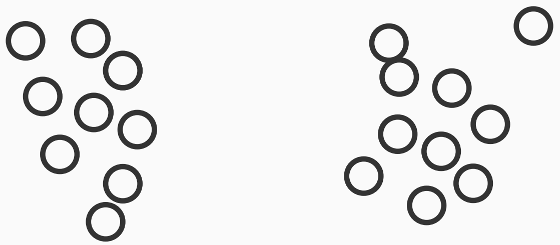
Unlabeled data is everywhere (*almost*)

Unsupervised learning

Can we learn without any labeled data?

Unsupervised learning

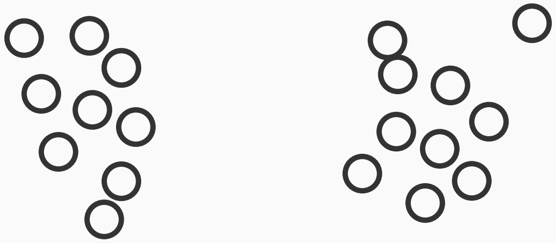
Can we learn without any labeled data?



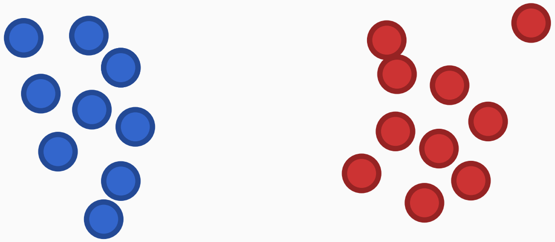
How would you label these points?

Unsupervised learning

Can we learn without any labeled data?

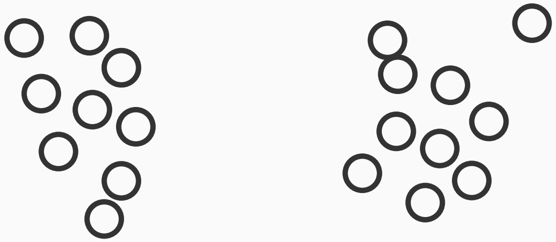


Perhaps this is a good labeling

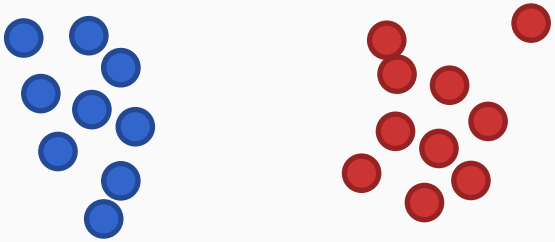


Unsupervised learning

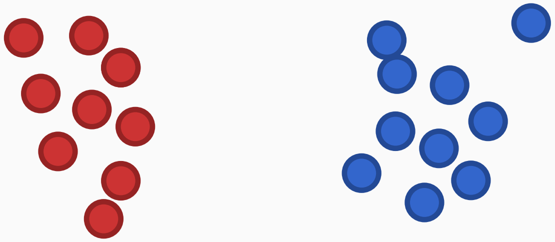
Can we learn without any labeled data?



Perhaps this is a good labeling

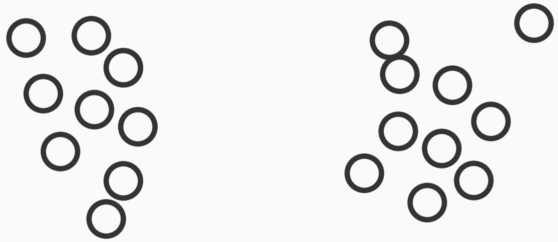


Or maybe this one



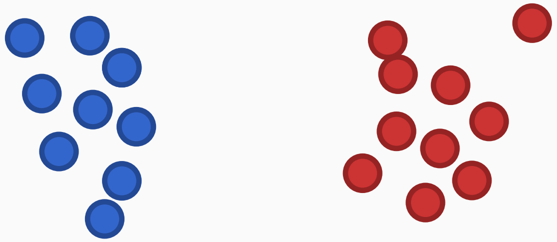
Unsupervised learning

Can we learn without any labeled data?

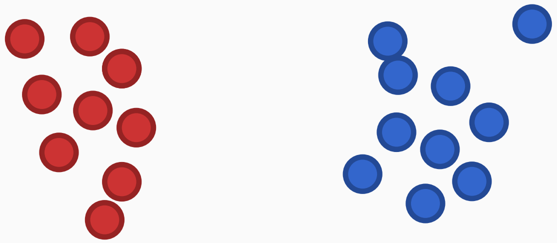


Without *any* labeled data, we might get parameters only up to symmetry

Perhaps this is a good labeling

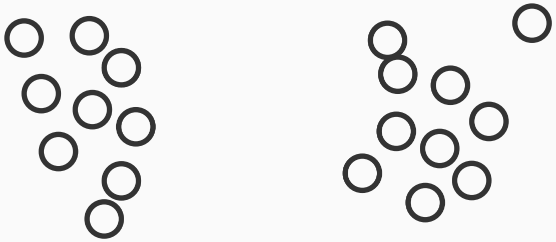


Or maybe this one

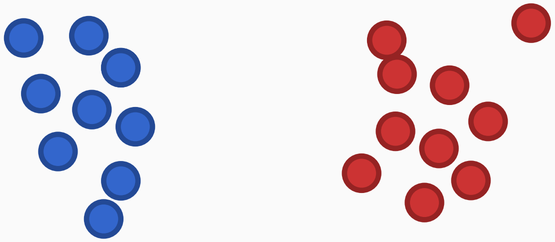


Unsupervised learning

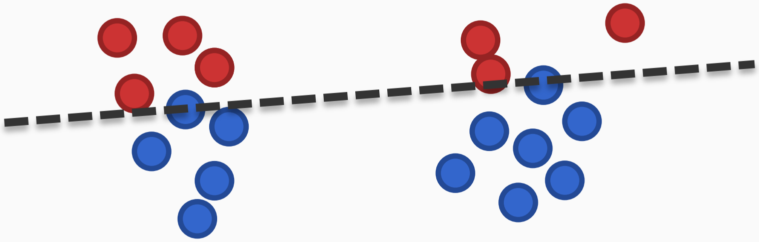
Can we learn without any labeled data?



Perhaps this is a good labeling

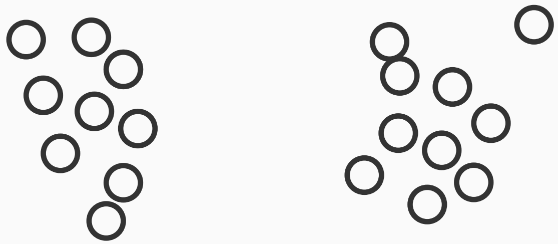


Why not this one?

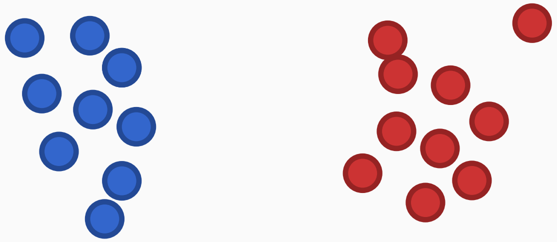


Unsupervised learning

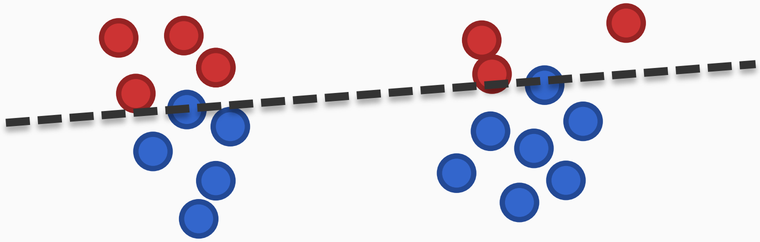
Can we learn without any labeled data?



Perhaps this is a good labeling



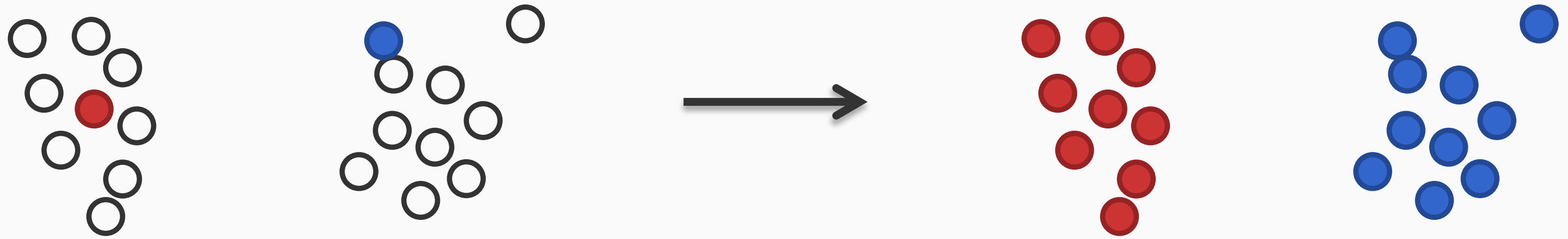
Why not this one?



Without *any* labeled data, we might have to make assumptions about regularities in the instance space

Semi-Supervised learning

Having a few labeled examples can help break symmetries



Example: Naïve Bayes

Suppose we are using a naïve Bayes classifier

- Features: x_1, x_2, x_3, x_4
- Label: y

If we had training data, we know how to estimate parameters of the model

$$p = P(y = 1) \qquad a_j = P(x_j = 1 | y = 1) \quad b_j = P(x_j = 1 | y = 0)$$

With the parameters, we can predict y for new examples

$$P(y | x_1, x_2, x_3, x_4) \propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)P(x_4|y)$$

Learning the naïve Bayes Classifier

If we had data, maximum likelihood estimation is easy

$$p = \frac{\text{Count}\,(y_i = 1)}{\text{Count}\,(y_i = 1) + \text{Count}\,(y_i = 0)}$$

← $P(y = 1) = p$

$$a_j = \frac{\text{Count}\,(y_i = 1, x_{ij} = 1)}{\text{Count}\,(y_i = 1)}$$

← $P(x_j = 1 \mid y = 1) = a_j$

$$b_j = \frac{\text{Count}\,(y_i = 0, x_{ij} = 1)}{\text{Count}\,(y_i = 0)}$$

← $P(x_j = 1 \mid y = 0) = b_j$

Using unlabeled examples

Say we use *ten* labeled examples to get the following probabilities

$p = P(y = 1) = 1/2$
$1 - p = P(y = 0) = 1/2$

j	$a_j = P(x_j = 1 \mid y_j = 1)$	$b_j = P(x_j = 1 \mid y_j = 0)$
1	3/4	1/4
2	1/2	1/4
3	1/2	3/4
4	1/2	1/2

Now, for a new example (1,0,0,0):

$$P(y|x_1, x_2, x_3, x_4) \propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)P(x_4|y)$$

Using unlabeled examples

Say we use *ten* labeled examples to get the following probabilities

p = P(y = 1) = 1/2
1- p = P(y = 0) = 1/2

j	$a_j = P(x_j = 1 \mid y_j = 1)$	$b_j = P(x_j = 1 \mid y_j = 0)$
1	3/4	1/4
2	1/2	1/4
3	1/2	3/4
4	1/2	1/2

Now, for a new example (1,0,0,0):

$$P(y = 1|1,0,0,0) \propto \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{64} = \frac{12}{256}$$

$$P(y = 0|1,0,0,0) \propto \frac{1}{2} \cdot \frac{1}{4} \cdot \frac{3}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{3}{256}$$

$$P(y = 1|\mathbf{x}) = \frac{12}{15} \qquad P(y = 0|\mathbf{x}) = \frac{3}{15}$$

Using unlabeled examples

Say we use *ten* labeled examples to get the following probabilities

p = P(y = 1) = 1/2
1- p = P(y = 0) = 1/2

j	a_j = P(x_j = 1 | y_j= 1)	b_j = P(x_j = 1 | y_j = 0)
1	3/4	1/4
2	1/2	1/4
3	1/2	3/4
4	1/2	1/2

Now, for a new example (1,0,0,0):

$$P(y=1|\mathbf{x}) = \frac{12}{15} \qquad P(y=0|\mathbf{x}) = \frac{3}{15}$$

What could we do with this information to improve our probability estimates?

Using unlabeled examples

For an unlabeled data point (1, 0, 0, 0), our model estimates that

$$P(y = 1|\mathbf{x}) = \frac{12}{15} \qquad P(y = 0|\mathbf{x}) = \frac{3}{15}$$

Some options:

1. The model predicts a label. Use it as a labeled example
   - In this case y = 1
   - Or perhaps, we could only do this when our classifier is confident enough
2. The model does not predict a label. It predicts a fractional label!
   - Recall: learning only needed counts. Counts do not need to be integers
   - This example is a 12/15 positive example and a 3/15 negative example

Broad strategies for using unlabeled data

1. Use a confidence threshold: When the label for an example is predicted with high enough confidence by the current model,
    1. Treat it as a labeled example [1 *or* 0]
    2. Retrain the model

2. Use fractional examples:
    1. Label examples as [P(y=1 |x) of 1 ***and*** P(y=0 | x) of 0]
    2. Retrain the model

Both approaches can be used iteratively

Unsupervised learning

Previous discussion: What if we had *ten* labeled examples and many unlabeled examples

What if: We have ***zero*** labeled examples and many unlabeled examples

We could still do the same

- Start with a guess for the probabilities
- Continue as above

This is a version of *Expectation Maximization*

This lecture

- Semi-supervised/Unsupervised learning
- Expectation-Maximization
- Variants of EM
  - K-Means

Expectation Maximization

- A meta-algorithm to estimate a probability distribution in when attributes are missing

- Needs assumptions about the underlying probability distribution
  - Suited to generative models
  - Performance sensitive to the validity of the assumption (and also the initial guess of the parameters)

- Converges to a local maximum of the likelihood function

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

The three coin example

We have three coins

Coin 0: P(Heads) = α
Coin 1: P(Heads) = p
Coin 2: P(Heads) = q

Scenario 1: Someone picks either coin 1 or coin 2 and tosses it

Observation: H H T H

Which coin is more likely to have been tossed?

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 1: Someone picks either coin 1 or coin 2 and tosses it

Observation: H H T H

Which coin is more likely to have been tossed?

$$P(\text{Coin 1} \mid \text{H H T H}) \propto P(\text{H H T H} \mid \text{Coin 1}) = p^3(1-p)$$

$$P(\text{Coin 2} \mid \text{H H T H}) \propto P(\text{H H T H} \mid \text{Coin 2}) = q^3(1-q)$$

If we know p and q, we could compute these values and decide which is higher

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

If we know what the probabilities are, we can compute the probability that an observation came from a particular coin

Scenario 1: Someone picks either coin 1 or coin 2 and tosses it

Observation: H H T H

Which coin is more likely to have been tossed?

$$P(\text{Coin } 1 \mid H\ H\ T\ H) \propto P(H\ H\ T\ H \mid \text{Coin } 1) = p^3(1-p)$$

$$P(\text{Coin } 2 \mid H\ H\ T\ H) \propto P(H\ H\ T\ H \mid \text{Coin } 2) = q^3(1-q)$$

If we know p and q, we could compute these values and decide which is higher

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

The three coin example

We have three coins

Coin 0: P(Heads) = α
Coin 1: P(Heads) = p
Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

Observations: ***H*** HHHT, ***T*** HTHT, ***H*** HHHT, ***H*** HTTH

From these observations, estimate the values of p, q and α?

The three coin example

We have three coins

Coin 0: P(Heads) = α

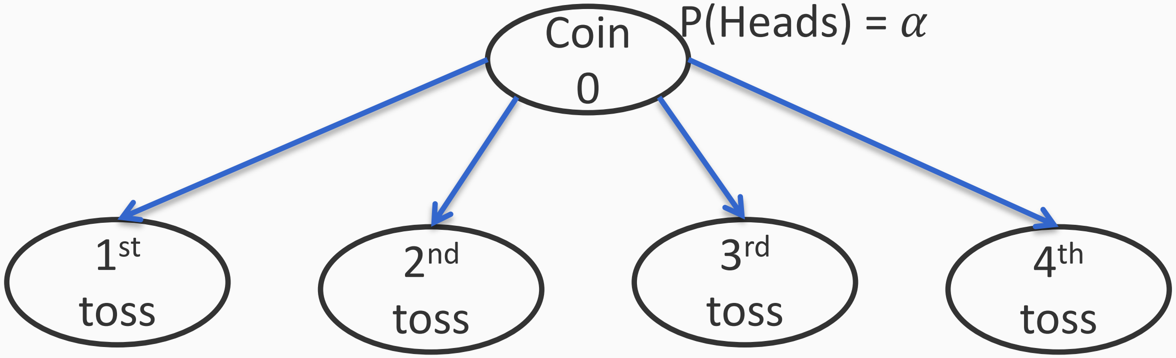
Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

Observations: ***H*** HHHT, ***T*** HTHT, ***H*** HHHT, ***H*** HTTH

From these observations, estimate the values of p, q and α?



The three coin example

We have three coins

Coin 0: P(Heads) = α
Coin 1: P(Heads) = p
Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

Observations: ***H***HHHT, ***T***HTHT, ***H***HHHT, ***H***HTTH

From these observations, estimate the values of p, q and α?

α = 3/4

The three coin example

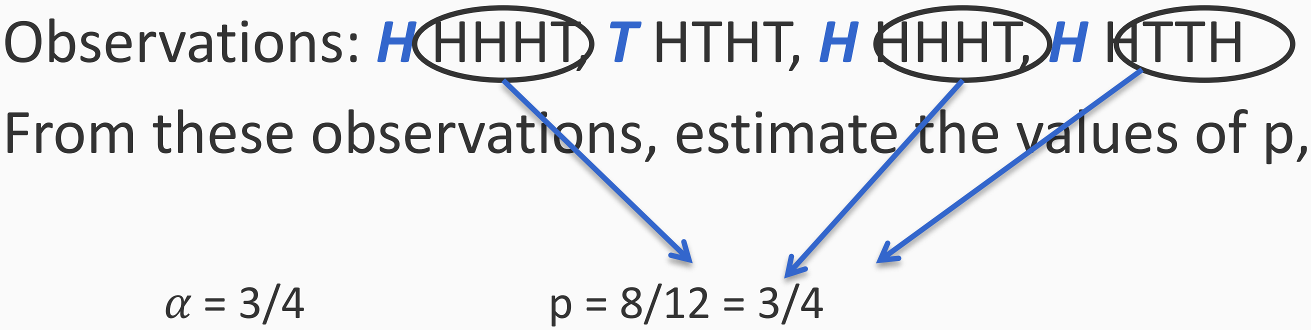
We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times



Observations: ***H*** HHHT, ***T*** HTHT, ***H*** HHHT, ***H*** HTTH

From these observations, estimate the values of p, q and α?

α = 3/4

p = 8/12 = 3/4

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

Observations: ***H*** HHHT, ***T*** HTHT, ***H*** HHHT, ***H*** HTTH

From these observations, estimate the values of p, q and α?

$\alpha = 3/4$ p = 8/12 = 3/4 q = 2/4= 1/2

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 2: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

Observations: ***H*** HHHT, ***T*** HTHT, ***H*** HHHT, ***H*** HTTH

From these observations, estimate the values of p, q and α?

α = 3/4 p = 8/12 = 3/4 q = 2/4= 1/2

If we knew which of the data points came from Coin1 and which from Coin2, there is no problem

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 3: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

But we observe only the tosses produced by coins 1 and 2

Observations: HHHT, HTHT, HHHT, HTTH

From these observations, estimate the values of p, q and α?

The three coin example

We have three coins

Coin 0: P(Heads) = α
Coin 1: P(Heads) = p
Coin 2: P(Heads) = q

Scenario 3: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

But we observe only the tosses produced by coins 1 and 2

Observations: HHHT, HTHT, HHHT, HTTH

From these observations, estimate the values of p, q and α?

There is no known analytical solution to this problem (in the general setting).

That is, it is not known how to compute the values of the parameters so as to maximize the likelihood of the data

What we know

1. Scenario 1: If we know what the coin biases are, we can compute the probability that an observation came from a particular coin

   P(missing variable | observation, coin biases)

2. Scenario 2: If we knew which of the data points came from Coin1 and which from Coin2, we can compute the P(heads) for all the coins

One approach

1. Guess the probability that an observation (e.g: HHHT) comes from coin 1 or coin 2

One approach

1. Guess the probability that an observation (e.g: HHHT) comes from coin 1 or coin 2
2. Loop:

One approach

1. Guess the probability that an observation (e.g: HHHT) comes from coin 1 or coin 2
2. Loop:
    1. Use this probability to get label (Coin 0’s value for each observation), possibly fractional

One approach

1. Guess the probability that an observation (e.g: HHHT) comes from coin 1 or coin 2
2. Loop:
    1. Use this probability to get label (Coin 0’s value for each observation), possibly fractional
    2. Now we have fully labeled data, estimate the maximum likelihood estimates of the coin biases

One approach

1. Guess the probability that an observation (e.g: HHHT) comes from coin 1 or coin 2
2. Loop:
    1. Use this probability to get label (Coin 0’s value for each observation), possibly fractional
    2. Now we have fully labeled data, estimate the maximum likelihood estimates of the coin biases
    3. Now we know the coin biases, re-estimate the probability that an observation comes from coin 1 or 2

One approach

1. Guess the probability that an observation (e.g: HHHT) comes from coin 1 or coin 2
2. Loop:
    1. Use this probability to get label (Coin 0's value for each observation), possibly fractional
    2. Now we have fully labeled data, estimate the maximum likelihood estimates of the coin biases
    3. Now we know the coin biases, re-estimate the probability that an observation comes from coin 1 or 2

This will converge to a local maximum of the overall likelihood function

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

In scenario 3:

- Parameters are α, p, q
- Each example $\mathbf{x}_i$ is ith sequence of coin tosses of coin 1 or 2 at that round
- Let us refer to the value of coin 0 for each $\mathbf{x}_i$ as y_i

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

In scenario 3:

$$LL(\text{data}|p, q, \alpha) = \sum_i \log P(\text{example}_i|p, q, \alpha)$$

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

In scenario 3:

$$LL(\text{data}|p, q, \alpha) = \sum_i \log P(\text{example}_i|p, q, \alpha)$$

The ***full*** example is $\mathbf{x}_i$ and y_i. And a part of it is hidden.

So how do we get P(example$_i$ | parameters)?

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

In scenario 3:

$$LL(\text{data}|p, q, \alpha) = \sum_i \log P(\text{example}_i|p, q, \alpha)$$

The ***full*** example is $\mathbf{x}_i$ and y_i. And a part of it is hidden.

So how do we get P(example$_i$ | parameters)?

Answer: Marginalize out the hidden variables

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

In scenario 3:

$$LL(\text{data}|p, q, \alpha) = \sum_i \log P(\text{example}_i|p, q, \alpha)$$

$$P(\mathbf{x}_i|p, q, \alpha) = \sum_{y_i} P(\mathbf{x}_i, y_i|p, q, \alpha)$$

Maximum likelihood estimation

MLE: Find parameters that maximize the likelihood (or equivalently log-likelihood) of the data

$$LL(\text{data}|\text{parameters}) = \sum_i \log P(\text{example}_i|\text{parameters})$$

In scenario 3:

$$LL(\text{data}|p, q, \alpha) = \sum_i \log \sum_{y_i} P(\mathbf{x}_i, y_i|p, q, \alpha)$$

This is the log likelihood we would like to maximize for MLE

This maximization is not easy. Sum inside log

Expectation Maximization

What we want (but can't have)

Log-likelihood of the observations

$$LL(\text{data}|p,q,\alpha) = \sum_i \log \sum_{y_i} P(\mathbf{x}_i, y_i|p,q,\alpha)$$

The strategy: Think of log probabilities as random variables

Learn by repeatedly maximizing a lower bound of LL

$$\mathcal{L}(\theta;Q) = \sum_i E_{y\sim Q_i}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y\sim Q_i}\left[\log Q_i(y)\right]$$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$LL(\text{data}|\theta) \quad = \quad \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

Why do we want to maximize this? Because this gives us the maximum likelihood estimate

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$
\begin{aligned}
LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\
&= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right)
\end{aligned}
$$

This is true for *any* probability distribution $Q_i(y)$

The summation over y is the definition of expectation with respect to $Q_i(y)$

$$E_{z\sim Q}\left[f(z)\right] = \sum_z Q(z) f(z)$$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$\begin{aligned} LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\ &= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \end{aligned}$$

$$E_{z \sim Q}\left[f(z)\right] = \sum_z Q(z) f(z)$$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$\begin{aligned} LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\ &= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \\ &= \sum_i \log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \end{aligned}$$

$$E_{z \sim Q}\left[f(z)\right] = \sum_z Q(z) f(z)$$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$\begin{aligned} LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\ &= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \\ &= \sum_i \log E_{y\sim Q_i}\left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \end{aligned}$$

Jensen’s inequality

If f is a convex function and X is a random variable, then

$$f(E[X]) \leq E[f(X)]$$

Or:

If f is a concave function and X is a random variable, then

$$f(E[X]) \geq E[f(X)]$$

Jensen’s inequality

If f is a concave function and X is a random variable, then

$$f(E[X]) \geq E[f(X)]$$

Let us apply this to the following function:

$$\log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right]$$

log is a concave function and *the function inside the expectation* is a random variable

$$\log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \geq E_{y \sim Q_i} \left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right]$$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$
\begin{aligned}
LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\
&= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \\
&= \sum_i \log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right]
\end{aligned}
$$

By Jensen's inequality $\log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \geq E_{y \sim Q_i} \left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right]$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$\begin{aligned} LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\ &= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \\ &= \sum_i \log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \\ &\geq \sum_i E_{y \sim Q_i} \left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \end{aligned}$$

By Jensen's inequality $\log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \geq E_{y \sim Q_i} \left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right]$

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$
\begin{aligned}
LL(\text{data}|\theta) &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\
&= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \\
&= \sum_i \log E_{y \sim Q_i} \left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \\
&\geq \sum_i E_{y \sim Q_i} \left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \\
&= \sum_i E_{y \sim Q_i} [\log P(\mathbf{x}_i, y|\theta)] - \sum_i E_{y \sim Q_i} [\log Q_i(y)]
\end{aligned}
$$

Rewrite log

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$\begin{aligned}
\boxed{LL(\text{data}|\theta)} &= \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta) \\
&= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right) \\
&= \sum_i \log E_{y\sim Q_i}\left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \\
&\geq \sum_i E_{y\sim Q_i}\left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right] \\
&= \boxed{\sum_i E_{y\sim Q_i}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y\sim Q_i}\left[\log Q_i(y)\right]}
\end{aligned}$$

Greater than

Let us build an approximation

What we want: Maximize LL(data|p, q, α). Denote (p, q, α) = θ

$$LL(\text{data}|\theta) = \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

$$= \sum_i \log \sum_y \left(Q_i(y) \cdot \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right)$$

Gre
tha

The strategy: Let us maximize this lower bound on the likelihood instead

$$\geq \sum_i E_{y\sim Q_i} \left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)} \right]$$

$$= \sum_i E_{y\sim Q_i} [\log P(\mathbf{x}_i, y|\theta)] - \sum_i E_{y\sim Q_i} [\log Q_i(y)]$$

Expectation Maximization

What we want (but can’t have)

Log-likelihood of the observations

$$LL(\text{data}|p, q, \alpha) = \sum_i \log \sum_{y_i} P(\mathbf{x}_i, y_i | p, q, \alpha)$$

Expectation Maximization

What we want (but can't have) Log-likelihood of the observations

$$LL(\text{data}|p, q, \alpha) = \sum_i \log \sum_{y_i} P(\mathbf{x}_i, y_i | p, q, \alpha)$$

The strategy: Think of log probabilities as random variables

Learn by repeatedly maximizing a lower bound of LL

$$\mathcal{L}(\theta; Q) = \sum_i E_{y \sim Q_i} [\log P(\mathbf{x}_i, y|\theta)] - \sum_i E_{y \sim Q_i} [\log Q_i(y)]$$

Expectation Maximization

Learning by maximizing expected log likelihood of the data

$$\mathcal{L}(\theta; Q) = \sum_i E_{y \sim Q_i}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y \sim Q_i}\left[\log Q_i(y)\right]$$

Expectation Maximization

Learning by maximizing expected log likelihood of the data

$$\mathcal{L}(\theta; Q) = \sum_i E_{y\sim Q_i}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y\sim Q_i}\left[\log Q_i(y)\right]$$

Still need to decide what is a good Q_i

What we would like is the one that makes this lower bound tight

(Jensen's inequality) $\log E_{y\sim Q_i}\left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)}\right] \geq E_{y\sim Q_i}\left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)}\right]$

Expectation Maximization

Learning by maximizing expected log likelihood of the data

$$\mathcal{L}(\theta; Q) = \sum_i E_{y\sim Q_i}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y\sim Q_i}\left[\log Q_i(y)\right]$$

Still need to decide what is a good Q_i

What we would like is the one that makes this lower bound tight

(Jensen's inequality) $\log E_{y\sim Q_i}\left[\frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)}\right] \geq E_{y\sim Q_i}\left[\log \frac{P(\mathbf{x}_i, y|\theta)}{Q_i(y)}\right]$

We can show that if we had an estimate of the θ, say θ^t, then a tight lower bound is given by setting

$$Q_i(y) = P(y|\mathbf{x}_i, \theta^t)$$

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\mathcal{L}(\theta; Q^t) = \sum_i E_{y\sim Q_i^t}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y\sim Q_i^t}\left[\log Q_i^t(y)\right]$$

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\mathcal{L}(\theta; Q^t) = \sum_i E_{y\sim Q_i^t}\left[\log P(\mathbf{x}_i, y|\theta)\right] - \sum_i E_{y\sim Q_i^t}\left[\log Q_i^t(y)\right]$$

Independent of θ

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

Intuitively: What is distribution over the hidden variables for this set of parameters

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

Intuitively: What is distribution over the hidden variables for this set of parameters

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

Intuitively: Using the current estimate for the hidden variables, what is the best set of parameters for the entire data

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

Given the parameters, we can compute this function. Why?

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

- Return final θ

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

Given the parameters, we can compute this function. Why?

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} [\log P(\mathbf{x}_i, y|\theta)]$$

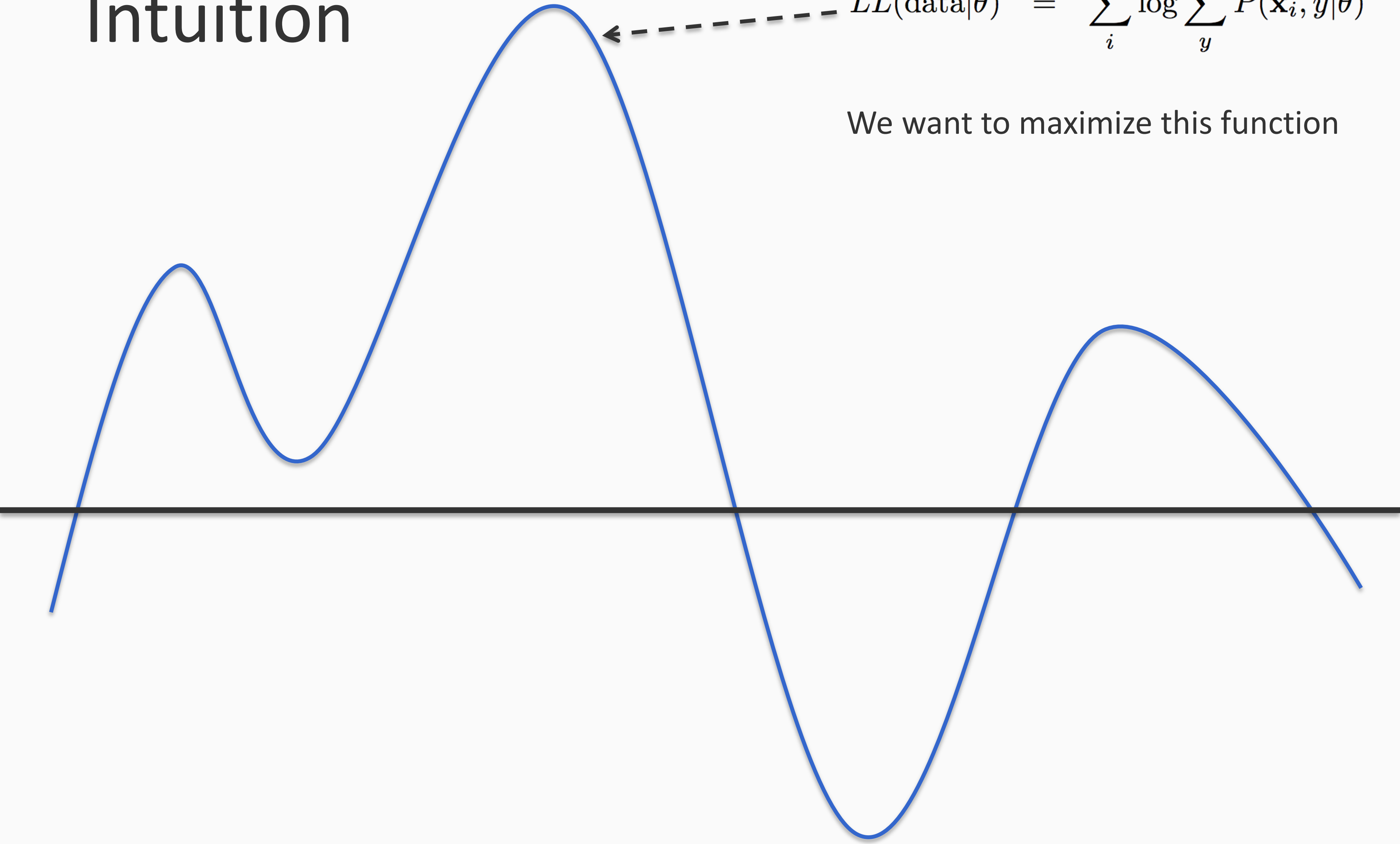
This step needs can be solved either analytically or algorithmically.

- Return final θ

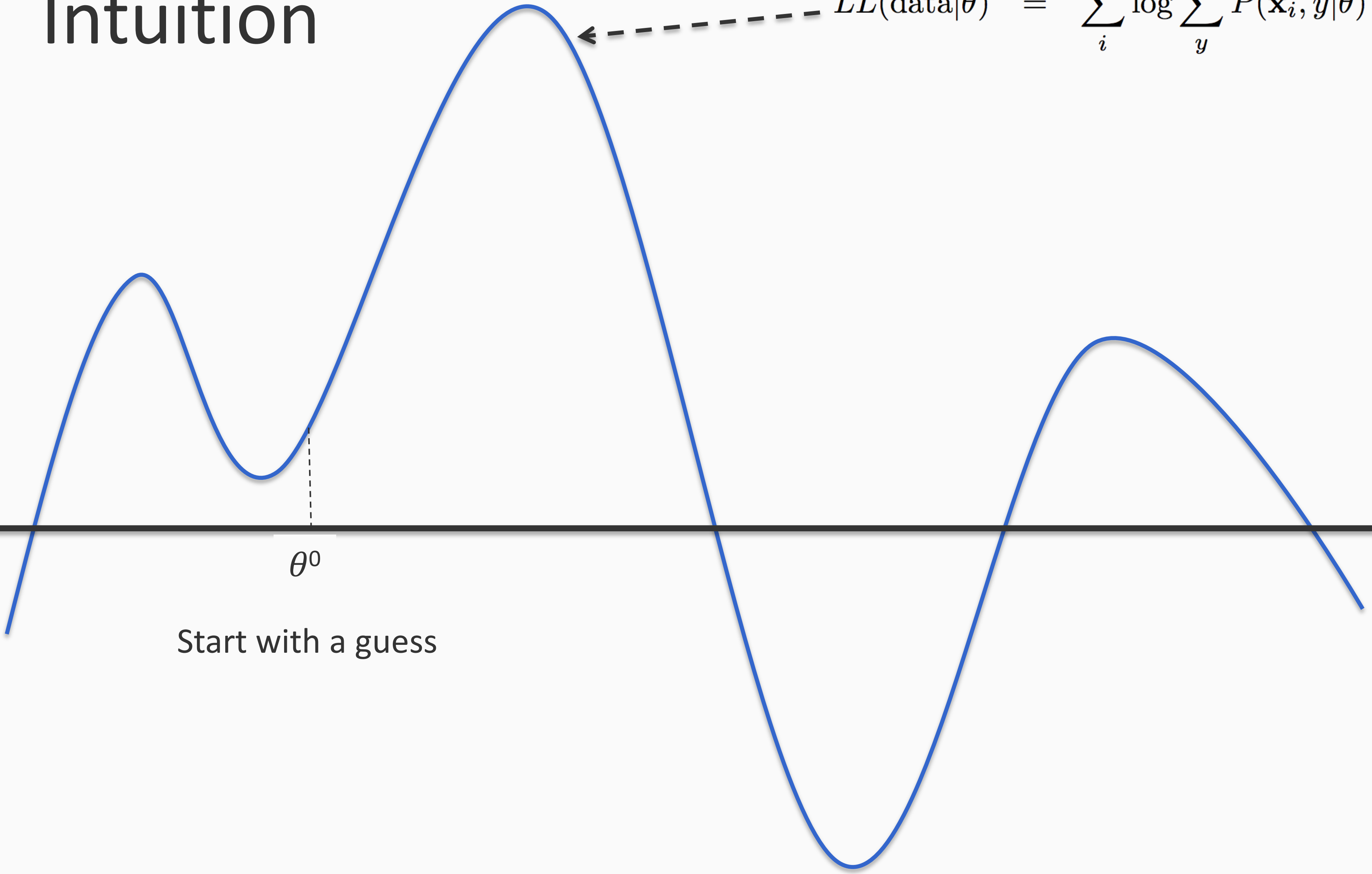
Intuition

$$LL(\text{data}|\theta) \quad = \quad \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

We want to maximize this function



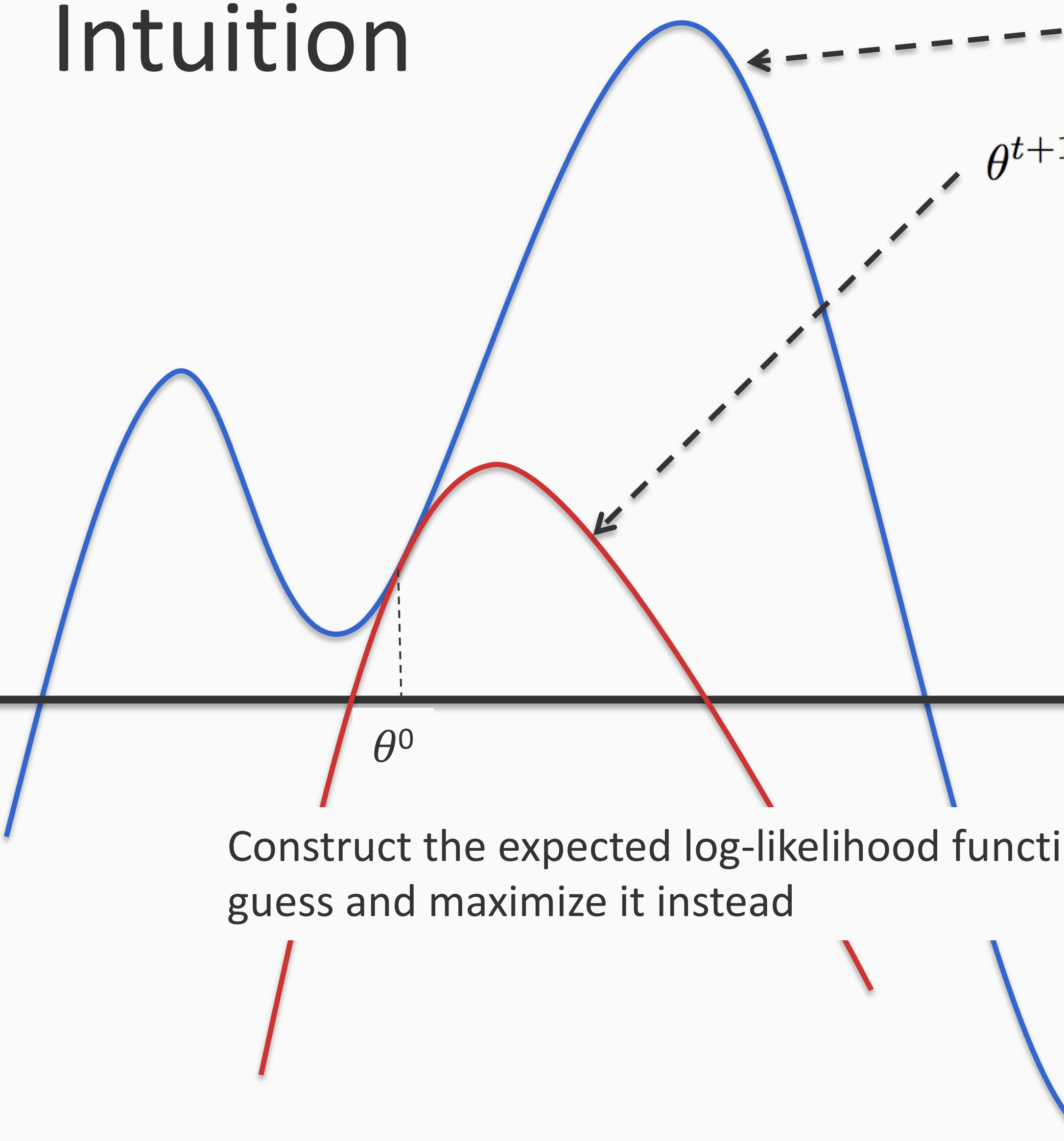
Intuition



$$LL(\text{data}|\theta) \;=\; \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

Start with a guess

Intuition



$$LL(\text{data}|\theta) \quad = \quad \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

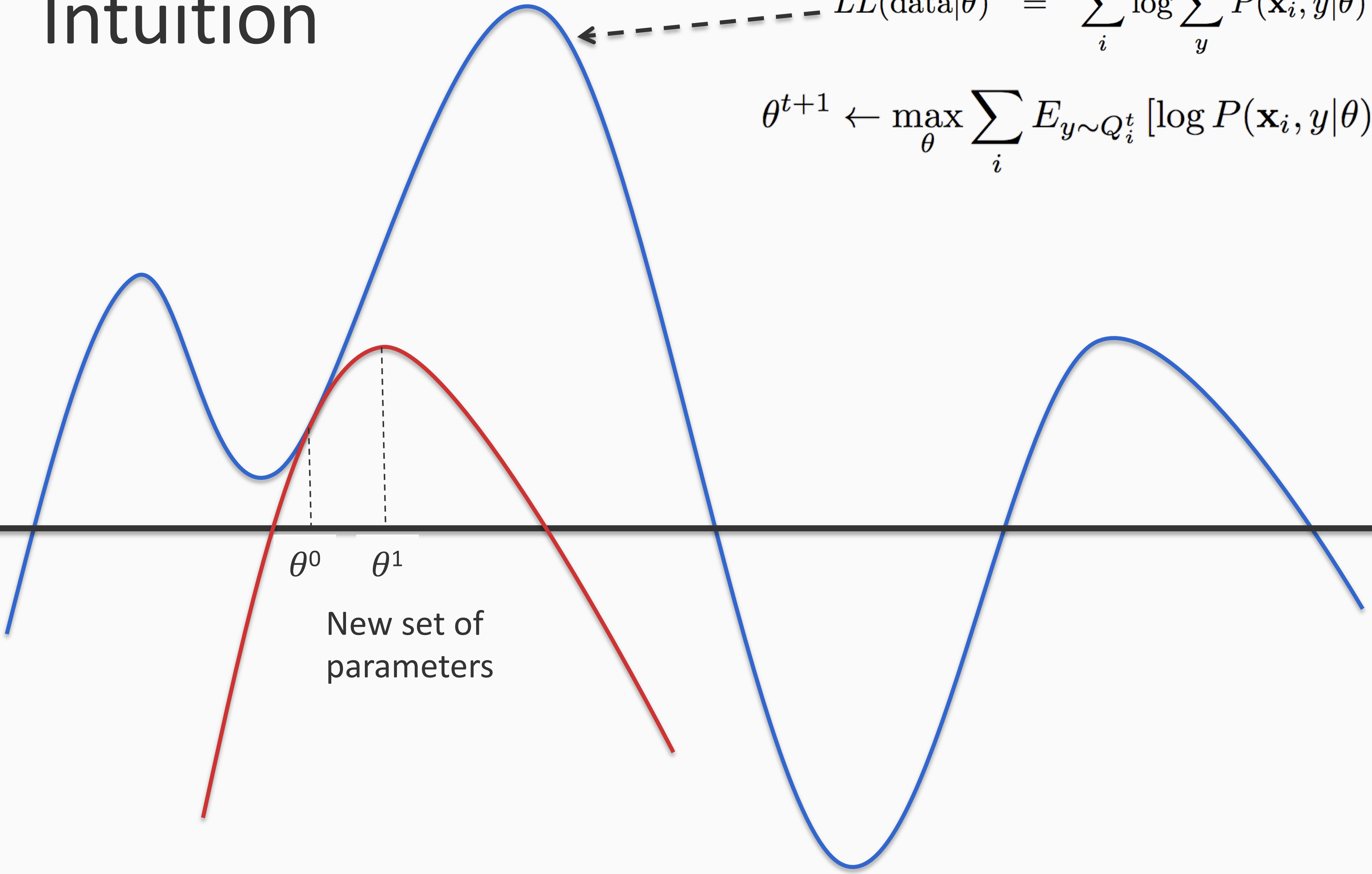
$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

Construct the expected log-likelihood function using the current guess and maximize it instead

Intuition

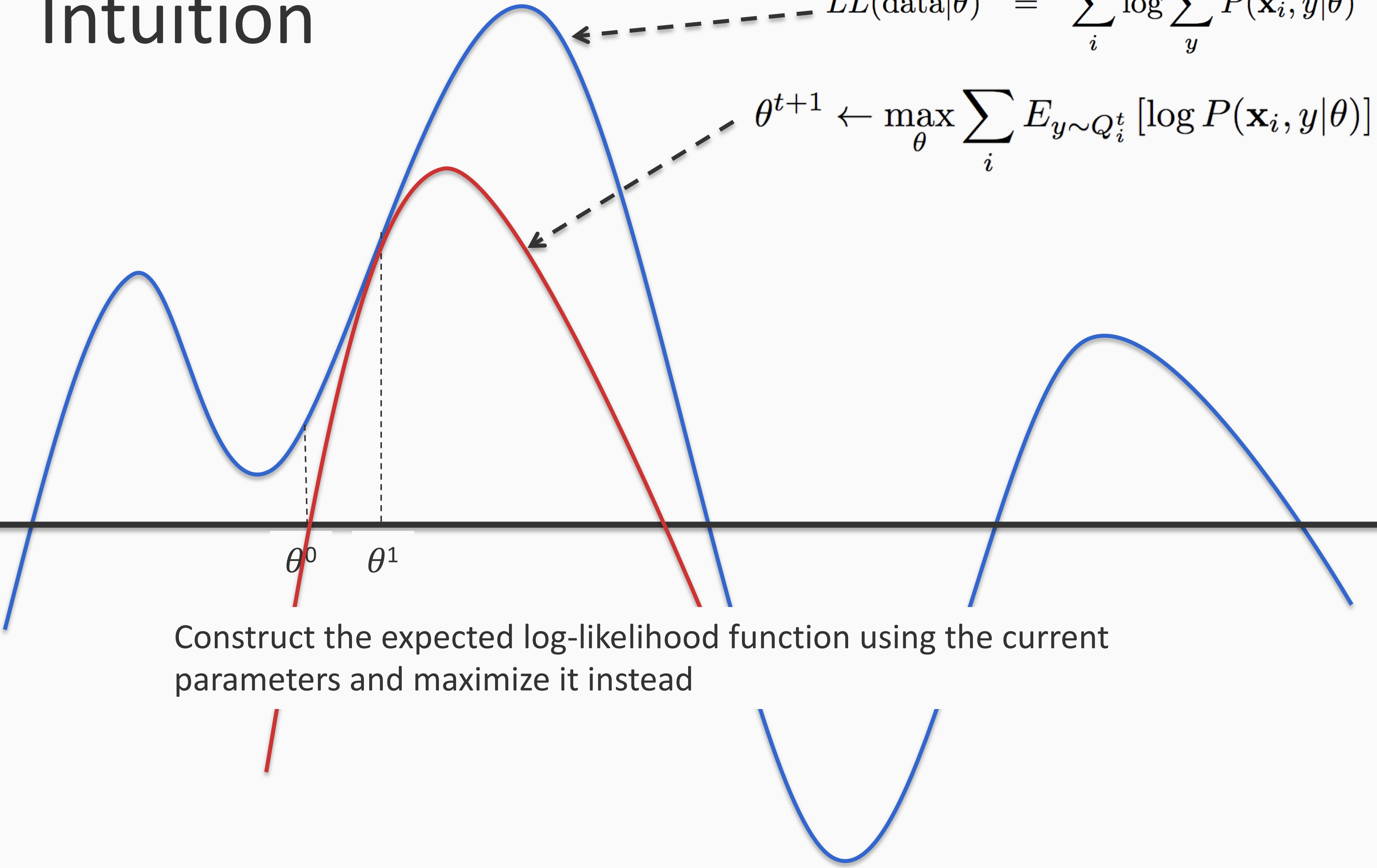
$$LL(\text{data}|\theta) = \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$



New set of parameters

Intuition



$$LL(\text{data}|\theta) = \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

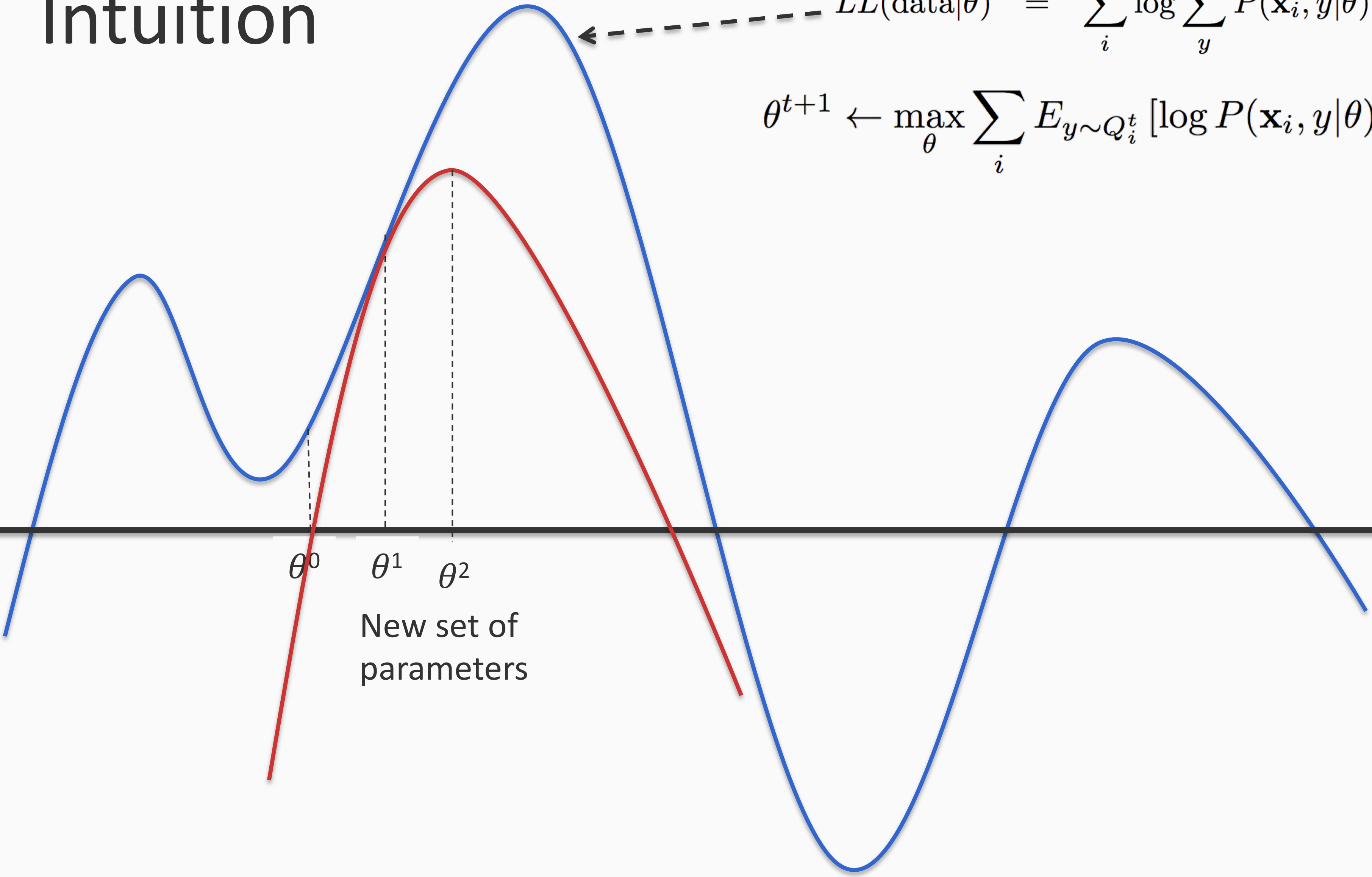
$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y\sim Q_i^t}\left[\log P(\mathbf{x}_i, y|\theta)\right]$$

Construct the expected log-likelihood function using the current parameters and maximize it instead

Intuition

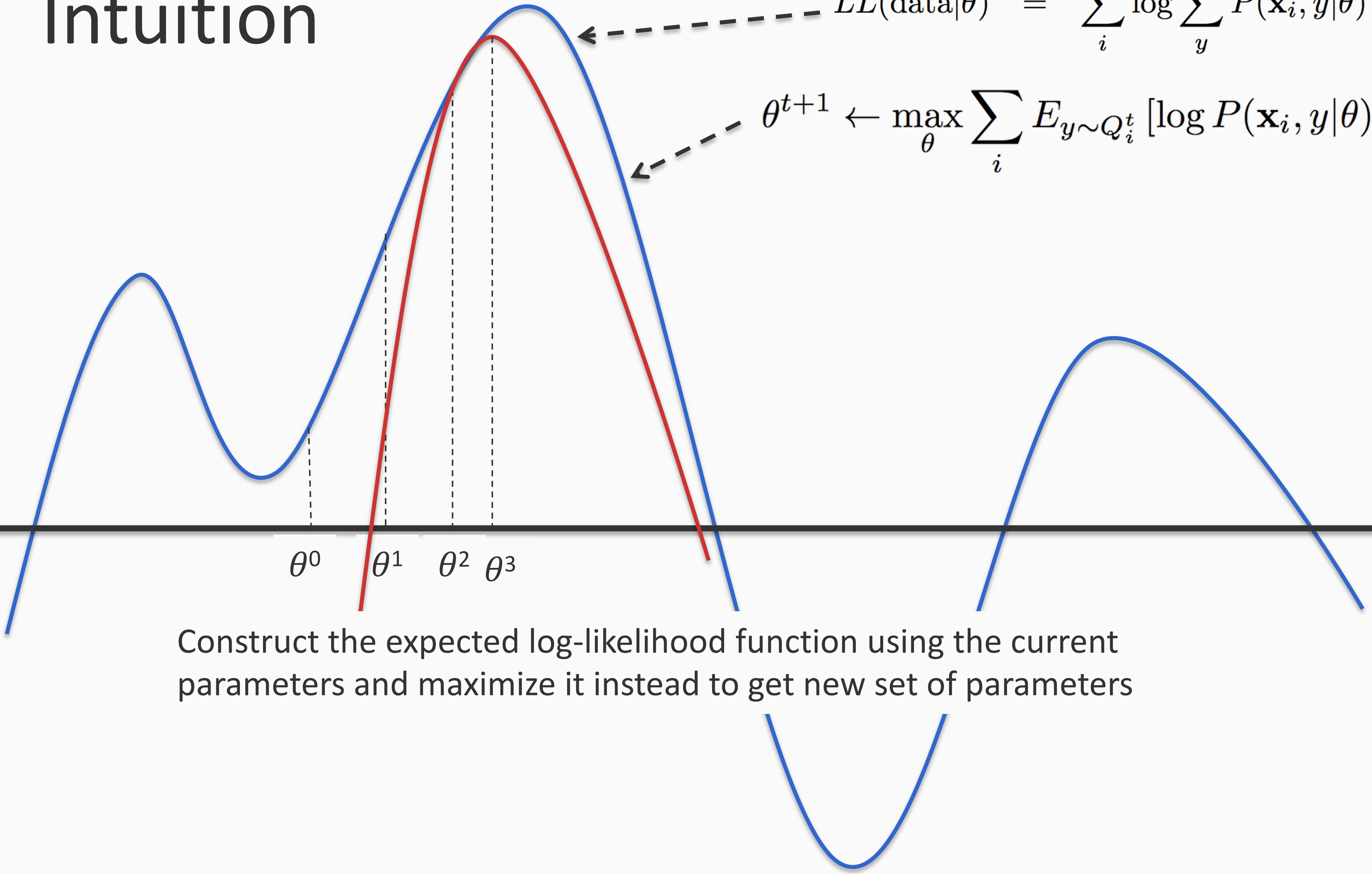
$$LL(\text{data}|\theta) \quad = \quad \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y\sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$



New set of parameters

Intuition



$$LL(\text{data}|\theta) = \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

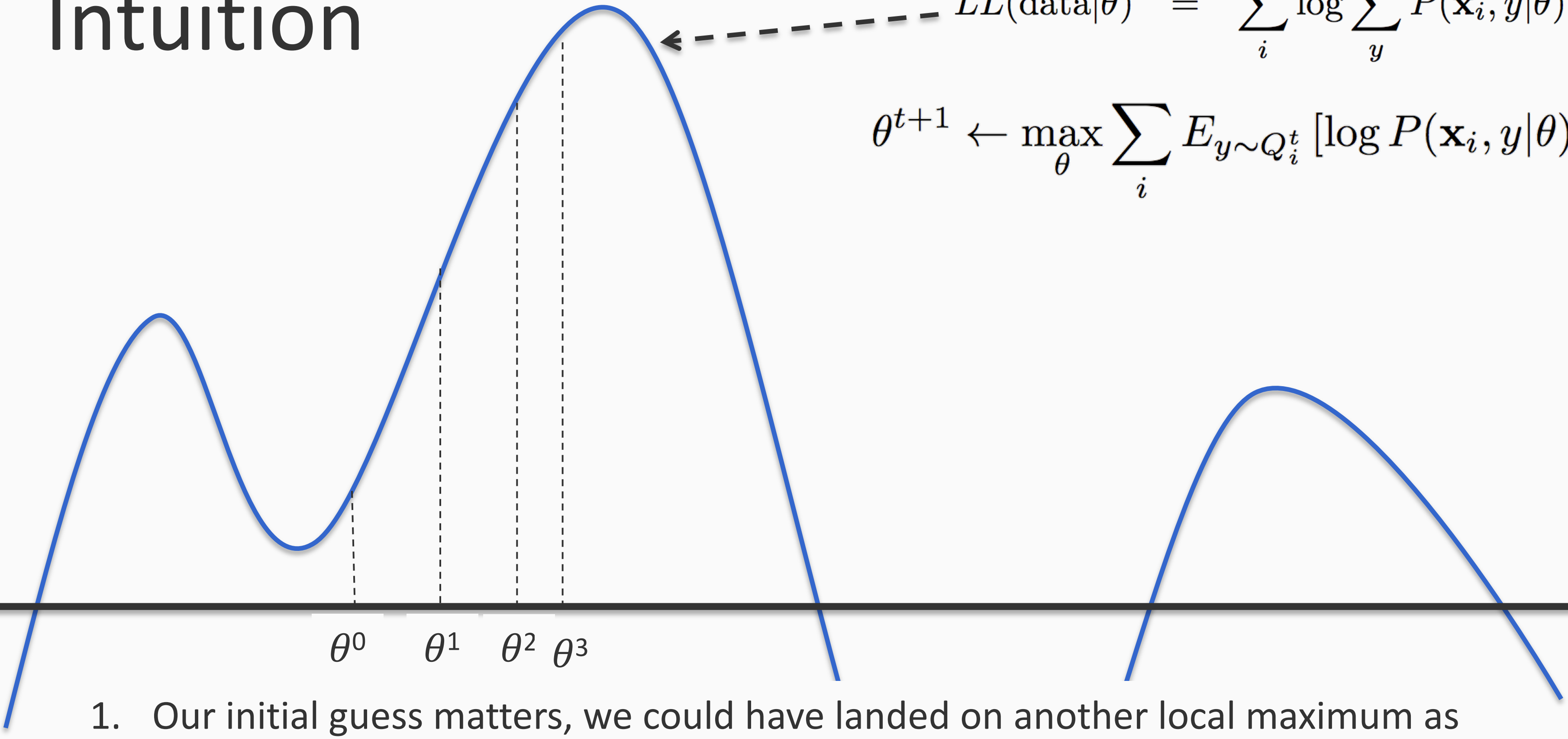
$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} [\log P(\mathbf{x}_i, y|\theta)]$$

Construct the expected log-likelihood function using the current parameters and maximize it instead to get new set of parameters

Intuition

$$LL(\text{data}|\theta) = \sum_i \log \sum_y P(\mathbf{x}_i, y|\theta)$$

$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y\sim Q_i^t}\left[\log P(\mathbf{x}_i, y|\theta)\right]$$



1. Our initial guess matters, we could have landed on another local maximum as well. But we will always end up at one of the local maxima

2. We are replacing our “difficult” optimization problems with a sequence of “easy” ones.

Comments about EM

- Will converge to a local maximum of the log-likelihood
  - Different initializations can give us different final estimates of probabilities

- How many iterations
  - Till convergence. Keep track of expected log likelihood across iterations and if the change is smaller than some ² then stop

- What we need to specify the learning algorithm
  - A task-specific definition of the probabilities
  - A way to solve the maximization (the M-step)

Checkpoint: Where are we

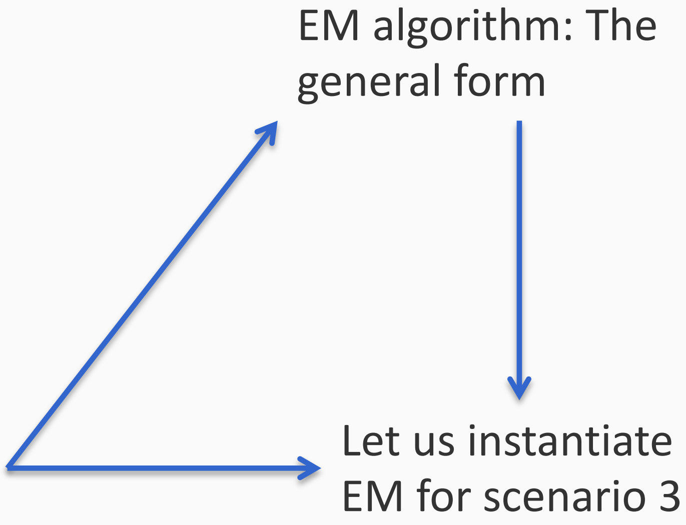
Learning with missing labels

Three coins example

Scenario 1: If we knew the (p, q, α) and coin 0's toss was hidden, we can estimate what it was from the rest of the observations

Scenario 2: If we had complete data, we could estimate all probabilities

Scenario 3: Can we estimate probabilities if coin 0 tosses were hidden?



EM algorithm: The general form

Let us instantiate EM for scenario 3

The three coin example

We have three coins

Coin 0: P(Heads) = α

Coin 1: P(Heads) = p

Coin 2: P(Heads) = q

Scenario 3: Toss coin 0 first. If heads, then toss coin 1 four times. If tails, then toss coin 2 four times

But we observe only the tosses produced by coins 1 and 2

Observations: HHHT, HTHT, HHHT, HTTH

From these observations, estimate the values of p, q and α?

$\mathbf{x}_i$ = one of these examples

y_i = the corresponding value of the coin 0's toss

The model

$$P(\mathbf{x}_i, y|p, q, \alpha) = \begin{cases} \alpha p^{k_i}(1-p)^{4-k_i} & \text{if } y = H \\ (1-\alpha)q^{k_i}(1-q)^{4-k_i} & \text{if } y = T \end{cases}$$

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

- Return final θ

E step

Data = {HHHT, HTHT, HHHT, HTTH}
$\mathbf{x}_i$ = one of these examples
y_i = the corresponding value of the coin 0's toss

The i^{th} observation $\mathbf{x}_i$ consists of 4 coin tosses, of which k_i are heads

Suppose we know the following estimates

Coin 0: P(Heads) = $\bar{\alpha}$
Coin 1: P(Heads) = $\bar{p}$
Coin 2: P(Heads) = $\bar{q}$

For an observation $\mathbf{x}_i$ we want to compute P(y_i | $\mathbf{x}_i$, current parameters)

Define $c_i^H = P(y_i = H|\mathbf{x}_i) \propto P(\mathbf{x}_i|y_i = H)P(y_i = H)$

E step

Data = {HHHT, HTHT, HHHT, HTTH}
x$_i$ = one of these examples
y$_i$ = the corresponding value of the coin 0's toss

The ith observation **x**$_i$ consists of 4 coin tosses, of which k$_i$ are heads

Suppose we know the following estimates
Coin 0: P(Heads) = $\bar{\alpha}$
Coin 1: P(Heads) = $\bar{p}$
Coin 2: P(Heads) = $\bar{q}$

For an observation **x**$_i$ we want to compute P(y$_i$ | **x**$_i$, current parameters)

Define $c_i^H = P(y_i = H|\mathbf{x}_i) \propto P(\mathbf{x}_i|y_i = H)P(y_i = H)$

$$\begin{aligned} P(\mathbf{x}_i|y_i = H)P(y_i = H) &= P(k_i \text{ heads}, 4 - k_i \text{ tails}|y_i = H)P(y_i = H) \\ &= \bar{p}^{k_i}\,(1-\bar{p})^{4-k_i}\,\bar{\alpha} \end{aligned}$$

E step

Data = {HHHT, HTHT, HHHT, HTTH}
$\mathbf{x}_i$ = one of these examples
y_i = the corresponding value of the coin 0's toss

The ith observation $\mathbf{x}_i$ consists of 4 coin tosses, of which k_i are heads

Suppose we know the following estimates

Coin 0: P(Heads) = $\bar{\alpha}$
Coin 1: P(Heads) = $\bar{p}$
Coin 2: P(Heads) = $\bar{q}$

For an observation $\mathbf{x}_i$ we want to compute P(y_i | $\mathbf{x}_i$, current parameters)

Define $c_i^H = P(y_i = H|\mathbf{x}_i) \propto P(\mathbf{x}_i|y_i = H)P(y_i = H)$

$$\bar{p}^{k_i}\,(1-\bar{p})^{4-k_i}\,\bar{\alpha}$$

E step

Data = {HHHT, HTHT, HHHT, HTTH}
x$_i$ = one of these examples
y$_i$ = the corresponding value of the coin 0's toss

The ith observation **x**$_i$ consists of 4 coin tosses, of which k$_i$ are heads

Suppose we know the following estimates
Coin 0: P(Heads) = $\bar{\alpha}$
Coin 1: P(Heads) = $\bar{p}$
Coin 2: P(Heads) = $\bar{q}$

For an observation **x**$_i$ we want to compute P(y$_i$ | **x**$_i$, current parameters)

Define $c_i^H = P(y_i = H|\mathbf{x}_i) \propto P(\mathbf{x}_i|y_i = H)P(y_i = H)$

$$\bar{p}^{k_i}\left(1-\bar{p}\right)^{4-k_i}\bar{\alpha}$$

$$c_i^H = \frac{\bar{p}^{k_i}\left(1-\bar{p}\right)^{4-k_i}\bar{\alpha}}{\bar{p}^{k_i}\left(1-\bar{p}\right)^{4-k_i}\bar{\alpha}+\bar{q}^{k_i}\left(1-\bar{q}\right)^{4-k_i}\left(1-\bar{\alpha}\right)}$$

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

These are the c's

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

- Return final θ

M step

Data = {HHHT, HTHT, HHHT, HTTH}

$\mathbf{x}_i$ = one of these examples

y_i = the corresponding value of the coin 0's toss

What we want $\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta) \right]$

M step

Data = {HHHT, HTHT, HHHT, HTTH}
x$_i$ = one of these examples
y$_i$ = the corresponding value of the coin 0's toss

What we want $\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$

Let us first write the log likelihood in terms of the parameters

$$P(\mathbf{x}_i, y|p, q, \alpha) = \begin{cases} \alpha p^{k_i}(1-p)^{4-k_i} & \text{if } y = H \\ (1-\alpha)q^{k_i}(1-q)^{4-k_i} & \text{if } y = T \end{cases}$$

$$\log P(\mathbf{x}_i, y|p, q, \alpha) = \begin{cases} \log \alpha + k_i \log(p) + (4-k_i)\log(1-p) & \text{if } y = H \\ \log(1-\alpha) + k_i \log(q) + (4-k_i)\log(1-q) & \text{if } y = T \end{cases}$$

M step

Data = {HHHT, HTHT, HHHT, HTTH}
$\mathbf{x}_i$ = one of these examples
y_i = the corresponding value of the coin 0's toss

What we want $\theta^{t+1} \leftarrow \max_{\theta} \sum_{i} E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta) \right]$

Expand the expectation

$$Q_i(H) \log P(\mathbf{x}_i, y = H | \theta) + Q_i(T) \log P(\mathbf{x}_i, y = T | \theta)$$

Substitute in the Q_i's

$$c_i^H \log P(\mathbf{x}_i, y = H | \theta) + \left(1 - c_i^H\right) \log P(\mathbf{x}_i, y = T | \theta)$$

M step

Data = {HHHT, HTHT, HHHT, HTTH}
x$_i$ = one of these examples
y$_i$ = the corresponding value of the coin 0's toss

What we want $\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$

Expand the expectation

$$Q_i(H) \log P(\mathbf{x}_i, y = H|\theta) + Q_i(T) \log P(\mathbf{x}_i, y = T|\theta)$$

Substitute in the Q$_i$'s

$$c_i^H \log P(\mathbf{x}_i, y = H|\theta) + \left(1 - c_i^H\right) \log P(\mathbf{x}_i, y = T|\theta)$$

We have all the pieces

1. The c$_i$'s are constants with respect to θ
2. We just wrote the log P's in terms of θ

M step

Data = {HHHT, HTHT, HHHT, HTTH}
x$_i$ = one of these examples
y$_i$ = the corresponding value of the coin 0's toss

What we want $\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta)\right]$

$$\max_{p,q,\alpha} \sum_i \left(c_i^H \log P(\mathbf{x}_i, y = H | p, q, \alpha) + \left(1 - c_i^H\right) \log P(\mathbf{x}_i, y = T | p, q, \alpha)\right)$$

We can now take derivatives with respect to p, q and α and set them to zero

Exercise: Do it

M step

Data = {HHHT, HTHT, HHHT, HTTH}

$\mathbf{x}_i$ = one of these examples

y_i = the corresponding value of the coin 0's toss

What we want $\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta)\right]$

The solution

$$\alpha^{t+1} = \frac{\sum_i c_i^H}{\text{number of examples}} \qquad p^{t+1} = \frac{\sum_i c_i^H \cdot k_i}{4 \sum_i c_i^H} \qquad q^{t+1} = \frac{\sum_i (1 - c_i^H) \cdot k_i}{4 \sum_i (1 - c_i^H)}$$

M step

Data = {HHHT, HTHT, HHHT, HTTH}

$\mathbf{x}_i$ = one of these examples

y_i = the corresponding value of the coin 0's toss

What we want $$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

The solution

$$\alpha^{t+1} = \frac{\sum_i c_i^H}{\text{number of examples}} \qquad p^{t+1} = \frac{\sum_i c_i^H \cdot k_i}{4 \sum_i c_i^H} \qquad q^{t+1} = \frac{\sum_i (1 - c_i^H) \cdot k_i}{4 \sum_i (1 - c_i^H)}$$

This has an intuitive interpretation

If c_i^H is an indicator for whether the ith toss of coin zero is a head, then

$$\alpha^{t+1} = \frac{\text{number of heads for coin zero}}{\text{number of examples}}$$

$$p^{t+1} = \frac{\text{number of heads for coin 1}}{\text{number of tosses of coin 1}}$$

$$q^{t+1} = \frac{\text{number of heads for coin 2}}{\text{number of tosses of coin 2}}$$

M step

Data = {HHHT, HTHT, HHHT, HTTH}

$\mathbf{x}_i$ = one of these examples

y_i = the corresponding value of the coin 0's toss

What we want $$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta) \right]$$

The solution

$$\alpha^{t+1} = \frac{\sum_i c_i^H}{\text{number of examples}} \qquad p^{t+1} = \frac{\sum_i c_i^H \cdot k_i}{4 \sum_i c_i^H} \qquad q^{t+1} = \frac{\sum_i (1 - c_i^H) \cdot k_i}{4 \sum_i (1 - c_i^H)}$$

This has an intuitive interpretation

If c_i^H is an indicator for whether the ith toss of coin zero is a head, then

$$\alpha^{t+1} = \frac{\text{number of heads for coin zero}}{}$$

$$p^{t+1} = \frac{\text{number of heads for coin 1}}{\text{number of tosses of coin 1}}$$

$$q^{t+1} = \frac{\text{oin 2}}{\text{number of tosses of coin 2}}$$

Instead, the probabilities end up being treated like *soft counts*

Expectation Maximization

- Initialize the parameters θ^0
- Repeat until convergence (t = 1, 2, …)
  - E-Step: For every example $\mathbf{x}_i$, estimate for every y

$$Q_i^t(y) = P(y|\mathbf{x}_i, \theta^t)$$

These are the c's

  - M-Step: Find θ^{t+1} by maximizing with respect to θ

$$\theta^{t+1} \leftarrow \max_\theta \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

Analytically estimate the value of the next θ

- Return final θ

EM for Naïve Bayes

The setting

- Input: features $\mathbf{x} \in \{0,1\}^d$
- Output: $y \in \{0, 1\}$
- Dataset: $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_m\}$, m unlabeled examples

The model $$P(\mathbf{x}, y) = P(y) \prod_j P(x_j|y)$$

EM for Naïve Bayes

The setting

- Input: features $\mathbf{x} \in \{0,1\}^d$
- Output: $y \in \{0, 1\}$
- Dataset: $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_m\}$, m unlabeled examples

The model $$P(\mathbf{x}, y) = P(y) \prod_j P(x_j | y)$$

- Prior: $P(y = 1) = p$ and $P(y = 0) = 1 - p$

- Likelihood for each feature given a label
  - $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$
  - $P(x_j = 1 \mid y = 0) = b_j$ and $P(x_j = 0 \mid y = 0) = 1 - b_j$

EM for Naïve Bayes

The setting

- Input: features **x** ∈ $\{0,1\}^d$
- Output: y ∈ {0, 1}
- Dataset: {$\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, ⋯, $\mathbf{x}_m$}, m unlabeled examples

The model $$P(\mathbf{x}, y|\theta) = P(y|\theta) \prod_j P(x_j|y, \theta)$$

$$\theta = (p, a_1, a_2, \cdots, a_d, b_1, b_2, \cdots, b_d)$$

The E-step

Goal: Suppose we have a current estimate of θ, compute $Q_i(y) = P(y \mid \mathbf{x}_i, \theta)$ for each example

$$P(y=1|\mathbf{x}_i,\theta) = \frac{P(y=1,\mathbf{x}_i|\theta)}{P(y=1,\mathbf{x}_i|\theta)+P(y=0,\mathbf{x}_i|\theta)}$$

The E-step

Goal: Suppose we have a current estimate of θ, compute $Q_i(y) = P(y \mid \mathbf{x}_i, \theta)$ for each example

$$P(y=1|\mathbf{x}_i,\theta) = \frac{P(y=1,\mathbf{x}_i|\theta)}{P(y=1,\mathbf{x}_i|\theta) + P(y=0,\mathbf{x}_i|\theta)}$$

And we know how to compute these using our model

$$P(\mathbf{x},y|\theta) = P(y|\theta)\prod_j P(x_j|y,\theta)$$

The M-Step

Goal $$\theta^{t+1} \leftarrow \max_{\theta} \sum_{i} E_{y \sim Q_i^t} [\log P(\mathbf{x}_i, y | \theta)]$$

Step 1: Expand log P(x_i, y | θ) in terms of p, a's and b's

Step 2: Substitute in Q_i to write down the full expectation

Step 3: Take derivative with respect to each p, a_j and b_j

Step 4: Set derivatives to zero to get a new estimate for p, a_j and b_j

Exercise: Work out these steps on paper

The M-Step

Goal $$\theta^{t+1} \leftarrow \max_{\theta} \sum_{i} E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta) \right]$$

Taking derivatives and setting to zero gives

$$p = \frac{\text{SoftCount}\,(y = 1)}{\text{SoftCount}\,(y = 1) + \text{SoftCount}\,(y = 0)}$$

$$a_j = \frac{\text{SoftCount}\,(y = 1, x_j = 1)}{\text{SoftCount}\,(y = 1)}$$

$$b_j = \frac{\text{SoftCount}\,(y = 0, x_j = 1)}{\text{SoftCount}\,(y = 0)}$$

$P(y = 1) = p$ $P(x_j = 1 \mid y = 1) = a_j$ $P(x_j = 1 \mid y = 0) = b_j$

The M-Step

Goal $$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t}\left[\log P(\mathbf{x}_i, y|\theta)\right]$$

Taking derivatives and setting to zero gives

$$p = \frac{\text{SoftCount}\,(y=1)}{\text{SoftCount}\,(y=1) + \text{SoftCount}\,(y=0)}$$

$$a_j = \frac{\text{SoftCount}\,(y=1, x_j=1)}{\text{SoftCount}\,(y=1)}$$

$$b_j = \frac{\text{SoftCount}\,(y=0, x_j=1)}{\text{SoftCount}\,(y=0)}$$

$$\text{SoftCount}\,(y=1) = \sum_i P(y=1|\mathbf{x}_i, \theta^t)$$

$$\text{SoftCount}\,(y=1, x_j=1) = \sum_i P(y=1|\mathbf{x}_i, \theta^t)\,[x_{ij}=1]$$

And so on…

$P(y = 1) = p$ $P(x_j = 1 \mid y = 1) = a_j$ $P(x_j = 1 \mid y = 0) = b_j$

The M-Step: Intuition

$$p = \frac{\cancel{\text{Soft}}\text{Count}\left(y = 1\right)}{\cancel{\text{Soft}}\text{Count}\left(y = 1\right) + \cancel{\text{Soft}}\text{Count}\left(y = 0\right)}$$

$$a_j = \frac{\cancel{\text{Soft}}\text{Count}\left(y = 1, x_j = 1\right)}{\cancel{\text{Soft}}\text{Count}\left(y = 1\right)}$$

$$b_j = \frac{\cancel{\text{Soft}}\text{Count}\left(y = 0, x_j = 1\right)}{\cancel{\text{Soft}}\text{Count}\left(y = 0\right)}$$

If we had fully labeled data, we could learn the Naïve Bayes classifier using counts.

The M-Step: Intuition

$$p = \frac{\text{SoftCount}\left(y = 1\right)}{\text{SoftCount}\left(y = 1\right) + \text{SoftCount}\left(y = 0\right)}$$

$$a_j = \frac{\text{SoftCount}\left(y = 1, x_j = 1\right)}{\text{SoftCount}\left(y = 1\right)}$$

$$b_j = \frac{\text{SoftCount}\left(y = 0, x_j = 1\right)}{\text{SoftCount}\left(y = 0\right)}$$

If we had fully labeled data, we could learn the Naïve Bayes classifier using counts.

Since we can not count, we keep the uncertainty by allowing fractional counts

$$\text{SoftCount}\left(y = 1\right) = \sum_i P(y = 1|\mathbf{x}_i, \theta^t)$$

P(y=1|xi, θ^t) behaves like the indicator function [y=1], except it allows fractional values

EM Summary

- A general procedure for learning with unobserved variables
  - An iterative algorithm that converges to a local maximum of the likelihood function

- A family of algorithms
  - Specific instantiation depends on what probabilistic model you are using
    - You have to derive update rules for your own model
  - Instantiated the algorithm for a mixture of Bernoulli distributions

- Very useful in practice. But can be sensitive to
  - Choice of the probabilistic model
  - Initialization

This lecture

- Semi-supervised/Unsupervised learning
- Expectation-Maximization
- Variants of EM
  - K-Means

Hard EM

Many variants of EM exist: MCMC EM, Variational EM, Generalized EM,…
These tweak on the same general idea.

E-step in EM estimates the probability of the hidden variable using the current parameters

- $Q_i(y) = P(y \mid \mathbf{x}, \theta^t)$

Hard EM: Instead of estimating the probability, we find the most probable assignment and use that instead in the M step

- Equivalently:
  - Find the most probable value of y
  - Create a distribution $Q_i(y)$ that this value probability 1 and everything else zero

Mixture of Gaussians Or: Gaussian Mixture Model

Setting

- Examples **x** 2 $<^d$
- K possible labels y 2 {l_1, l_2, ···, l_K}

Generative model

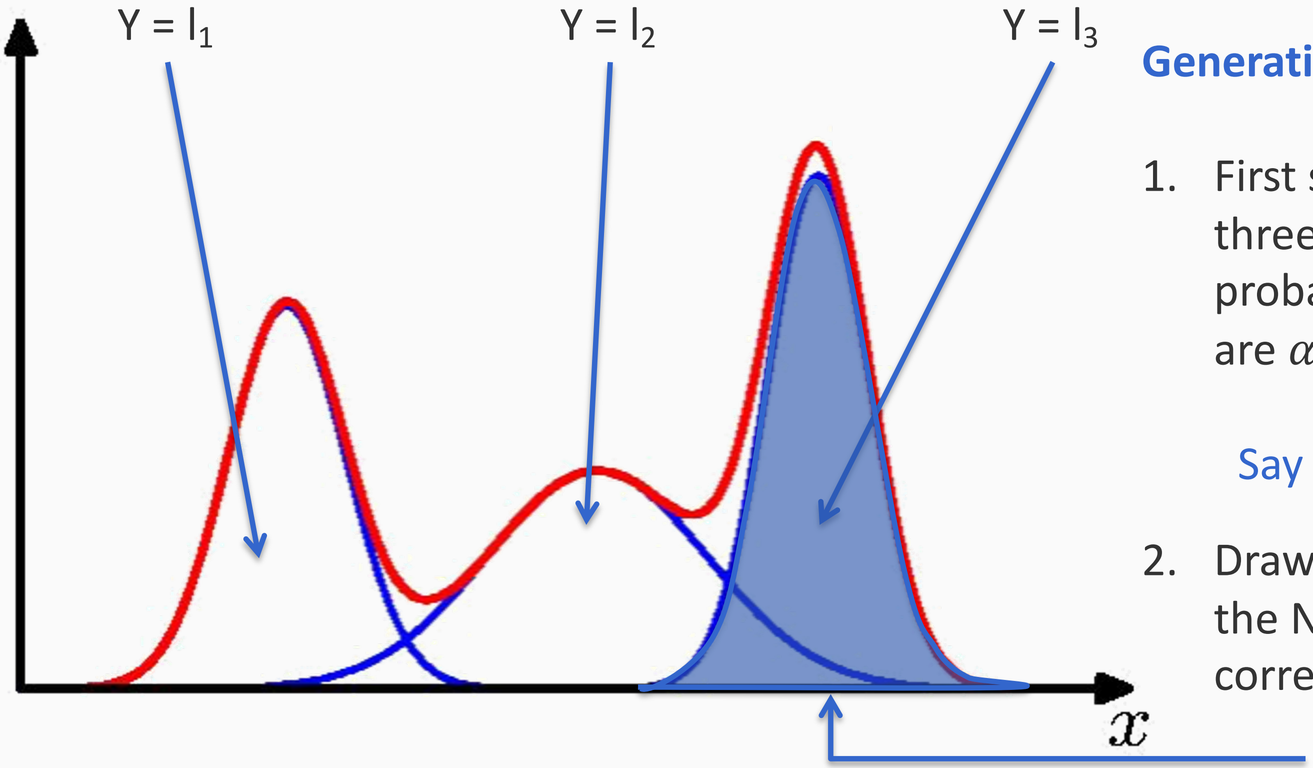
- First draw a label from a multinomial distribution

  P(y = l_i) = α_i

- Then, the example **x** is drawn from a d-dimensional Normal distribution with mean $^1{}_i$ and variance §$_i$

$^1{}_i$ is a d dimensional vector and §$_i$ is a d£ d matrix

Example: 1 dimensional case (Three Gaussians)



Generating an example:

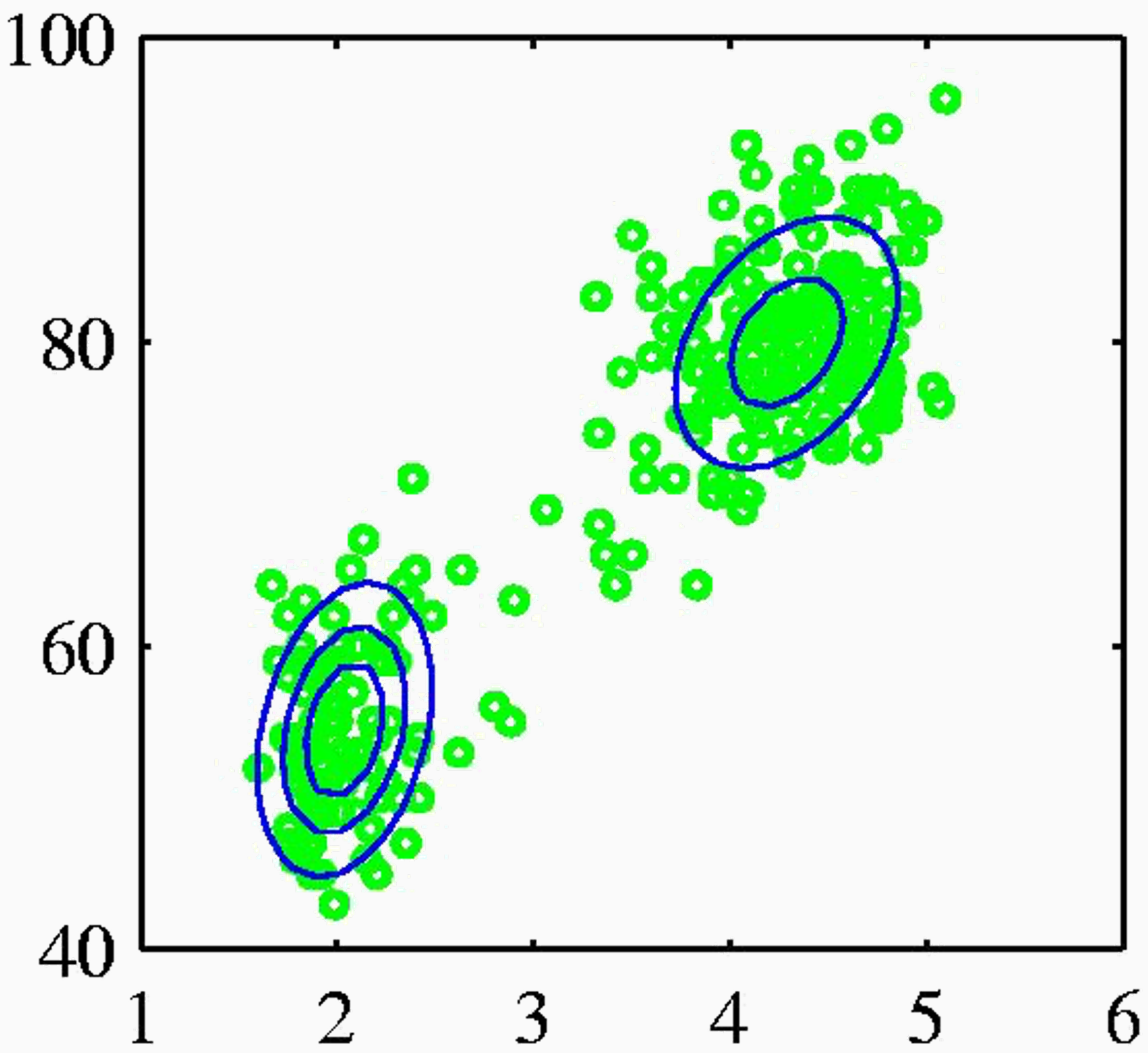
1. First sample a Y. Roll a three sided die where probability of l_1, l_2 and l_3 are α_1, α_2 and α_3 resp.

   Say the die picks Y = l_3

2. Draw a point x from the the Normal distribution corresponding to Y = l_3

Say this point

Example: 2 dimensional case



Likelihood of a point

Suppose we have a point x whose label is l_i

Likelihood of this point is

$$P(l_i)\ P(x \mid y = l_i) = \alpha_i \boxed{N(x;\mu_i,\sigma_i)}$$

Probability density for a d dimensional Normal distribution with mean μ_i and standard deviation σ_i

Unsupervised learning

Suppose we only have a collection of points and we want to assign labels to them one of K possible labels $\{l_1, l_2, \cdots, l_K\}$

Input: $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, each $\mathbf{x}_i$ a real valued, d dimensional vector

Goal: Label each input point

Assumption: Suppose the points were generated according to the Gaussian mixture model

$$P(l_i)\, P(x \mid y = l_i) = \alpha_i\, N(x; \mu_i, \sigma_i)$$

Unsupervised learning

Input: $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, each $\mathbf{x}_i$ a real valued, d dimensional vector

Goal: Label each input point

Assumption: Suppose the points were generated according to the Gaussian mixture model

$$P(l_i)\ P(x \mid y = l_i) = \alpha_i\ N(x; \mu_i, \sigma_i)$$

(For now), simplify the problem by assuming that α_i are all equal to 1/K and σ_i are all the identity matrix

- All labels are equally likely
- The jth input feature for label l_i is drawn independently from a Gaussian with mean μ_{ij} and variance one

Mixture of Gaussians

Given an example (x, y), we can compute its likelihood under the model

$$p(\mathbf{x}, y = l|\theta) = \frac{1}{K} \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{||\mathbf{x} - \mu_l||^2}{2}\right)$$

We only have the points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$

E-step: Given an estimate of the ^{1}’s

For each input point, probability that it belongs to a particular label

$$P(y = l | \mathbf{x}_i, \text{parameters}) \quad = \quad \frac{P(y = l, \mathbf{x}_i | \text{parameters})}{\sum_{l'=1}^{K} P(y = l', \mathbf{x}_i | \text{parameters})}$$

Parameters = all the ^{1}’s

E-step: Given an estimate of the ^{1}’s

For each input point, probability that it belongs to a particular label

$$P(y = l|\mathbf{x}_i, \text{parameters}) = \frac{P(y = l, \mathbf{x}_i|\text{parameters})}{\displaystyle\sum_{l'=1}^{K} P(y = l', \mathbf{x}_i|\text{parameters})}$$

Because we assume that the points are generated from a Gaussian mixture model

$$= \frac{\frac{1}{K} N(\mathbf{x}_i; \mu_l, I)}{\displaystyle\sum_{l'=1}^{K} \frac{1}{K} N(\mathbf{x}_i; \mu_{l'}, I)}$$

E-step: Given an estimate of the ^{1}’s

For each input point, probability that it belongs to a particular label

$$\begin{aligned}
P(y=l|\mathbf{x}_i,\text{parameters}) &= \frac{P(y=l,\mathbf{x}_i|\text{parameters})}{\sum_{l'=1}^{K} P(y=l',\mathbf{x}_i|\text{parameters})} \\
&= \frac{\cancel{\frac{1}{K}}N(\mathbf{x}_i;\mu_l,I)}{\sum_{l'=1}^{K}\cancel{\frac{1}{K}}N(\mathbf{x}_i;\mu_{l'},I)} \\
&= \frac{\exp\left(-\frac{1}{2}||\mathbf{x}_i-\mu_l||^2\right)}{\sum_{l'=1}^{K}\exp\left(-\frac{1}{2}||\mathbf{x}_i-\mu_{l'}||^2\right)}
\end{aligned}$$

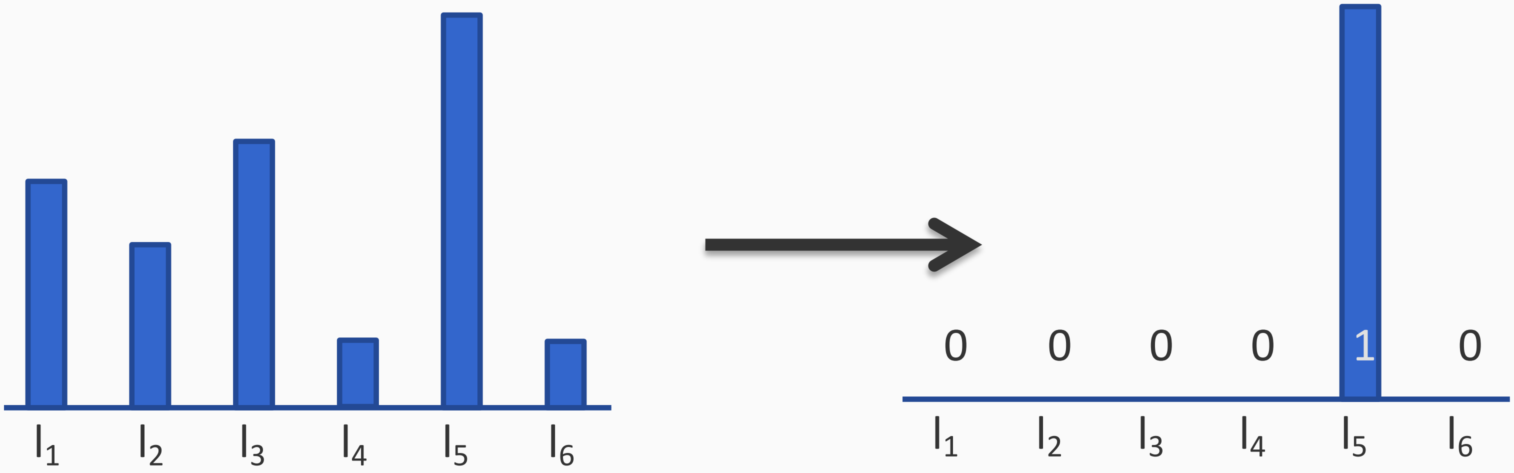
E-step: Given an estimate of the ^{1}'s

For each input point, probability that it belongs to a particular label

$$P(y = l|\mathbf{x}_i, \text{parameters}) = \frac{\exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_l||^2\right)}{\sum_{l'=1}^{K} \exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_{l'}||^2\right)}$$

This is a distribution over the labels for point $\mathbf{x}_i$

Hard EM uses only the highest scoring label for the M step



E-Step in hard EM

For each point, assign its label to be the one with the highest probability according to the current parameters

E-Step in hard EM (for mixture of gaussians)

For each point, assign its label to be the one with the highest probability according to the current parameters

$$P(y = l|\mathbf{x}_i, \text{parameters}) = \frac{\exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_l||^2\right)}{\sum_{l'=1}^{K} \exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_{l'}||^2\right)}$$

$$\text{Label for } \mathbf{x}_i \quad = \quad \arg\max_{l} P(y = l|\mathbf{x}, \text{parameters})$$

E-Step in hard EM (for mixture of gaussians)

For each point, assign its label to be the one with the highest probability according to the current parameters

$$P(y = l|\mathbf{x}_i, \text{parameters}) = \frac{\exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_l||^2\right)}{\sum_{l'=1}^{K} \exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_{l'}||^2\right)}$$

$$\begin{aligned}\text{Label for } \mathbf{x}_i \quad &= \quad \arg\max_l P(y = l|\mathbf{x}, \text{parameters}) \\ &= \quad \arg\max_l \exp\left(-\frac{1}{2}||\mathbf{x}_i - \mu_l||^2\right) \\ &= \quad \arg\min_k ||\mathbf{x}_i - \mu_l||^2\end{aligned}$$

E-Step in hard EM (for mixture of gaussians)

For each point, assign its label to be the one with the highest probability according to the current parameters

$$\text{Label for } \mathbf{x}_i = \arg\min_k ||\mathbf{x}_i - \mu_l||^2$$

Or equivalently: Find the label, whose mean is closest in Euclidean distance to the point

Let us call this label y_i for the point $\mathbf{x}_i$

M-step for mixture of Gaussians

Goal $$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta)\right]$$

Step 1: Let us write down log P($\mathbf{x}_i$, y | parameters)

$$p(\mathbf{x}, y = l|\theta) = \frac{1}{K} \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{||\mathbf{x} - \mu_l||^2}{2}\right)$$

This comes from the the definition of our model

$$\log p(\mathbf{x}, y = l|\theta) = -\log(K\sqrt{2\pi}) - \frac{1}{2}||\mathbf{x}_i - \mu_l||^2$$

M-step for mixture of Gaussians

Goal $$\theta^{t+1} \leftarrow \max_{\theta} \sum_{i} E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta)\right]$$

Step 2: Let us write down E[log P($\mathbf{x}_i$, y | parameters)]

In the general case, we will have a distribution over the labels $Q_i(y)$

For hard EM, this distribution is zero everywhere except at y_i, where it is one

M-step for mixture of Gaussians

Goal $\theta^{t+1} \leftarrow \max_{\theta} \sum_{i} E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y|\theta) \right]$

Step 3: Maximization

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_{i} - \log(K\sqrt{2\pi}) - \frac{1}{2}||\mathbf{x}_i - \mu_{y_i}||^2$$

M-step for mixture of Gaussians

Goal $$\theta^{t+1} \leftarrow \max_{\theta} \sum_i E_{y \sim Q_i^t} \left[\log P(\mathbf{x}_i, y | \theta) \right]$$

Step 3: Maximization

$$\theta^{t+1} \leftarrow \max_{\theta} \sum_i - \log(K\sqrt{2\pi}) - \frac{1}{2}||\mathbf{x}_i - \mu_{y_i}||^2$$

Solution:

$^{1}_{l}$ = Mean of points assigned with label l

Hard EM for GMMs: Full algorithm

Input: A set of d-dimensional points {$\mathbf{x}_1$, $\mathbf{x}_2$, $\cdots$, $\mathbf{x}_m$} and K, the number of labels

1. Initialize the means $^{1}{}_{1}$, $^{1}{}_{2}$, $\cdots$, $^{1}{}_{K}$ randomly
   - these are d dimensional vectors

2. Loop:
   1. Label each point as the mean closest to it

      $$\text{Label for } \mathbf{x}_i = \arg\min_k ||\mathbf{x}_i - \mu_l||^2$$

   2. For every label l:
      - Re-compute the mean $^{1}{}_{l}$ as the average of all points that were assigned to it

3. Return the final labels

Hard EM for GMMs: Full algorithm

Input: A set of d-dimensional points {$\mathbf{x}_1$, $\mathbf{x}_2$, ···, $\mathbf{x}_m$} and K, the number of labels

1. Initialize the means $^1{}_1$, $^1{}_2$, ···, $^1{}_K$ randomly
   – these are d dimensional vectors

2. Loop:
   1. Label each point as the mean closest to it

      $$\text{Label for } \mathbf{x}_i = \arg\min_k ||\mathbf{x}_i - \mu_l||^2$$

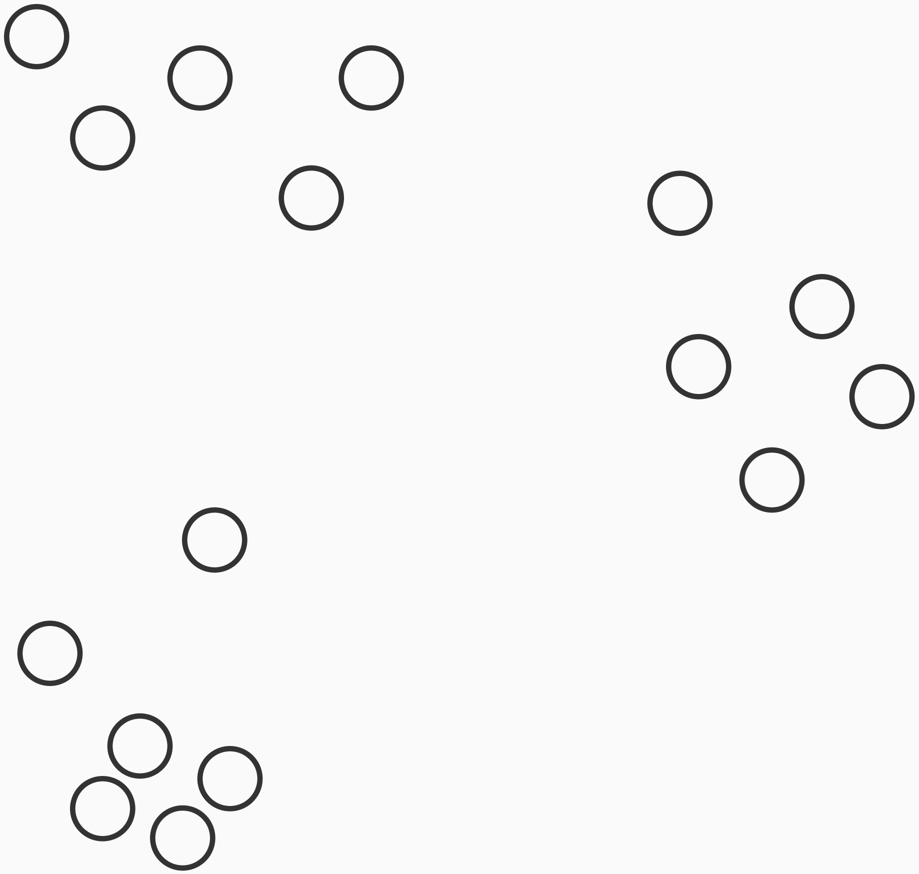
   2. For every label l:
      - Re-compute the mean $^1{}_l$ as the average of all points that were assigned to it

3. Return the final labels

This is the popular K-Means algorithm for clustering

K means example

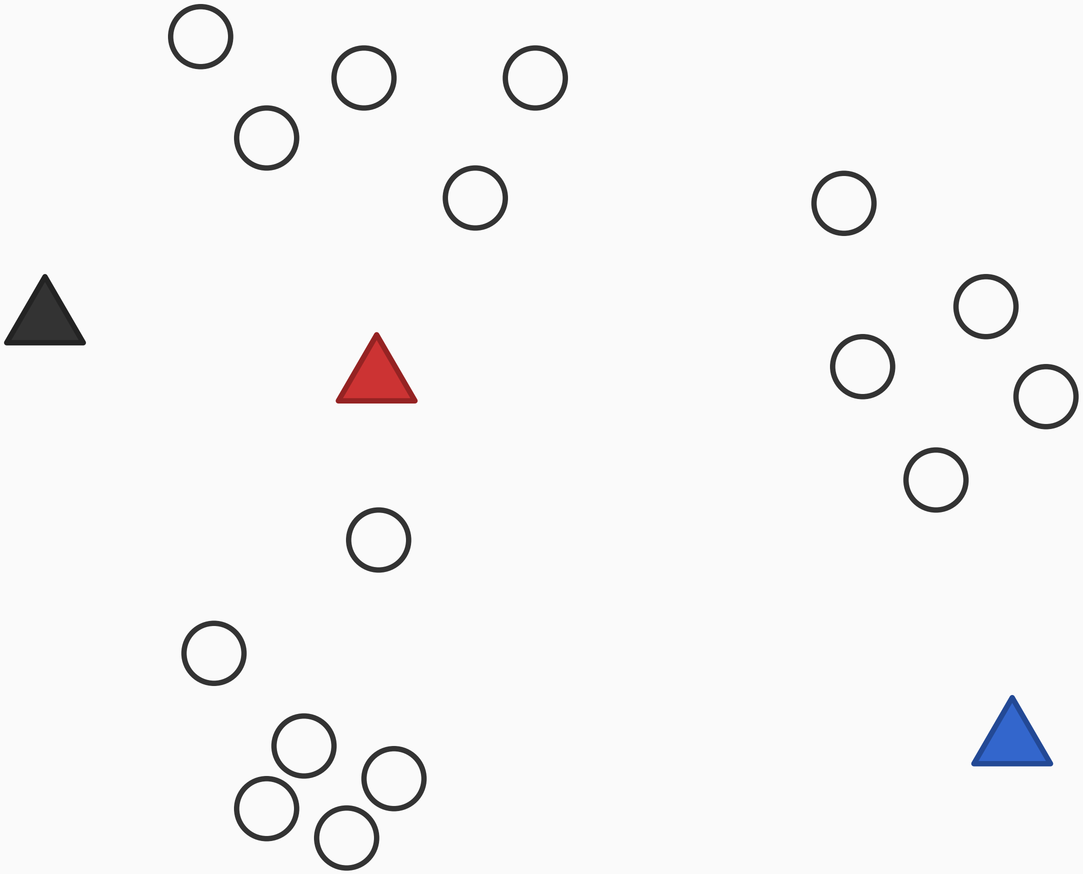
Suppose K = 3



K means example

Suppose K = 3

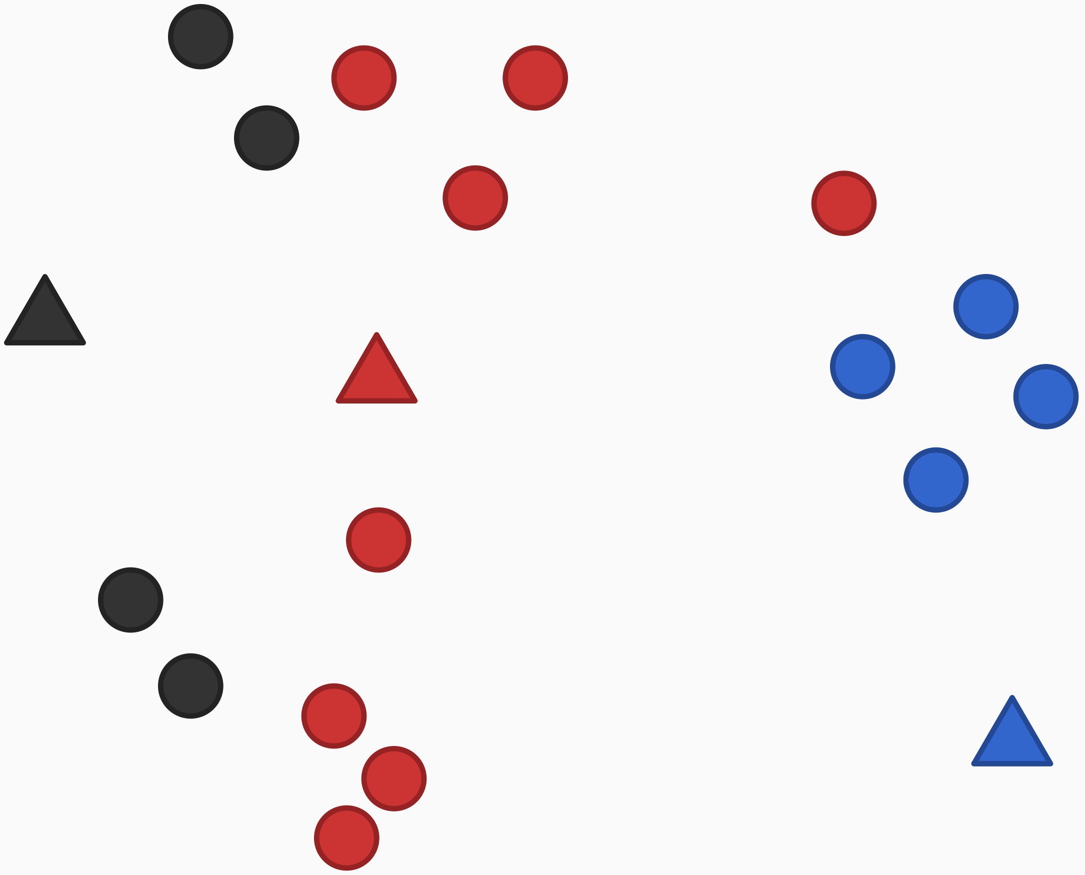
Initialize: Pick random means



K means example

Suppose K = 3

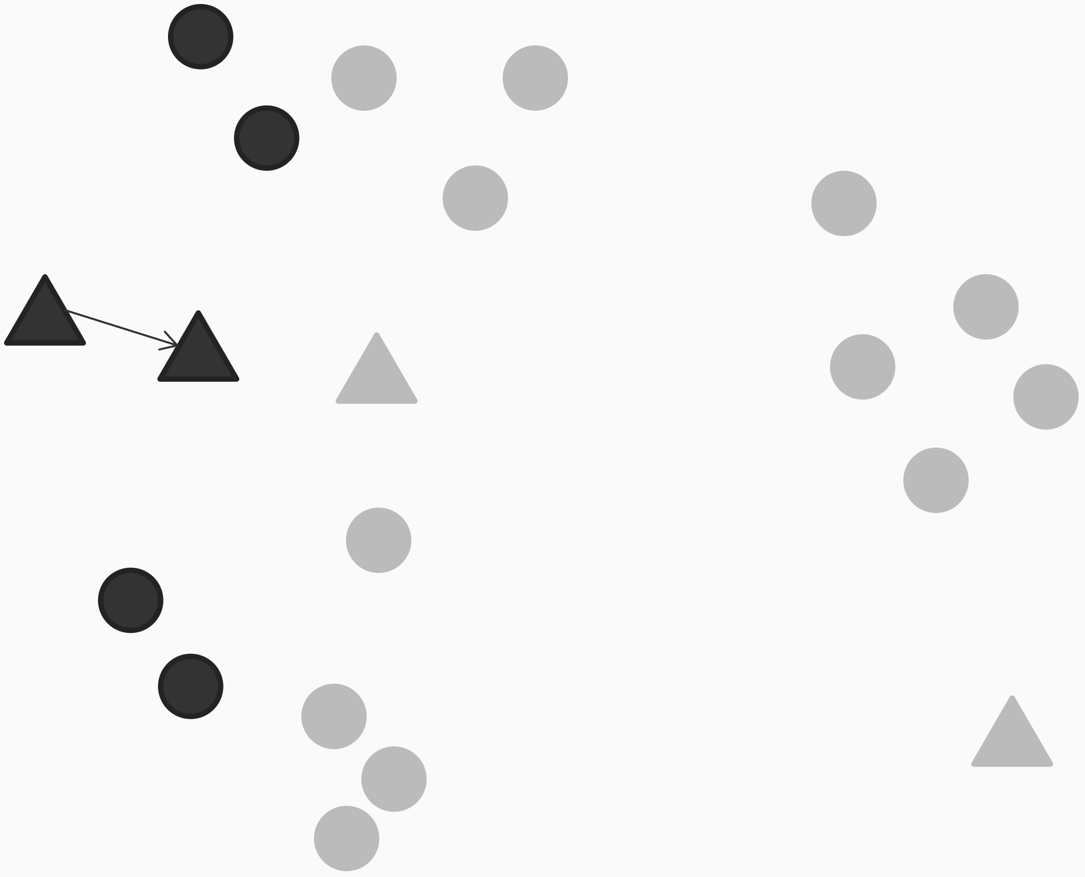
Iteration 1: Assign points to means



K means example

Suppose K = 3

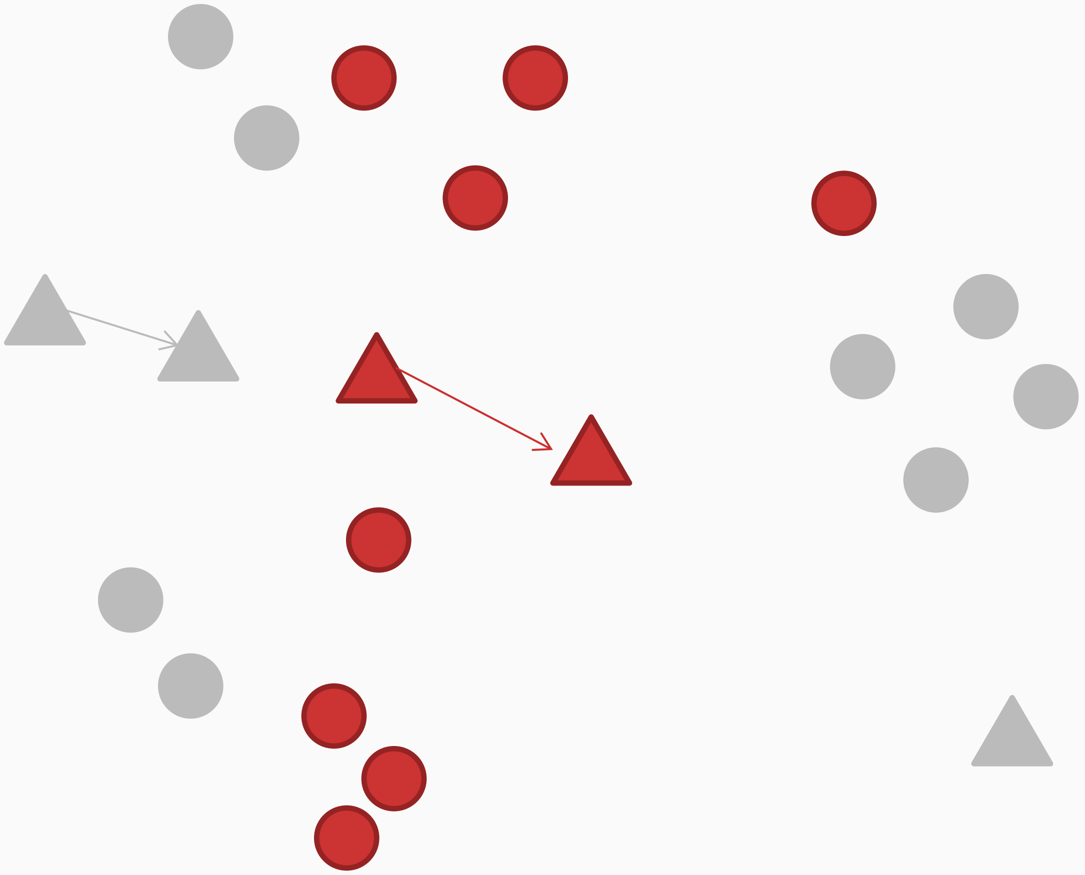
Iteration 1: Re-estimate the means



K means example

Suppose K = 3

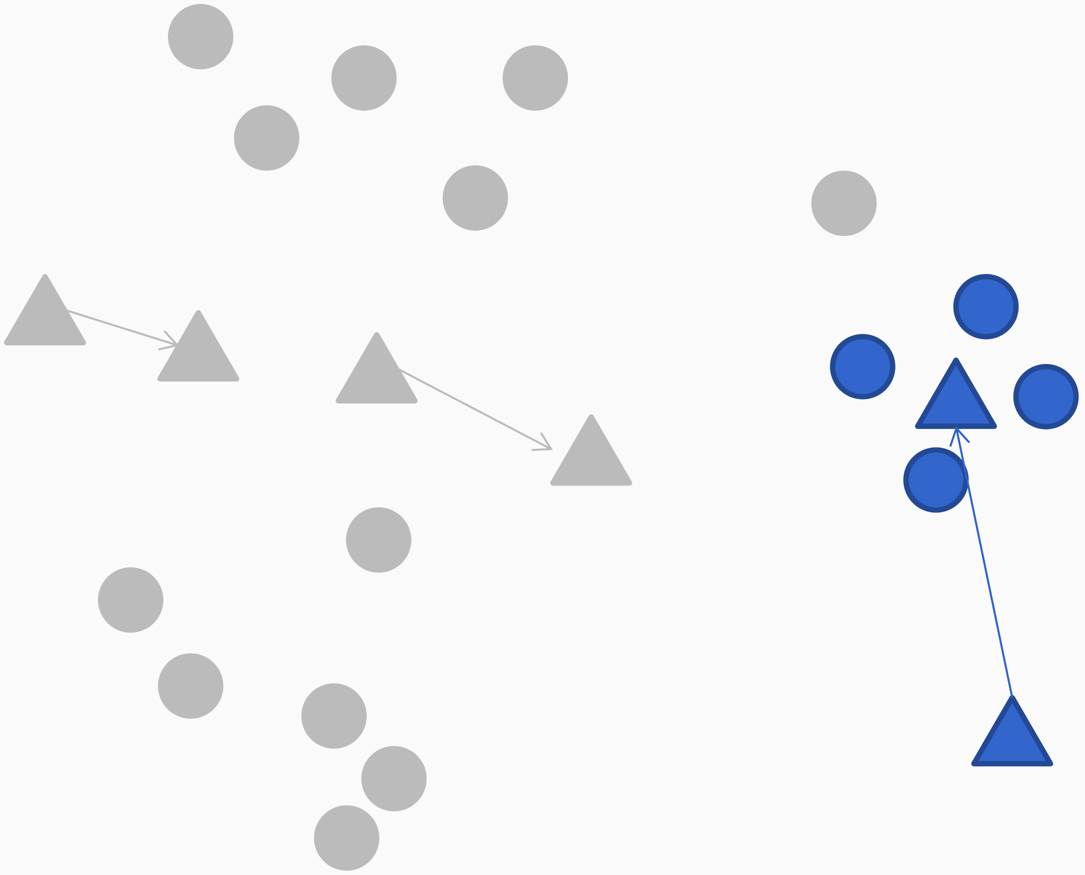
Iteration 1: Re-estimate the means



K means example

Suppose K = 3

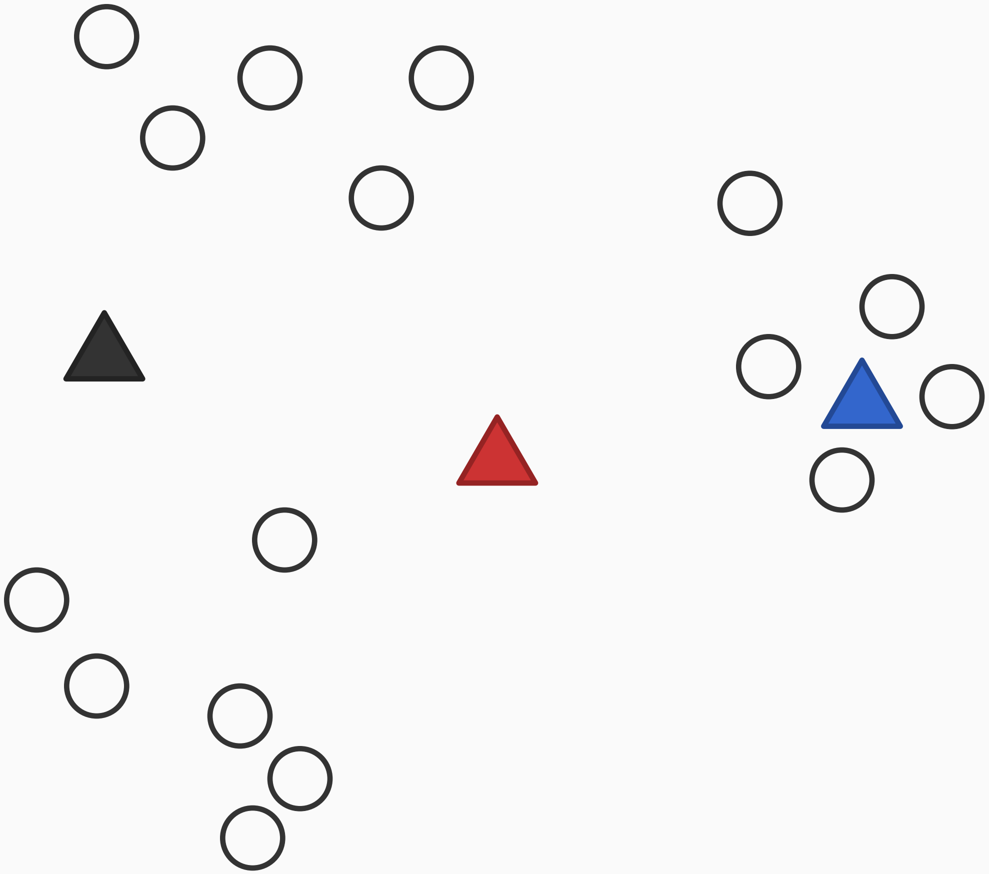
Iteration 1: Re-estimate the means



K means example

Suppose K = 3

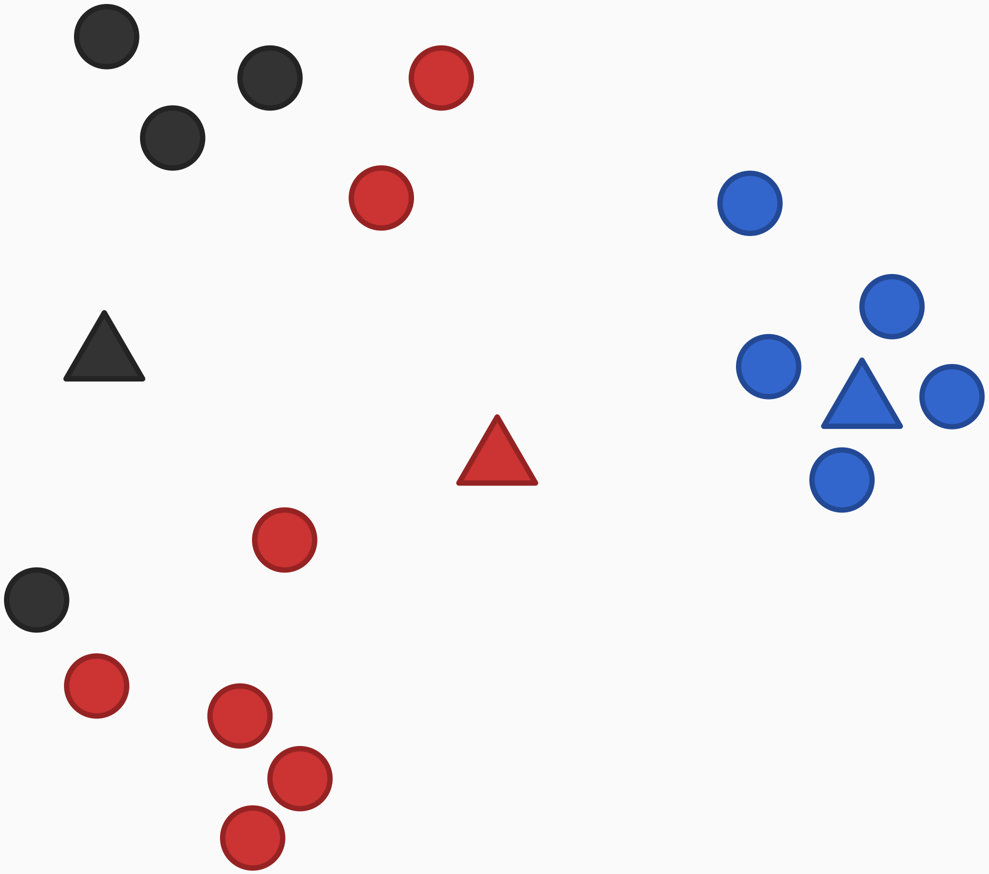
Iteration 2: Re-label points



K means example

Suppose K = 3

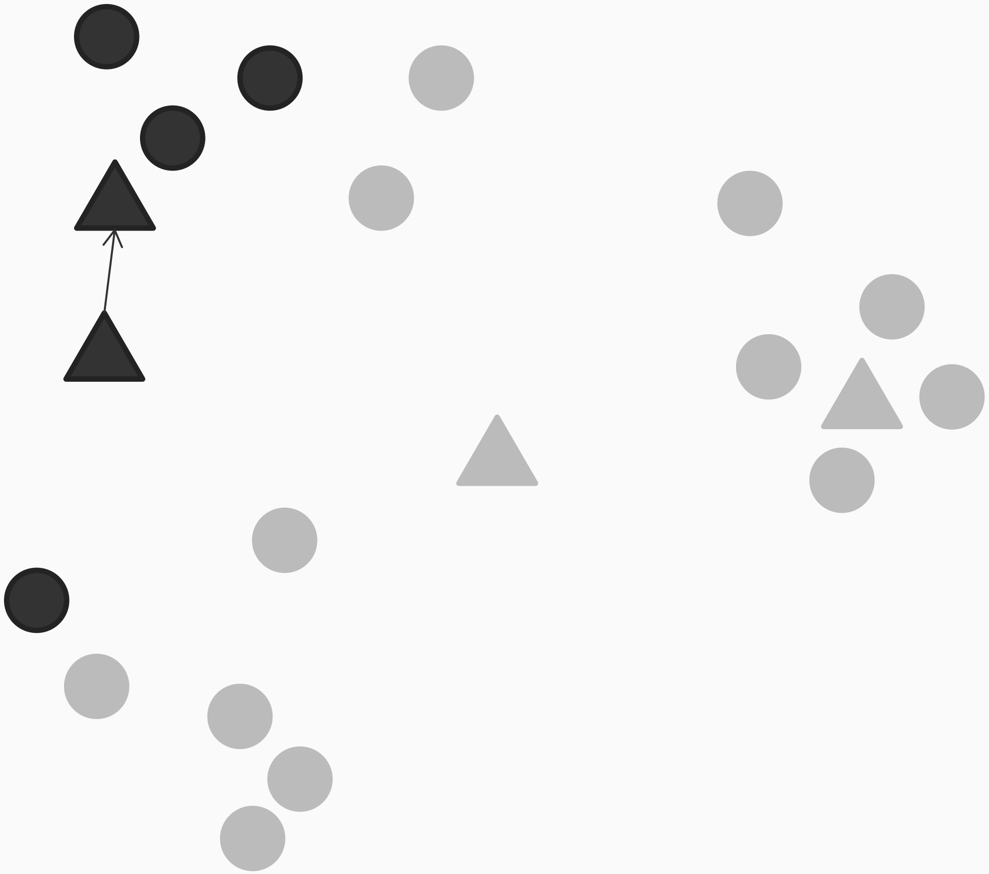
Iteration 2: Re-label points



K means example

Suppose K = 3

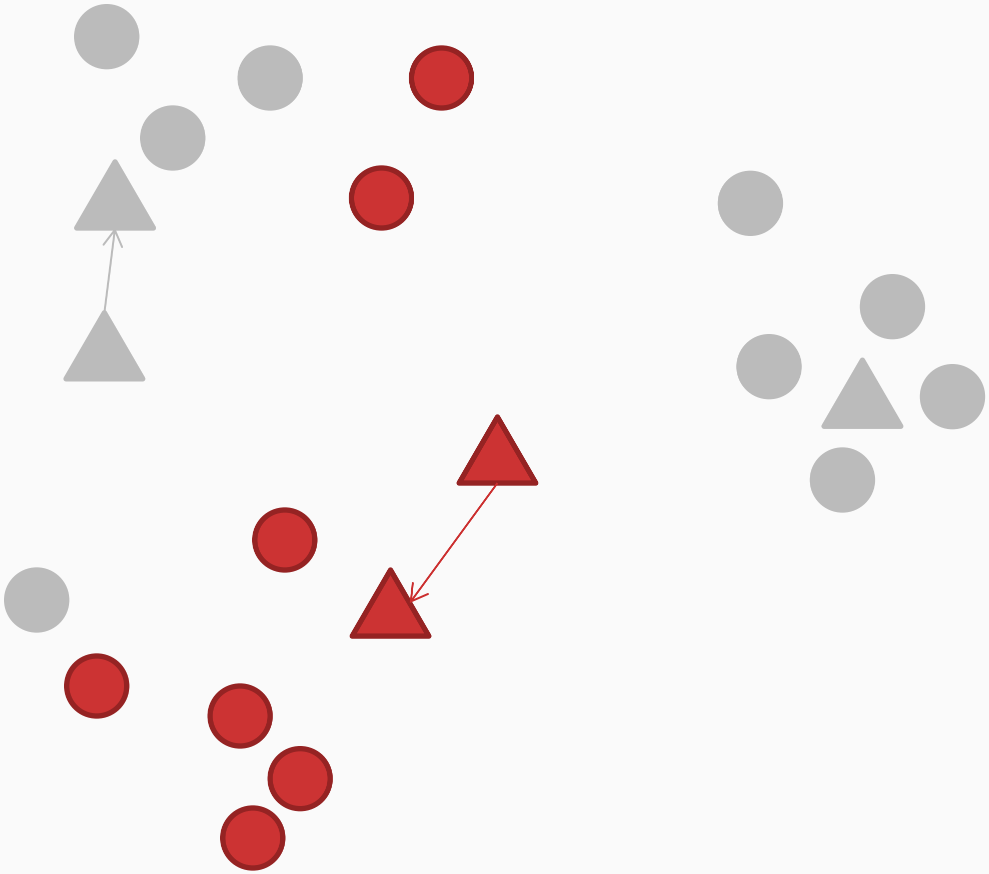
Iteration 2: Re-estimate the means



K means example

Suppose K = 3

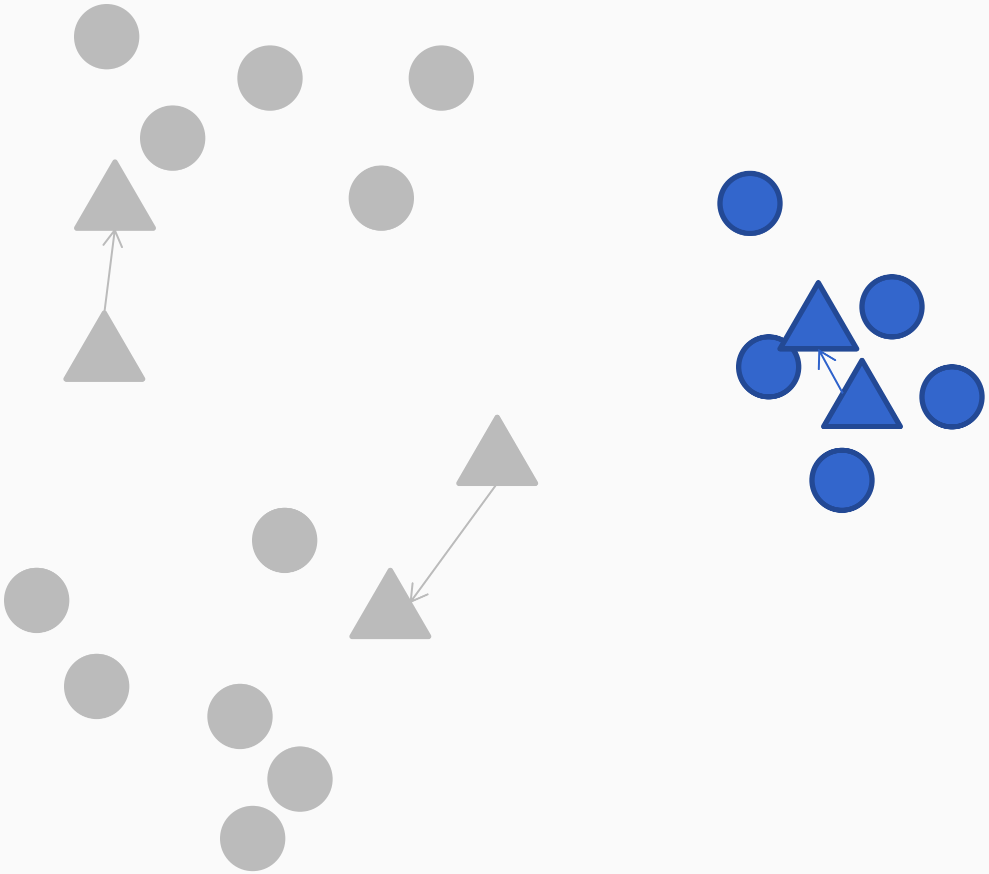
Iteration 2: Re-estimate the means



K means example

Suppose K = 3

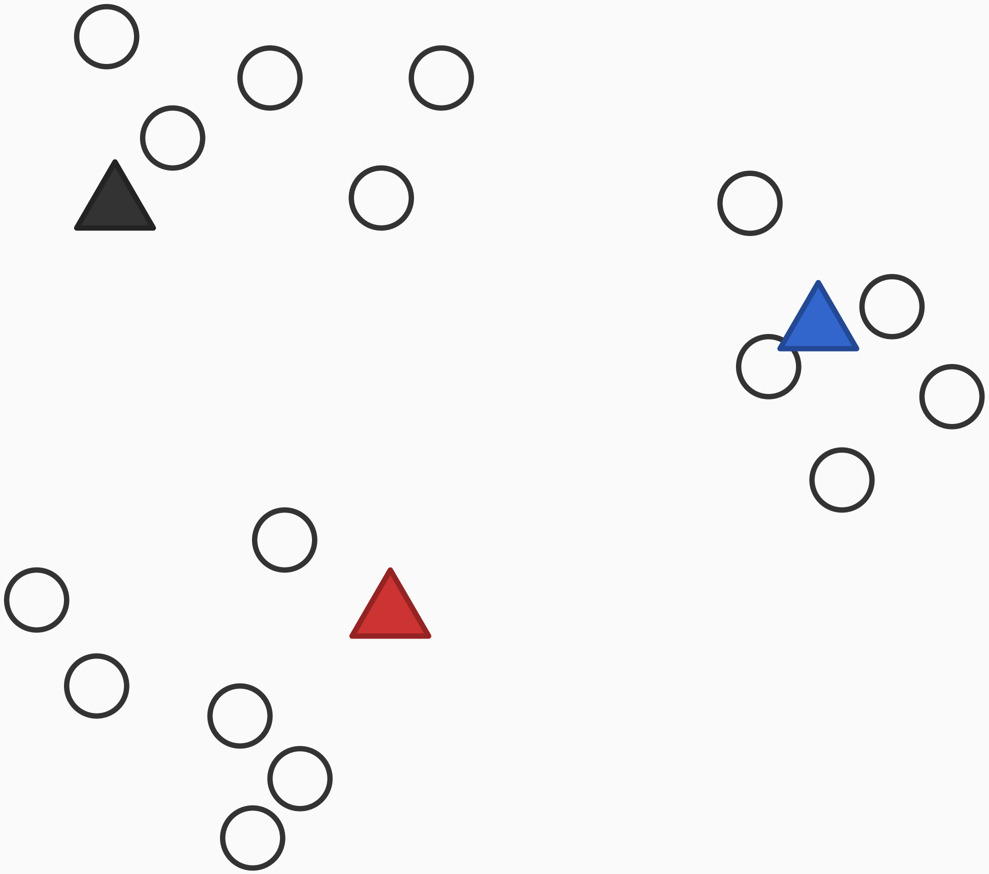
Iteration 2: Re-estimate the means



K means example

Suppose K = 3

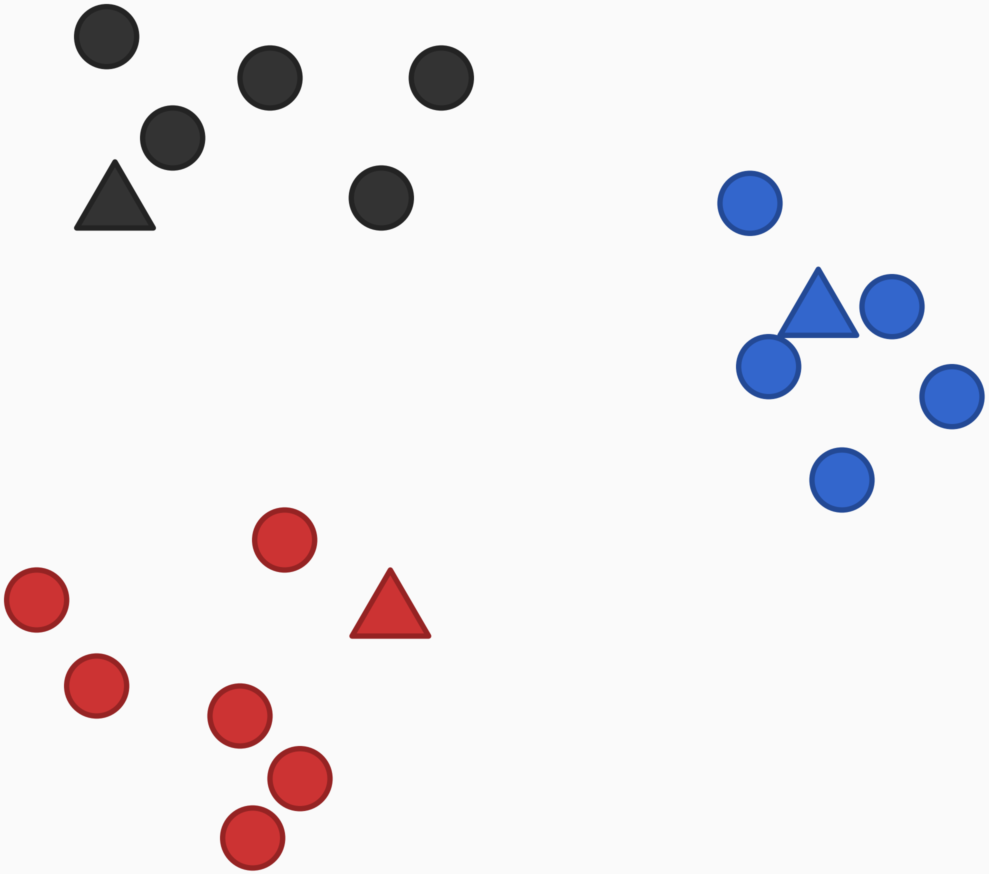
Iteration 3: Re-label the points



K means example

Suppose K = 3

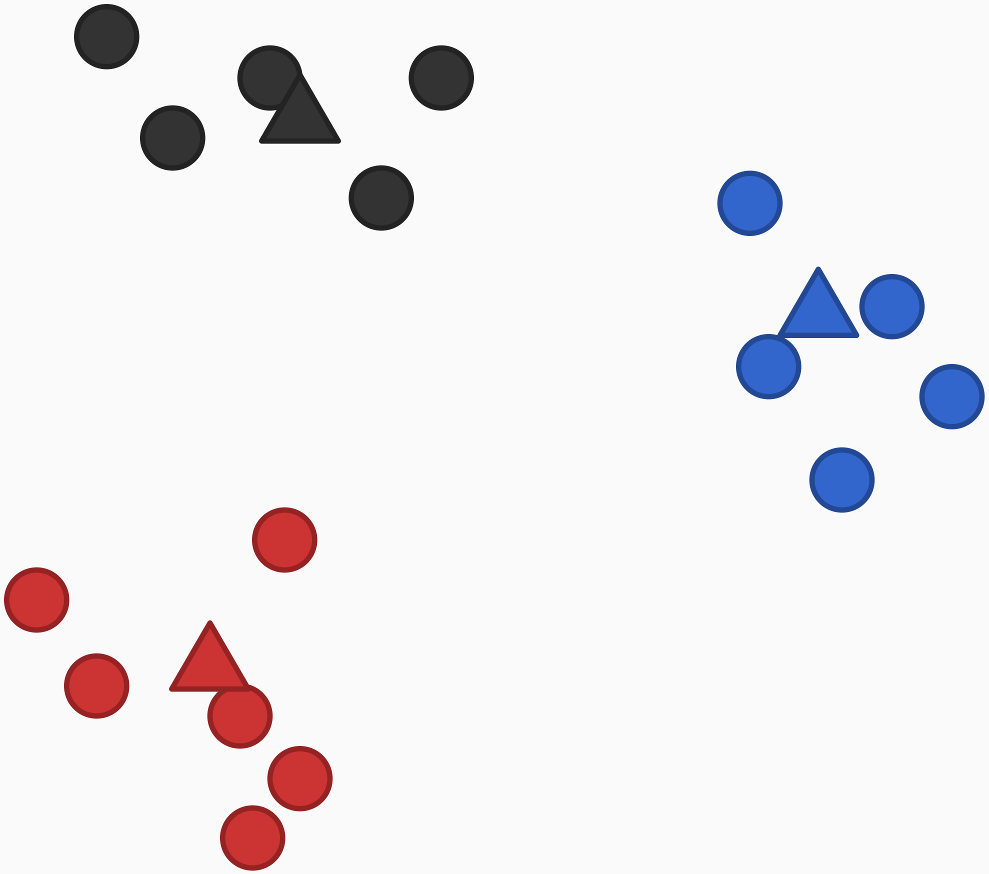
Iteration 3: Re-label the points



K means example

Suppose K = 3

Iteration 4: Re-estimate the means



Unsupervised learning

- Learning with missing labels/latent variables/hidden labels
  - Some examples could be labeled and some unlabeled – semi-supervised learning

- The EM algorithm
  - Assume a particular model for the joint distribution, and iteratively maximize expected log likelihood
  - A recipe for defining an algorithm

- Effectively this is clustering
  - Many, many, many clustering algorithms (a full semester’s worth)
  - We saw K-means, which is equivalent to Hard EM with the Gaussian mixture model