Maximum likelihood
Our approaches to modeling data thus far have been either probabilistic or non-probabilistic in motivation.

- **Probabilistic models**: Probability distributions defined on data, e.g.,
  1. Bayes classifiers
  2. Logistic regression
  3. Least squares and ridge regression (using proper interpretation)
  4. Bayesian linear regression

- **Non-probabilistic models**: No probability distributions involved, e.g.,
  1. Perceptron
  2. Support vector machine
  3. Decision trees
  4. K-means

In *every* case, we have some objective function we are trying to optimize (greedily vs non-greedily, locally vs globally).
Maximum likelihood

As we’ve seen, one probabilistic objective function is maximum likelihood.

**Setup:** In the most basic scenario, we start with

1. some set of model parameters $\theta$
2. a set of data $\{x_1, \ldots, x_n\}$
3. a probability distribution $p(x|\theta)$
4. an i.i.d. assumption, $x_i \overset{iid}{\sim} p(x|\theta)$

Maximum likelihood seeks to find the $\theta$ that maximizes the likelihood

$$
\theta_{\text{ML}} = \arg \max_{\theta} \ p(x_1, \ldots, x_n|\theta) \quad (a) \quad \Rightarrow \quad \arg \max_{\theta} \ \prod_{i=1}^{n} p(x_i|\theta) \quad (b) \quad \Rightarrow \quad \arg \max_{\theta} \ \sum_{i=1}^{n} \ln p(x_i|\theta)
$$

Equality (a) follows from i.i.d. assumption.
Equality (b) follows since $f(y) > f(x) \Rightarrow \ln f(y) > \ln f(x)$. 
We’ve discussed two maximum likelihood models: least squares linear regression (using the Gaussian interpretation) and the Bayes classifier.

Both of these models were “nice” because we could find their respective $\theta_{\text{ML}}$ directly by writing out an equation for it and plugging in data to solve.

**Gaussian with unknown mean and covariance**

In the first lecture, we saw if $x_i \sim \text{iid } N(\mu, \Sigma)$, where $\theta = \{\mu, \Sigma\}$, then

\[
\nabla_\theta \ln \prod_{i=1}^{n} p(x_i|\theta) = 0
\]

gives the maximum likelihood values for $\mu$ and $\Sigma$, and

\[
\mu_{\text{ML}} = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad \Sigma_{\text{ML}} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{\text{ML}})(x_i - \mu_{\text{ML}})^T
\]
COORDINATE ASCENT AND MAXIMUM LIKELIHOOD

In more complicated models, we might split the parameters into groups $\theta_1, \theta_2$ and try to maximize the likelihood over both of these,

$$\theta_{1,\text{ML}}, \theta_{2,\text{ML}} = \arg \max_{\theta_1, \theta_2} \sum_{i=1}^{n} \ln p(x_i|\theta_1, \theta_2),$$

Although we can solve one *given* the other, we can’t solve it *simultaneously*.

Coordinate ascent (probabilistic version)

We saw how K-means presented a similar situation, and that we could optimize using coordinate ascent. This technique is generalizable.

**Algorithm**: For iteration $t = 1, 2, \ldots$,

1. Optimize $\theta_1^{(t)} = \arg \max_{\theta_1} \sum_{i=1}^{n} \ln p(x_i|\theta_1, \theta_2^{(t-1)})$

2. Optimize $\theta_2^{(t)} = \arg \max_{\theta_2} \sum_{i=1}^{n} \ln p(x_i|\theta_1^{(t)}, \theta_2)$
There is a third (subtly) different situation, where we really want to find $\theta_{1,\text{ML}} = \arg \max_{\theta_1} \sum_{i=1}^{n} \ln p(x_i | \theta_1)$.

However, this function is “tricky” to optimize directly. However, we can add a second variable $\theta_2$ such that

$$\sum_{i=1}^{n} \ln p(x_i, \theta_2 | \theta_1)$$

is “easy” to work with. We’ll make this clearer later.

- Notice in this second case that $\theta_2$ is on the left side of the conditioning bar. This implies a prior on $\theta_2$, (whatever “$\theta_2$” turns out to be).
- We will next discuss a fundamental technique called the EM algorithm for finding $\theta_{1,\text{ML}}$ by using Function 2 instead.
EXPECTATION-MAXIMIZATION ALGORITHM
A Motivating Example

Let \( x_i \in \mathbb{R}^d \), be a vector with missing data. Split this vector into two parts:

1. \( x_i^o \) – observed portion (the sub-vector of \( x_i \) that is measured)
2. \( x_i^m \) – missing portion (the sub-vector of \( x_i \) that is still unknown)
3. The missing dimensions can be different for different \( x_i \).

If we assume that \( x_i \overset{iid}{\sim} N(\mu, \Sigma) \). Then we want to “ignore the missingness”

\[
\mu_{ML}, \Sigma_{ML} = \arg \max_{\mu, \Sigma} \sum_{i=1}^{n} \ln p(x_i^o | \mu, \Sigma).
\]

This is difficult to optimize. However, if we knew \( x_i^m \) (and therefore \( x_i \)), then

\[
\mu_{ML}, \Sigma_{ML} = \arg \max_{\mu, \Sigma} \sum_{i=1}^{n} \ln p(x_i^o, x_i^m | \mu, \Sigma)
\]

is easy to optimize (we just did it on a previous slide).
In a more general setup, we have two parameter sets $\theta_1, \theta_2$, where

$$p(x|\theta_1) = \int p(x, \theta_2|\theta_1) \, d\theta_2 \quad \text{(marginal distribution)}$$

For the previous example we can show that

$$p(x_i^o|\mu, \Sigma) = \int p(x_i^o, x_i^m|\mu, \Sigma) \, dx_i^m = N(\mu_i^o, \Sigma_i^o),$$

where $\mu_i^o$ and $\Sigma_i^o$ are the sub-vector/sub-matrix of $\mu$ and $\Sigma$ defined by $x_i^o$.

**Q:** Why isn’t $\mu_{ML}, \Sigma_{ML} = \arg \max_{\mu, \Sigma} \sum_{i=1}^n \ln p(x_i^o|\mu, \Sigma)$ just a sub-problem that can be analytically solved like before?

**A:** It would be if the missing dimensions are the same for every $x_i$. But we are assuming the missing part can change across observations.
The EM objective function

We need to define a general *objective function* that gives us what we want:

1. It lets us optimize the marginal $p(x|\theta_1)$ over $\theta_1$,
2. It uses $p(x, \theta_2|\theta_1)$ in doing so purely for computational convenience.

The EM objective function

Before picking it apart, we claim that this objective function is

$$
\ln p(x|\theta_1) = \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} \, d\theta_2 + \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} \, d\theta_2
$$

Some immediate comments:

- $q(\theta_2)$ is *any* probability distribution (assumed continuous for now)
- To calculate this, we assume we know $p(\theta_2|x, \theta_1)$. That is, given a setting of $\theta_1$ and data $x$, we can solve the posterior of $\theta_2$. 
Let's show that this equality is actually true

$$\ln p(x|\theta_1) = \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} \, d\theta_2 + \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} \, d\theta_2$$

$$= \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)q(\theta_2)}{p(\theta_2|x, \theta_1)q(\theta_2)} \, d\theta_2$$

Remember basic rules of probability:

$$p(a, b|c) = p(a|b, c)p(b|c) \Rightarrow p(b|c) = p(a, b|c)/p(a|b, c).$$

Letting $a = \theta_1$, $b = x$ and $c = \theta_1$, we conclude

$$\ln p(x|\theta_1) = \int q(\theta_2) \ln p(x|\theta_1) \, d\theta_2$$

$$= \ln p(x|\theta_1)$$
The EM objective function splits our desired objective into two terms:

\[
\ln p(x | \theta_1) = \int q(\theta_2) \ln \frac{p(x, \theta_2 | \theta_1)}{q(\theta_2)} d\theta_2 + \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2 | x, \theta_1)} d\theta_2
\]

- A function only of \( \theta_1 \) (often called \( Q \))
- Kullback-Leibler divergence

Some more observations about the right hand side:

1. The KL divergence is always \( \geq 0 \) and only \( = 0 \) when \( q = p \).

2. We are assuming that the integral in “\( Q \)” can be calculated, leaving a function only of \( \theta_1 \) (for a particular setting of distribution \( q \)).
Q: What does it mean to iteratively optimize $\ln p(x|\theta_1)$ w.r.t. $\theta_1$?

A: One way to think about it is that we want a method for generating:

1. A sequence of values for $\theta_1$ such that $\ln p(x|\theta_1^{(t)}) \geq \ln p(x|\theta_1^{(t-1)})$.
2. We want $\theta_1^{(t)}$ to converge to a local optimum of $\ln p(x|\theta_1)$.

It doesn’t matter how we generate the sequence $\theta_1^{(1)}, \theta_1^{(2)}, \theta_1^{(3)}, \ldots$

We will show how EM generates #1 and just mention that EM satisfies #2.
The EM objective function

\[
\ln p(x|\theta_1) = \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} \, d\theta_2 + \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} \, d\theta_2
\]

define this to be \( Q(x, \theta_1) \)

Kullback-Leibler divergence

Definition: The EM algorithm

Given the value \( \theta_1^{(t)} \), find the value \( \theta_1^{(t+1)} \) as follows:

**E-step:** Set \( q(\theta_2) = p(\theta_2|x, \theta_1^{(t)}) \) and calculate

\[
Q(x, \theta_1) = \int q(\theta_2) \ln p(x, \theta_2|\theta_1) \, d\theta_2 - \int q(\theta_2) \ln q(\theta_2) \, d\theta_2.
\]

can ignore this entropy term

**M-step:** Set \( \theta_1^{(t+1)} = \arg \max_{\theta_1} Q(x, \theta_1) \).
Once we’re comfortable with the moving parts, the proof that the sequence \( \theta_1^{(t)} \) monotonically improves \( \ln p(x|\theta_1) \) just requires analysis:

\[
\ln p(x|\theta_1^{(t)}) = Q(x, \theta_1^{(t)}) + KL\left(q(\theta_2) \parallel p(\theta_2|x_1, \theta_1^{(t)})\right)
\]

\[= 0 \text{ by setting } q = p\]

\[= Q(x, \theta_1^{(t)}) \quad \leftarrow \text{E-step}\]

\[\leq Q(x, \theta_1^{(t+1)}) \quad \leftarrow \text{M-step}\]

\[\leq Q(x, \theta_1^{(t+1)}) + KL\left(q(\theta_2) \parallel p(\theta_2|x_1, \theta_1^{(t+1)})\right)\]

\[> 0 \text{ because } q \neq p\]

\[= \ln p(x|\theta_1^{(t+1)})\]
ONE ITERATION OF EM

Start: Current setting of $\theta_1$ and $q(\theta_2)$

For reference:

$$\ln p(x|\theta_1) = Q + KL$$

$$Q = \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} \, d\theta_2$$

$$KL = \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} \, d\theta_2$$
**One Iteration of EM**

**E-step:** Set \( q(\theta_2) = p(\theta_2|x, \theta_1) \) and update \( Q \).

\[
\text{KL}(q||p) = 0
\]

For reference:

\[
\ln p(x|\theta_1) = Q + KL
\]

\[
Q = \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} d\theta_2
\]

\[
KL = \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} d\theta_2
\]
**One Iteration of EM**

**M-step:** Maximize $Q$ wrt $\theta_1$. Now $q \neq p$.

\[
\ln p(X | \theta_{1_{up}}) = Q + KL
\]

For reference:

\[
Q = \int q(\theta_2) \ln \frac{p(x, \theta_2 | \theta_1)}{q(\theta_2)} d\theta_2
\]

\[
KL = \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} d\theta_2
\]
EM for missing data
We have a data matrix with missing entries. We model the columns as

\[ x_i \overset{iid}{\sim} N(\mu, \Sigma). \]

Our goal could be to

1. Learn \( \mu \) and \( \Sigma \) using maximum likelihood
2. Fill in the missing values “intelligently” (i.e., using model assumptions)
3. Both

We can’t use the analytic ML equations because we don’t have all the data. However, we can learn \( \mu_{\text{ML}} \) and \( \Sigma_{\text{ML}} \) using the EM algorithm.
EM FOR SINGLE GAUSSIAN MODEL WITH MISSING DATA

The original, generic EM objective is

\[
\ln p(x|\theta_1) = \int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} d\theta_2 + \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} d\theta_2
\]

The EM objective for this specific problem and notation is

\[
\sum_{i=1}^{n} \ln p(x_i^o|\mu, \Sigma) = \sum_{i=1}^{n} \int q(x_i^m) \ln \frac{p(x_i^o, x_i^m|\mu, \Sigma)}{q(x_i^m)} dx_i^m + \\
\sum_{i=1}^{n} \int q(x_i^m) \ln \frac{q(x_i^m)}{p(x_i^m|x_i^o, \mu, \Sigma)} dx_i^m
\]

We can calculate everything required to do this.
Prior to the E-step

Set \( q(x^m_i) = p(x^m_i | x^o_i, \mu, \Sigma) \) using current \( \mu, \Sigma \)

Let \( x^o_i \) and \( x^m_i \) represent the observed and missing dimensions of \( x_i \). For notational convenience, think

\[
x_i = \begin{bmatrix} x^o_i \\ x^m_i \end{bmatrix} \sim N \left( \begin{bmatrix} \mu^o_i \\ \mu^m_i \end{bmatrix}, \begin{bmatrix} \Sigma_{ii}^{oo} & \Sigma_{ii}^{om} \\ \Sigma_{ii}^{mo} & \Sigma_{ii}^{mm} \end{bmatrix} \right)
\]

Then we can show that \( p(x^m_i | x^o_i, \mu, \Sigma) = N(\hat{\mu}^m_i, \hat{\Sigma}_i^m) \), where

\[
\hat{\mu}^m_i = \mu^m_i + \Sigma_{ii}^{mo} (\Sigma_{ii}^{oo})^{-1} (x^o_i - \mu^o_i), \quad \hat{\Sigma}_i^m = \Sigma_{ii}^{mm} - \Sigma_{ii}^{mo} (\Sigma_{ii}^{oo})^{-1} \Sigma_{ii}^{om}.
\]

It doesn’t look nice, but these are just functions of sub-vectors of \( \mu \) and sub-matrices of \( \Sigma \) using the relevant dimensions defined by \( x_i \).
E-step: \( \mathbb{E}_{q(x_i^m)}[\ln p(x_i^o, x_i^m | \mu, \Sigma)] \)

For each \( i \) we will need to calculate the following term,

\[
\mathbb{E}_{q}[(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)] = \mathbb{E}_{q}[\text{trace}\{\Sigma^{-1} (x_i - \mu)(x_i - \mu)^T\}] \\
= \text{trace}\{\Sigma^{-1} \mathbb{E}_{q}[(x_i - \mu)(x_i - \mu)^T]\}
\]

The expectation is calculated using \( q(x_i^m) = p(x_i^m | x_i^o, \mu, \Sigma) \). So only the \( x_i^m \) portion of \( x_i \) will be integrated.

To this end, recall \( p(x_i^m | x_i^o, \mu, \Sigma) = N(\hat{\mu}_i^m, \hat{\Sigma}_i^m) \). We define

1. \( \bar{x}_i \): A vector where we replace the missing values in \( x_i \) with \( \hat{\mu}_i^m \).
2. \( \bar{\Sigma}_i \): A matrix of 0’s, plus sub-matrix \( \hat{\Sigma}_i^m \) in the missing dimensions.
**M-step**

**M-step:** Maximize $\sum_{i=1}^{n} \mathbb{E}_q[\ln p(x_i^o, x_i^m | \mu, \Sigma)]$

We’ll skip the mathematical details, but

$$\arg \max_{\mu, \Sigma} \sum_{i=1}^{n} \mathbb{E}_q[\ln p(x_i^o, x_i^m | \mu, \Sigma)]$$

can now be solved in closed form. Recalling the ¨ notation,

$$\mu_{up} = \frac{1}{n} \sum_{i=1}^{n} \ddot{x}_i,$$

$$\Sigma_{up} = \frac{1}{n} \sum_{i=1}^{n} \{(\ddot{x}_i - \mu_{up})(\ddot{x}_i - \mu_{up})^T + \dddot{\Sigma}_i\}$$

We then return to the E-step to calculate the new $p(x_i^m | x_i^o, \mu_{up}, \Sigma_{up})$. 

IMPLEMENTATION DETAILS

Need to initialize $\mu$ and $\Sigma$. For example, initialize by setting missing values to zero and calculate $\mu_{ML}$ and $\Sigma_{ML}$. (Also can just randomly initialize them.)

The EM objective function can be calculated after each update to $\mu$ and $\Sigma$ and will look like the figure above. Stop when the change is “small.”