COMS 4721: Machine Learning for Data Science
Lecture 15, 3/26/2015

Prof. John Paisley

Columbia University
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. assumption data is *independent and identically distributed*: $x_i \overset{iid}{\sim} p(x|\theta)$

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

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

Equality (a) follows from i.i.d. assumption. Equality (b) follows from the fact that the function $\ln(\cdot)$ preserves greater-than/less-than relationships.
Two maximum likelihood models we’ve discussed are least squares for linear regression and the Bayes classifier.

Both of these models were “nice” because we could calculate \( \theta_{\text{ML}} \) directly, i.e., we could write out the equation for \( \theta_{\text{ML}} \) as a simple function of our data.

**Gaussian with unknown mean and covariance**

As another example, if \( x_i \overset{iid}{\sim} N(\mu, \Sigma) \), where \( \theta = \{\mu, \Sigma\} \), then we saw

\[
\nabla_\theta \ln \prod_{i=1}^{n} p(x_i | \theta) = 0 \iff \mu_{\text{ML}} = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad \Sigma_{\text{ML}} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{\text{ML}})(x_i - \mu_{\text{ML}})^T
\]

Aside: We’ve used this as the class-conditional distribution of the Bayes classifier, where we restricted its calculation to the \( x_i \) in a particular class.
In more complicated models, we may have two sets of parameters $\theta_1, \theta_2$ that we wish to maximize the likelihood over,

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

where we can find the best $\theta_1$ given $\theta_2$ and vice versa, but not both together.

**Coordinate ascent (probabilistic version)**

We saw how K-means presented a similar situation, and that we could optimize using coordinate ascent. The same holds for maximum likelihood:

**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

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

but for some reason it is much easier to optimize

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

Notice in this case that:

1. \( \theta_2 \) is on the left side of the conditioning bar. This implies a prior distribution on \( \theta_2 \) and so the maximum is the MAP solution.

2. Optimizing Objective 2 does not optimize Objective 1.

We will discuss a method for working with something that looks like Objective 2, but that we can prove optimizes Objective 1.
EXPECTATION-MAXIMIZATION ALGORITHM
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 (a subset of $x_i$ that is measured)
2. $x_i^m$ – missing portion (a subset of $x_i$ that is still a random variable)
3. Missing dimensions can vary, and some $x_i$ can have nothing missing.

If we assume that $x_i \sim iid \sim N(\mu, \Sigma)$. 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

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

For the previous example we can show that

$$p(x^o_i|\mu, \Sigma) = \int p(x^o_i, x^m_i|\mu, \Sigma) \, dx^m_i = N(\mu^o_i, \Sigma^o_i),$$

where $\mu^o_i$ and $\Sigma^o_i$ are the sub-vector/sub-matrix of $\mu$ and $\Sigma$ defined by $x^o_i$.

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

A: Only if the missing dimensions are the same for every $x_i$. But again, we assume the missing part can vary 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 find the posterior of $\theta_2$ analytically.
Let’s quickly 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:

\[
p(a, b|c) = p(a|b, c)p(b|c) \iff p(b|c) = p(a, b|c)/p(a|b, c).
\]

Therefore, replacing \(a, b, c\) with the symbols above, 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 whether we generate these $\theta_1^{(1)}, \theta_1^{(2)}, \ldots$ directly using $\ln p(x|\theta_1)$ or using some other, easier function.

We will show how EM gives #1 and only state that EM satisfies #2.
The EM Algorithm

The EM objective function

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

A function only of \( \theta_1 \) (often called \( Q \))

Kullback-Leibler divergence

Definition: The EM algorithm

For a value \( \theta_1^{(t)} \) at iteration \( t \):

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

\[
Q(x, \theta_1) = \int q(\theta_2) \ln p(x, \theta_2|\theta_1) \, d\theta_2 + \text{Entropy of } q.
\]

can ignore this

**M-step:** Set \( \theta_1^{(t+1)} = \arg \max_{\theta_1} Q(x, \theta_1) \).
PROOF OF MONOTONIC IMPROVEMENT

Once we’re comfortable with all the moving parts, the proof that the sequence $\theta_1^{(t)}$ is monotonically improving $\ln p(x|\theta_1)$ just requires logic:

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

= 0 by setting $q = p$

= $Q(x, \theta_1^{(t)})$ ← E-step

\leq Q(x, \theta_1^{(t+1)}) ← M-step

\leq Q(x, \theta_1^{(t+1)}) + KL(q(\theta_2)||p(\theta_2|x_1, \theta_1^{(t+1)})) > 0$ 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 $Q$.

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 $q \neq p$.

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

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 values because we don’t have all the data. However, we can learn \( \mu_{ML} \) and \( \Sigma_{ML} \) using the EM algorithm.
EM for Single Gaussian Model with Missing Data

The original, generic EM objective is

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

The EM objective for this specific problem and notation is

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

We can calculate everything required to do this.
Set \( q(x_i^m) = p(x_i^m|x_i^o, \mu, \Sigma) \) using current \( \mu, \Sigma \)

Let \( x_i^o \) and \( x_i^m \) represent the observed and missing dimensions of \( x_i \). For notational convenience, think

\[
x_i = \begin{bmatrix} x_i^o \\ x_i^m \end{bmatrix} \sim N \left( \begin{bmatrix} \mu_i^o \\ \mu_i^m \end{bmatrix}, \begin{bmatrix} \Sigma_{oo} & \Sigma_{om} \\ \Sigma_{mo} & \Sigma_{mm} \end{bmatrix} \right)
\]

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

\[
\hat{\mu}_i^m = \mu_i^m + \Sigma_{im}^{mo}(\Sigma_{ii}^{oo})^{-1}(x_i^o - \mu_i^o), \quad \hat{\Sigma}_i^m = \Sigma_{iii}^{mm} - \Sigma_{im}^{mo}(\Sigma_{ii}^{oo})^{-1}\Sigma_{im}^{om}.
\]

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**

**E-step:** \[ E_q(x_i^m) [\ln p(x_i^o, x_i^m | \mu, \Sigma)] \]

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

\[
E_q[(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)] = E_q[\text{trace}\{\Sigma^{-1} (x_i - \mu)(x_i - \mu)^T\}] = \text{trace}\{\Sigma^{-1} E_q[(x_i - \mu)(x_i - \mu)^T]\}
\]

The expectation is calculated using \( q(x_i^m) = p(x_i^m | x_i^o, \mu, \Sigma) \).

We need to find \( E_q[(x_i - \mu)(x_i - \mu)^T] \). We can show that this is equivalent to

1. Replacing the missing values in \( x_i \) with \( \hat{\mu}_i^m \), call the result \( \hat{x}_i \).
2. Adding a matrix \( \hat{\Sigma}_i \) equal to \( \hat{\Sigma}_i^m \) in the correct sub-matrix, 0 elsewhere.
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

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

is now straightforward and in closed form. Recalling the “§” notation,

\[
\mu_{\text{update}} = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_i,
\]

\[
\Sigma_{\text{update}} = \frac{1}{n} \sum_{i=1}^{n} \{(\bar{x}_i - \mu_{\text{update}})(\bar{x}_i - \mu_{\text{update}})^T + \Sigma_i\}\]
IMPLEMENTATION DETAILS

Requires an initialization — e.g., to initialize $\mu$ and $\Sigma$, set the missing values in $x_i$ to 0 and calculate $\mu_{ML}$ and $\Sigma_{ML}$ as if those were the true values.

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.”