ColumbiaX: Machine Learning

Lecture 15

Prof. John Paisley

Department of Electrical Engineering
& Data Science Institute

Columbia University

General inference technique - Expectation maximization algorithm.
↳ used for Gaussian mixture modelling (another clustering algo.)
↳ very general technique for doing maximum likelihood/maximum posterior inference.
Today, we'll focus on maximum likelihood.
↳ *

And also,
1. for learning missing data,
2.for learning posterior distributions,
3. proximate conditional, posterior distributions of model variables of interest
And today, we're gonna talk about it mostly in terms of learning.
Values of missing data, and also doing maximal likelihood,
in the case where we have missing data.

MAXIMUM LIKELIHOOD

APPROACHES TO DATA MODELING

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.,

  some part of data or entire dataset

  So we would use probability distributions to define what's a good setting for a model parameter and what's a bad setting for a model parameter.

  1. Bayes classifiers
  2. Logistic regression
  3. Least squares and ridge regression (using ML and MAP interpretation)
  4. Bayesian linear regression
- Non-probabilistic models: No probability distributions involved, e.g.,
  1. Perceptron (no probability interpretation at all.)
  2. Support vector machine
  3. Decision trees
  4. K-means

  [No discussion about probability involved.]

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

(non-convex) (convex)

in order to learn the model parameters, according to how well we can model the data.

So, in every case, though we wanna optimize something, there are now different choices that can be made to optimize them, and to say what type of optimum we're finding.

MAXIMUM LIKELIHOOD

[We're trying to optimize a probabilistic objective function called MLE.]

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 θ → define a prob. distribution on data, we want to learn them.
2. a set of data $\{x_1, \ldots, x_n\}$
3. a probability distribution $p(x|\theta)$ → prob. of data x given model parameters Θ
4. an i.i.d. assumption, $x_i \overset{iid}{\sim} p(x|\theta)$ each observation is generated iid from distribution. Condition on Θ, all data are independent and they have the same distribution.

Maximum likelihood seeks to find the θ that maximizes the likelihood *

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

(a) follows from i.i.d. assumption.
(b) follows since $f(y) > f(x) \;\Rightarrow\; \ln f(y) > \ln f(x)$.

Once we have a probability distribution on our data set,
depending on what we set theta to be,
that will change the probability of the data set that we observe.

MAXIMUM LIKELIHOOD

We've discussed maximum likelihood for a few models, e.g., least squares linear regression and the Bayes classifier.

Both of these models were "nice" because we could find their respective θ_{ML} analytically by writing an equation and plugging in data to solve.
(closed form)
& evaluate to solve for the parameter.

Gaussian with unknown mean and covariance

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

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

gives the following maximum likelihood values for μ and Σ:

empirical covariance of our data

$$\mu_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \Sigma_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{\text{ML}})(x_i - \mu_{\text{ML}})^T$$

saw this in k-means where we call it coordinate descent because we wanted to minimize that objective function.

In more complicated models, we might split the parameters into 2 groups θ_1, θ_2 and try to maximize the likelihood over both of these,

With prob. functions, we're always trying to max. something, we want to find the most probable things.

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

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

(Because we can't take the derivative of both Θ_1 & Θ_2 and optimize them together. However, given 1 value we can optimize the other. This will lead to a probabilistic version of coordinate ascent.

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.

?

For a particular iteration t, take Θ_2 and fix it's value at $t-1$.

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

* Given this update of $\Theta_{(1)}$, we plug that value in for $\Theta_{(1)}$. and now we maximize this thing over Θ_2, holding Θ_1 fixed.

And so we iterate back and forth between updating theta one, holding theta two fixed, and then updating theta two while holding theta one fixed. And eventually, we converge to either a global optimal, or more likely, a local optimal solution if this is non-convexed.

There is a third (subtly) different situation, where we really want to find

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

[θ_1 is the only parameter that we have in this function.]

[ex: no stable gradients]

Except this function is "tricky" to optimize directly. However, we figure out that we can add a second variable θ_2 such that

$$\sum_{i=1}^{n} \ln p(x_i, \theta_2|\theta_1) \qquad \text{(Function 2)}$$

is easier to work with. We'll make this clearer later.

isn't the prior on θ_1?

- Notice in this second case that θ_2 is on the *left* side of the conditioning bar. This implies a prior on θ_2, (whatever "θ_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

2 steps within each iteration of co-ordinate ascent algorithm:

1. expectation step.
2. maximization step.

in particular a certain type of missing data with a certain model assumption.

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) (only has measured values.)
2. x_i^m – missing portion (the sub-vector of x_i that is still unknown) where we don't know any of the values in it.
3. The missing dimensions can be different for different x_i. the dimensionality of this can change for each i.

(multivariate Gaussian)

** We assume that $x_i \overset{iid}{\sim} N(\mu, \Sigma)$, and want to solve

And after it is generated, some of values for whatever reason go missing.

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

only over what we observed portion of vector.

This is tricky. However, if we knew x_i^m (and therefore x_i), then

hidden variable ← θ_2 θ_1

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

but that assumes we know the values in the missing portions of the vector.

is very easy to optimize (we just did it on a previous slide).

* But for each vector, there might be some values that are missing, and these values don't form any pattern.
Some data can have many values missing, some data can have no values missing.
And the dimensions along which they're missing is totally random.
So there's no pattern in the way that the data is missing.

** We have to make a model assumption in order to use the EM Algorithm, otherwise it's not going to apply.

These xi os can have different dimensionality.
And those dimensions can correspond to different subsets of the dimensions in u and sigma, so it's not something that we can simply directly take a derivative of and set to 0.

We will discuss a method for optimizing $\sum_{i=1}^{n} \ln p(x_i^o|\mu, \Sigma)$ and imputing its missing values $\{x_1^m, \ldots, x_n^m\}$. This is a very general technique.

EM algorithm is going to give us a technique for actually optimizing this thing and also filling in all of these missing values.

General setup

(Discuss 1st in a more general setup.)

Imagine we have two parameter sets θ_1, θ_2, where

θ_2 in this model relate to θ_1 in the following way:

$$\underbrace{p(x|\theta_1)}_{\text{marginal distribution of } x \text{ given } \theta_1} = \int \overbrace{p(x, \theta_2|\theta_1) \, d\theta_2}^{\text{integral of joint distribution of } x \,\&\, \theta_2 \text{ given } \theta_1 \text{ over } \theta_2} \qquad \text{(marginal distribution)}$$

* missing data problem

Example: For the previous example we can show that

likelihood we want to maximize over the observed portion of the vector, where $\theta_1 = [\mu, \Sigma]$

$$\underline{p(x_i^o|\mu, \Sigma)} = \int \underset{**}{p(x_i^o, x_i^m|\mu, \Sigma) \, dx_i^m} = N(\mu_i^o, \Sigma_i^o),^{***}$$

where μ_i^o and Σ_i^o are the sub-vector/sub-matrix of μ and Σ defined by x_i^o.

This theta2 is like a hidden variable, or
a missing latent auxiliary variable, that we don't get to
observe in the data.

the integral of the joint distribution of the missing and observed portion.
So this is a multivariate Gaussian with mean u and
covariant sigma, but now we integrate out a missing portion of the vector.

equal to a multivariate Gaussian, where the mean is equal to the portion
of the mean vector restricted to the observed dimensions.
And the covariance is equal to the submatrix
formed by only considering the observed dimensions.

very useful derivation of working with multivariate Gaussians, where if we
wanna integrate out a subset of the dimensions of a multivariate Gaussian.
We simply get a Gaussian back with mean and
covariance equal to their appropriate subsets.

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 θ_1, [where Θ_2 is nowhere to be seen.]
2. It uses $p(x, \theta_2|\theta_1)$ in doing so purely for computational convenience.

this joint distribution (additional variable Θ_2) to help us maximize this thing for computational reasons.

The EM objective function

To achieve these 2 objectives using the EM algo.

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

what we want to maximize over Θ_1

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

joint likelihood over an additional variable Θ_2

** conditional posterior distribution of Θ_2 given x & Θ_1

* Some immediate comments:

- $q(\theta_2)$ is *any* probability distribution (assumed continuous for now)
- We assume we know $p(\theta_2|x, \theta_1)$. That is, given the data x and fixed values for θ_1, we can solve the conditional posterior distribution of θ_2.

*① What the EM algo. does instead of working with this L.H.S ; it's going to work with RHS. And it's going to work with this R.H.S, such that we end up with a value for θ_1 that gives us a local maximum of this L.H.S.

- So we're going to actually optimize this L.H.S without ever using (working with) it.

② q distribution on θ_2 → a probability distribution. And in principle, it can be any prob. distribution that we want on θ_2, as long as it is defined on the values that θ_2 can take.

However, the EM algorithm is going to tell us how to set this q distribution.

** This conditional posterior we're going to assume is something we can calculate in closed form very easily.

DERIVING THE EM OBJECTIVE FUNCTION [1st let's show the equality]

Let's show that this equality is actually true

same as previous slide

$$\underbrace{\ln p(x|\theta_1)}_{1} = \overbrace{\int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)}\, d\theta_2 + \int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)}\, d\theta_2}^{2}$$

logsum = log product of what's inside the logs.

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

Remember some rules of probability:

notice no a here.

$$\underbrace{p(a, b|c)}_{\text{joint}} = \underbrace{p(a|b, c)}_{\text{conditional}}\underbrace{p(b|c)}_{\text{conditional}} \quad \Rightarrow \quad p(b|c) = \frac{p(a, b|c)}{p(a|b, c)}.$$

[Divide both sides by]

Letting $a = \theta_2$, $b = x$ and $c = \theta_1$, we conclude

simplifies to

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

$$= \ln p(x|\theta_1)$$

*

So, we’ve shown that this term is equal to this term.
So, now the question is,
why in the world would it be easier to work with this much more complicated right hand side than this much simpler looking left hand side?
And so that's the genius of the Algorithm.

The EM objective function splits our desired objective into two terms:

$$\ln p(x|\theta_1) = \underbrace{\int \overbrace{q(\theta_2) \ln \frac{p(x,\theta_2|\theta_1)}{q(\theta_2)}}^{*} d\theta_2}_{\text{A function only of } \theta_1\text{, we'll call it } \mathcal{L}} + \underbrace{\int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x,\theta_1)} d\theta_2}_{\text{Kullback-Leibler divergence}}$$

can be thought of a distance measure between 2 probability distributions. It's not a probability distributions. It's not a prober distance, because it doesnot satisfy the triangle inequality.

Some more observations about the right hand side:

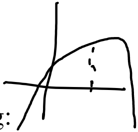
1. The **KL diverence** is always ≥ 0 and only $= 0$ when $q = p$. However it can be thought of a distance measure for 2 key reasons:
2. We are assuming that the integral in $\mathcal{L}$ can be calculated, leaving a function only of θ_1 (for a particular setting of the distribution q).

And then more intuitively speaking, we can think of when the distribution q overlaps more with the distribution p, the kl divergence gets smaller.
And then as our two distributions get farther and farther apart,
the kl divergence gets bigger.
But for our purposes,
the only two important properties are that this is always positive or equal to 0.
And it only equals 0 when this distribution and
this distribution are equal to each other.

* Once we define $q(\theta_2)$, this term 1 is actually only a $f(\theta_1)$. Because we have integrated out θ_2. (Also a function of data but we don't change the data we only change θ_1). So this thing actually even though it looks like it has θ_2 in it, once we solve it. → θ_2 will be integrated out, and we only have a function here of θ_1. So that's crucial observation 1.

BIGGER PICTURE

Okay, so we're going to see how we can use those two terms to optimize the log likelihood of x given theta1.
But before we do that, I first wanna ask,



Q: What does it mean to iteratively optimize $\ln p(x|\theta_1)$ w.r.t. θ_1?

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

1. A sequence of values for θ_1 such that $\ln p(x|\theta_1^{(t)}) \geq \ln p(x|\theta_1^{(t-1)})$.
   previous iteration
2. We want $\theta_1^{(t)}$ to converge to a local maximum of $\ln p(x|\theta_1)$.

as a function of iteration.

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.

If we plug in the sequence of values, we're monotonically increasing the log likelihood

THE EM ALGORITHM (define 2 steps to follow ① E step ② M step, Then prove in the next slide by following that rule we're monotically increasing the objective function.)

The EM objective function

θ_2 is integrated out.

$$\ln p(x|\theta_1) = \underbrace{\int q(\theta_2) \ln \frac{p(x, \theta_2|\theta_1)}{q(\theta_2)} \, d\theta_2}_{\text{define this to be } \mathcal{L}(x,\theta_1)} + \underbrace{\int q(\theta_2) \ln \frac{q(\theta_2)}{p(\theta_2|x, \theta_1)} \, d\theta_2}_{\text{Kullback-Leibler divergence}} \quad *$$

Definition: The EM algorithm (update θ_1)

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

Ist substep.

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

Ist do → IInd

$$\mathcal{L}_t(x, \theta_1) = \int q_t(\theta_2) \ln p(x, \theta_2|\theta_1) \, d\theta_2 - \underbrace{\int q_t(\theta_2) \ln q_t(\theta_2) \, d\theta_2}_{\text{can ignore this term}} .$$

t subscript to indicate what q distribution that we use.

only part that is functions of θ_2

Constant as far as θ_1 is concerned. So we're going ignore this part

M-step: Set $\theta_1^{(t+1)} = \arg\max_{\theta_1} \ \mathcal{L}_t(x, \theta_1)$.

Maximization

* And even though these 2 values can change for different values of q for different distributions q. The sum of the 2 values is always equal to the same thing. If we fix θ_1, then the sum of these 2 values equals the same thing for all distributions

** subscript q → which distribution it corresponds to. Changing for each iteration t.

o So the q distribution that we define on θ_2 at iteration t is going to be equal to the conditional posterior of θ_2 given x & θ_1 at t. Working at abstract level, can't tell what these distributions are but can calculate them.

oo So now the E step is completed, we have this term, l sub t. (L_t)
It's a function of the data x and theta1.
And it was calculated using the q distribution updated in iteration t.

ooo And in this step, we now treat this function as a function where we're free to change θ_1. So that's the M step. And o/p of that is the value of θ_1 for iteration t.

PROOF OF MONOTONIC IMPROVEMENT (following the 2 steps mentioned in previous slide)

objective f^n evaluated at θ_1^t < obj. $f^{(n)}$ evaluated at θ_1^{t+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*:

$$
\begin{aligned}
\ln p(x|\theta_1^{(t)}) &= \mathcal{L}(x, \theta_1^{(t)}) + \underbrace{KL\left(q(\theta_2) \,\|\, p(\theta_2|x_{\cancel{1}}, \theta_1^{(t)})\right)}_{=\,0 \text{ by setting } q\,=\,p} \\
&= \mathcal{L}_t(x, \theta_1^{(t)}) \qquad \leftarrow \text{E-step} \\
&\leq \mathcal{L}_t(x, \theta_1^{(t+1)}) \qquad \leftarrow \text{M-step} \\
&\leq \mathcal{L}_t(x, \theta_1^{(t+1)}) + \underbrace{KL\left(q_t(\theta_2) \,\|\, p(\theta_2|x_1, \theta_1^{(t+1)})\right)}_{>\,0 \text{ because } q \neq p} \\
&= \mathcal{L}(x, \theta_1^{(t+1)}) + KL\left(q(\theta_2) \,\|\, p(\theta_2|x_{\cancel{1}}, \theta_1^{(t+1)})\right) \\
&= \ln p(x|\theta_1^{(t+1)})
\end{aligned}
$$

we want maximize over θ_1

No subscript because true for all iterations

(But E step specifically sets q to be equal to this conditional posterior.)

t to indicate it is calculated using the distribution q_t

1st step: $q_t(\theta_2) \leftarrow p(\theta_2 | x, \theta_1^t)$

subscript not going to change, because we use the same q distribution.

evaluated at θ^{t+1}

(assuming θ_1 between iteration t & t+1)

o This is the q distribution that we updated using the value of θ_1 at iteration t.

* What do we do with the M step?
We simply take this algorithm, we don't change the q distribution on theta2.
And so the subscript t here is not going to change.
Cuz we use the same q distribution.
However, we let this distribution now vary in theta1.
We don't enforce it to be evaluated at theta1 for iteration t anymore.
We let it be a function of theta1.
And now we maximize it over theta1

* * Because we literally maximized this thing over theta1 and set that to be equal to theta1 t plus 1.
So if the value of theta1 t doesn't maximize this thing, then this value's gonna find the setting for theta1 that does maximize it, and so it'll be greater than.
If theta at t plus 1 is equal to theta at t, then it's equal.
And so that's where the equality can come in.
Okay, so this is the M Step.

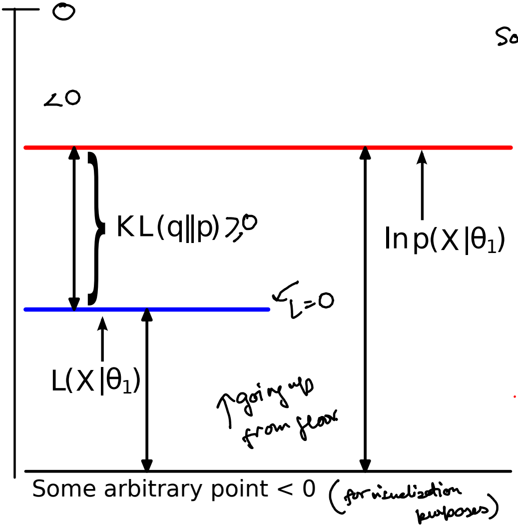
* * * Summary:

So the E step took this right hand side, put a particular qt in there to get rid of the K-L divergence, but we still have inequality.
The M step then let this variable theta1 be free to change.
And we change it to the value that maximized this term here.
So we now know that the update of this first term of the right hand side is greater than this term.

oo Finally, we recongize that the sum of these 2 terms is equal to the log of the likelihood of x, given the value of Θ_1 at iteration $t+1$

One iteration of EM

Start: Current setting of θ_1 and $q(\theta_2)$



So this is what we have already been working with.

For reference:

$$\ln p(x|\theta_1) = \mathcal{L} + KL \quad KL > 0.$$

* $\geq \mathcal{L}$ because

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

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

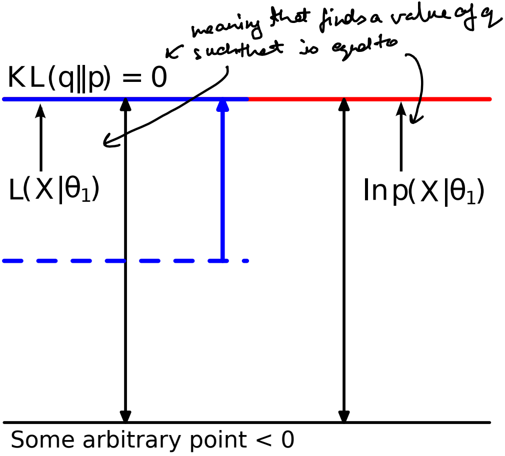
* This term has some particular value for a setting of θ,

Now, we will do EM.

ONE ITERATION OF EM

↱ is going to take the q distribution and set it to conditional posterior. and KL = 0.

E-step: Set $q(\theta_2) = p(\theta_2|x, \theta_1)$ and update $\mathcal{L}$.



For reference:

$$\ln p(x|\theta_1) = \mathcal{L} + KL$$
$$= \mathcal{L}$$

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

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

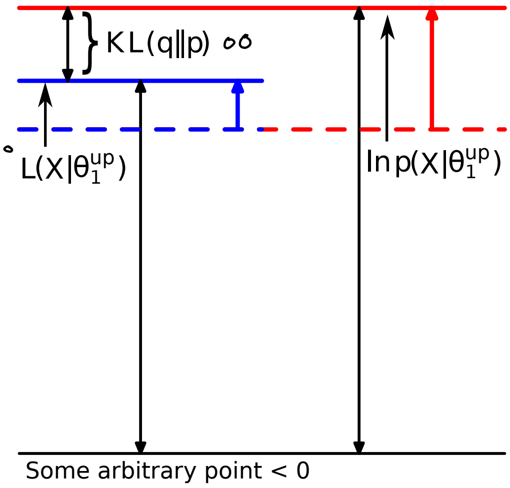
ONE ITERATION OF EM

Taken this value (old value of L) and found a new value for θ_1 such that this term increases

So by increasing this term it is pushing the Ceiling up but at the same time KL has also become non-negative. So we have even more slack here.

And so we have taken our objective function and found a new value that's pushed it upwards like this.

M-step: Maximize $\mathcal{L}$ wrt θ_1. Now $q \neq p$.



For reference:

$$\ln p(x|\theta_1) = \mathcal{L} + KL$$

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

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

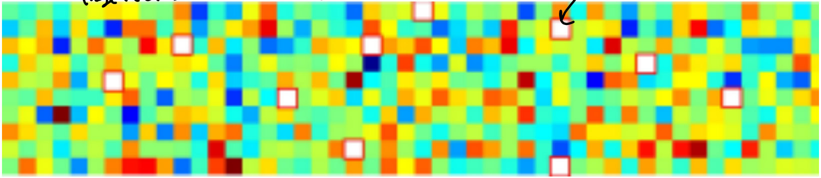
EM FOR MISSING DATA

THE PROBLEM

(Return to the problem discussed previously.)

each column is a multi-variate Gaussian with same mean and covariance. Except some of those vectors now have missing values.

don't have a value



First goal: find maximum likelihood values for μ and σ, given the data, where we take the missing points into consideration.

We have a data matrix with missing entries. We model the columns as

[vectors generated from multi-variate Gaussian with μ & Σ. But of those of have missing values]

Another goal: learn missing values given observed values.

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

Next: *

Our goal could be to

1. Learn μ and Σ using maximum likelihood
2. Fill in the missing values "intelligently" (e.g., using a model)
3. Both

We will see how to achieve both of these goals using the EM algorithm.

*
So next we're going to discuss an Algorithm that's going to do both of these things simultaneously.
It's going to allow us to maximize the log of the likelihood of the data, only takin
the measured points into consideration, over these two unknown parameters.
And then the q distributions that we're going to learn,
which we discussed previously, are going to be conditional posterior distributio
on these missing data points.
Which is going to tell us, or give us,
a probabilistic statement about what we believe these values to be.

EM FOR SINGLE GAUSSIAN MODEL WITH MISSING DATA

Up until now, we used this generic notation, where we assume we had some parameters θ_1 and some additional parameters θ_2 that we added.

The original, generic EM objective is

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

Because we have n different observations, we sum over RHS over our individual observations.

how this thing translates to our missing data problem with the multivariate Gaussian likelihood.

The EM objective for this specific problem and notation is

sub-vector of x_i by just considering the observed portions of it

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

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

Log likelihood of all observed portions of vectors, given the mean and covariance.

We can calculate everything required to do this.

* Because we made an IID assumption, the log of the likelihood of all of the data is equal to the sum of the logs of the individual likelihoods.

$\mu, \Sigma \rightarrow \theta_1$

$\theta_2 \rightarrow$ missing portions of the vectors.

** So here our joint likelihood is over the observed portion of i^{th} data vector and the missing portion of the i^{th} data vector.

So $x_i^m \rightarrow \theta_2$

E-STEP (PART ONE)

that's 2 sub-steps within the E step.

*

Set $q(x_i^m) = p(x_i^m | x_i^o, \mu, \Sigma)$ using current μ, Σ

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

simplifying notational change. **

mean corresponding to observed dimensions

covariance restricted to observed dimensions

cov of observed with missing dimensions

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

mean corresponding to missing dimensions

covariance restricted to missing dimensions

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

***

missing portion of mean vector.

covariance submatrices of the covariance.

$$\widehat{\mu}_i = \mu_i^m + \Sigma_i^{mo} (\Sigma_i^{oo})^{-1} (x_i^o - \mu_i^o), \quad \widehat{\Sigma}_i = \Sigma_i^{mm} - \Sigma_i^{mo} (\Sigma_i^{oo})^{-1} \Sigma_i^{om}.$$

data portion

It doesn't look nice, but these are just functions of sub-vectors of μ and sub-matrices of Σ using the relevant dimensions defined by x_i. And we can calculate these.

*
E-step I* requires us to take our q distribution on x_i over the missing portion of the vector for each i.
And set it equal to the conditional posterior of the missing portion of the i^{th} vector, given the observed portion of i^{th} vector and given the mean and covariance.

** If we make it for 1 value, for 1 x_i then for all the other vectors it might not hold that we have the observed portion in the top half and the missing portion in the bottom.

*** Posterior of the missing portion of the vector, given the observed portion and given the mean μ and covariant σ is still a multi-variante Gaussian with $\hat{\mu}$ (mean) and $\hat{\sigma}$ (covariance).

- We'll have a current value for the vector μ and sigma in our algorithm. At the current iteration, we'll have a value for these things.
And can actually calculate these functions to get the conditional posterior distribution of μ and the covariance on the missing portion of the vector.

E-STEP (PART TWO)

In the 1st part, we calculated this conditional posterior distribution for each value of i. *

Now, we need to take E step.

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,

outer product of these 2 vectors.

we have to calculate this expectation

also write as

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

complete data vector. mean covariance

bring the expectation inside.

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

***

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 $q(x_i^m) = N(\widehat{\mu}_i, \widehat{\Sigma}_i)$. We define

1. $\widehat{x}_i$: A vector where we replace the missing values in x_i with $\widehat{\mu}_i$.
2. $\widehat{V}_i$: A matrix of 0's, plus sub-matrix $\widehat{\Sigma}_i$ in the missing dimensions.

So our solⁿ is going to involve this vector ($\hat{x}_i$) and this matrix ($\hat{V}_i$) for each observation, so for each value of i.

*So for each data point we have a separate conditional posterior distribution on the missing portion of the vector.
If there is no missing portion of the vector then we simply don't, that q distribution is removed from the model, it doesn't appear.

** The E-step is the expectation of the log of the joint likelihood of the observed and missing portions of the vector, given the mean and covariance, using the q distribution of the missing portion

*** We calculate this integral over the missing portion of x_i which is something that we can do in closed form.
This something we can do in closed form. To simplify it say what we're going to o/p from this term (trace{ ... })

→ For the i^{th} observation,

1. Let's let $\hat{x}_i$ be equal to a vector where we take the missing values of x_i and replace them with $\hat{\mu}_i$. Remember that $\hat{\mu}_i$ is the q distribution, the conditional posterior distribution on the missing portion of the vector x_i so we fill in the missing terms in x_i with the mean of the conditional posterior distribution on it.

2. V_i be a matrix of zeros. So if our original data is in $\mathbb{R}^d$, then $\hat{V}_i$ is going to be a $d \times d$ matrix. We 1st fill it in with all zeros and then for submatrix that corresponds to the missing portion of the vector x_i, we fill those values in with the covariance matrix that we got from the conditional posterior of the missing values.

M-STEP

We've calculated the conditional posterior distribution on all of the missing portions of the vectors that we have, taken the E-step to calculate this. We can take the derivative of it with respect to μ and sigma and solve.

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

We'll omit the derivation, but the expectation can now be solved and

what we find when we do this

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

can be found. Recalling the $\widehat{\ }$ notation,

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

[Update of mean vector is the avg. of these $\hat{x}$. So we sum up these $\hat{x}$s divide by the no. of data, that's the update for the mean.] *

$$\Sigma_{\text{up}} = \frac{1}{n}\sum_{i=1}^{n} \{(\widehat{x}_i - \mu_{\text{up}})(\widehat{x}_i - \mu_{\text{up}})^T + \underset{**}{\widehat{V}_i}\}$$

So we our our update for μ & Σ, and

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

(conditional posteriors on all of the missing portions of each vector.)

* $\hat{x}_i$ ① x_i on the measured dimensions

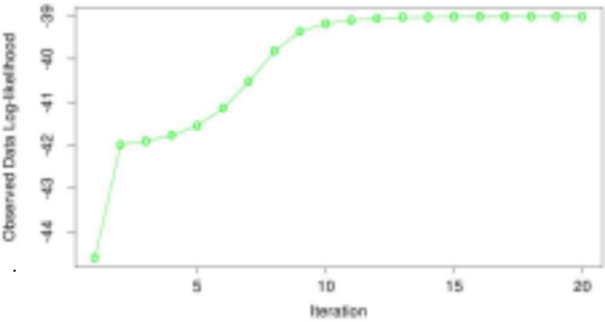
② mean of the conditional posterior on those dimensions. So the conditional posterior on the missing portion was a multivariate Gaussian. We take the mean and fill in the missing data with the mean.

And so this term looks identical to maximum likelihood when we have all of the data, where the only difference is that,
the missing data we've now filled in.
And then again, the covariance looks very much like the maximum likelihood solution,
when we have all the data.
The only difference is that we've first filled in all of the missing dimensions,
using the mean of the posterior distribution on xi,
the missing portion of xi.

** $\hat{V}_i$: We've added this additional vector matrix V for each observation i, where the submatrix of V_i is going to non-zero on the dimensions that correspond to the missing data, and it will equal zero on the dimensions that corresponds to the measured data. That simply reflects the fact that we certain about the measured dimensions, so there's no variance. But we're uncertain about the missing dimensions, and so there is some covariance for those dimensions.

explain

IMPLEMENTATION DETAILS



We need to initialize μ and Σ, for example, by setting missing values to zero and calculating μ_{ML} and Σ_{ML}. (We can also use random initialization.)

* The EM objective function is then calculated after each update to μ and Σ and will look like the figure above. Stop when the change is "small."

The output is μ_{ML}, Σ_{ML} and $q(x_i^m)$ for all missing entries.**

what we get → maximum likelihood solution last line on last slide

graph.

And when we do that, we iterate that process of going from e to m.
And when we go back and forth like that, we get a sequence of means and covariances, that look something like this when we evaluate the objective.

* So when we evaluate the log of the marginal likelihood we get a monotonically increasing function that eventually will converge. And we monitor this thing, and when the marginal improvement is very small we can terminate the algorithm, because it's converged.

**

So we have found a maximum likelihood solution for the original problem that we cared about, where we've integrated out all the missing data.

And

And we also get a conditional posterior distribution on all the missing dimensions.
And so this is going to allow us then to say something about the missing parts of the data that we have.
It says what the mean is of the missing portion so we can just fill in the data with mean if we want to. But we also get uncertainty, measure of uncertainty of the missing data in the form of the covariance of the Gaussian that corresponds to this q distribution.