E6892 Bayesian Models for Machine Learning

Columbia University, Fall 2015

Lecture 4, 10/1/2015

Instructor: John Paisley

- So far, we've been talking about posterior distributions where:

    1. The prior is conjugate to the likelihood. In this case we can calculate the posterior distribution exactly by giving the distribution name and its parameters.

    2. We can make a simple approximation using the Laplace's method. In this case, we approximate the posterior of the model variables with a multivariate Gaussian and use the MAP solution and the Hessian of the log joint likelihood evaluated at the MAP solution.

    3. We can calculate all conditional posterior distributions of the model variables and approximately sample i.i.d. from the posterior distribution using MCMC Gibbs sampling.

- In machine learning applications, the first approach is relatively rare since the model's are usually too complex. The second approach is also not very common since there are usually too many variables to estimate a full covariance matrix for them. The third approach *is* commonly used, but it suffers from poor scalability to large data sets, which is what machine learning problems typically encounter. A recent focus of research interest is on fixing this problem.

- There is a fourth major technique used widely in the machine learning community called *variational inference*. Unlike MCMC methods, which sample new values of the model variables in each iteration, variational methods set up an objective function (not unlike MAP) and optimize this objective function over parameters. As we will see in future lectures, these parameters correspond to parameters of distributions over the model variables. Variational methods try to find parameters to these distributions such that the result is a close approximation to the desired, but intractable posterior distribution.

- This will be made more precise in the next few lectures. However, to lay the groundwork for understanding variational inference (VI) it is necessary to go over the *Expectation-Maximization* (EM) algorithm in its full generality. The goal will be to present VI as a very simple but interesting modification of EM.

- Therefore, for the rest of this lecture we will return to the world of learning *point estimates* of model variables, instead of trying to learn their *posterior distributions*.

Maximum a posteriori (MAP) inference

- Recall the general modeling setup:

$$\begin{aligned} \text{Model:} \quad & X \sim p(X|\theta) \\ \text{Prior:} \quad & \theta \sim p(\theta) \end{aligned}$$

- We assume that we can't calculate the posterior $p(\theta|X)$ exactly, so instead we use MAP inference to find the value of θ that is a maximum of $p(\theta|X)$.

$$\theta_{\text{MAP}} = \arg\max_{\theta} \ \ln p(X, \theta) \tag{1}$$

(When $\ln p(X, \theta)$ is non-convex, we can usually only hope to find a local maximum. This is typically the case.)

- MAP inference is an optimization problem with a Bayesian interpretation of the objective function and what is being optimized.
  - Ultimately, we want to find the point θ that maximizes, or at least locally maximizes, the objective function $\ln p(X, \theta)$.
  - Equivalently, we want to search for a θ such that $\nabla_\theta \ln p(X, \theta) = 0$, since the derivative at any local maximum will equal zero. We also need to find this point in a way that assures us we haven't found a local *minimum*, since the gradient here will be zero as well.
  - Sometimes we can do this in closed form; we can take the derivative and solve for θ. In this case the problem is solved and no iterative optimization algorithm is necessary.
  - Often we can't find a solution for θ to the equation $\nabla_\theta \ln p(X, \theta) = 0$. Very much less often, sometimes we can't calculate $\nabla_\theta \ln p(X, \theta)$ to begin with. In this case, the typical approach is to use gradient methods, where we iteratively update θ according to the rule

$$\theta' \ \leftarrow \ \theta + \rho B \nabla \ln p(X, \theta). \tag{2}$$

    The value of $\rho > 0$ is a step size and the matrix B is positive semi-definite. When $B = I$, the identity matrix, this is steepest ascent. When B is the inverse of the negative Hessian of $\ln p(X, \theta)$, this is Newton's method.
  - What we should set ρ and B to are not solved problems. This is part of the "art" of optimization and there are several ways of figuring out what these should be, and they may change with each update of θ to take into account properties of the objective function $\ln p(X, \theta)$ at the current point.
  - In a sense, we want to have closed-form solutions. We want to say that, for such-and-such variable, the best setting for it is equal to something we can write out exactly, rather than something we stumble upon after being told which way to move by 50 different gradients (the number 50 is just for example).
  - The EM algorithm is a very clever way to do this. Before discussing EM in general, let's first look at one model for which we might give up hope if we wanted to do MAP by directly doing gradient ascent on the log joint likelihood as described above.

Probit regression

- Setup: The setup is the same as for logistic regression: We have a data set, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where $x \in \mathbb{R}^d$ is a feature vector and $y \in \{0, 1\}$ is its class label. We want to use this information to help predict the value of y for a new x.

- Model and prior: The probit model is similar to the logistic regression model. We assume that the labels are Bernoulli random variables that depend on the dot product of the corresponding features with a weight vector, which we assume has a Gaussian prior,

$$y_i \sim \text{Bernoulli}(\Phi(x_i^T w/\sigma)), \qquad w \sim \text{Normal}(0, \lambda^{-1} I) \tag{3}$$

  The value $\sigma > 0$ is a parameter we can change.

  As with the logistic regression model, the probit regression model is a discriminative classifier because it conditions on x throughout instead of modeling it with a distribution. However, where logistic regression uses the sigmoid function to map $x_i^T w$ to a number between 0 and 1, probit regression uses the cumulative distribution function (CDF) of a standard normal distribution,

$$\Phi(x_i^T w/\sigma) = \int_{-\infty}^{x_i^T w/\sigma} (2\pi)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}s^2} ds \tag{4}$$

  Notice that, as with the sigmoid function discussed in an earlier lecture, as $x_i^T w$ increases to $+\infty$, the function $\Phi(x_i^T w/\sigma)$ increases to 1. As $x_i^T w$ decreases to $-\infty$, the function $\Phi(x_i^T w/\sigma)$ decreases to 0. (In fact, it should be clear that any function that has the properties of a CDF is be a candidate for plugging into this Bernoulli distribution.)

- MAP inference: Ideally, we would do posterior inference for w. However, applying Bayes rule we quickly find that we run into the same problem as for logistic regression: The normalizing constant is not a tractable integral. A next step would be to learn the MAP solution for w,

$$\begin{aligned} w_{\text{MAP}} &= \arg\max_w \ \ln p(\vec{y}, w|X) \\ &= \arg\max_w \ \ln p(w) + \sum_{i=1}^N \ln p(y_i|w, x_i) \\ &= \arg\max_w \ -\frac{\lambda}{2} w^T w + \sum_{i=1}^N y_i \ln \Phi(x_i^T w/\sigma) + (1 - y_i) \ln(1 - \Phi(x_i^T w/\sigma)) \end{aligned} \tag{5}$$

  This requires us to calculate $\nabla_w \ln p(\vec{y}, w|X)$. One of the terms in this gradient is

$$\nabla_w \ln \Phi(x_i^T w/\sigma) = \nabla_w \ln \int_{-\infty}^{x_i^T w/\sigma} (2\pi)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}s^2} ds \tag{6}$$

  Taking the full gradient of $\ln p(\vec{y}, w|X)$, we can see that, not only would we not be able to solve for w, but it's not even clear what the analytic solution to $\nabla_w \ln \Phi(x_i^T w/\sigma)$ is! Therefore, we can't even get a closed form vector representation for $\ln p(\vec{y}, w|X)$ that tells us which direction to move in to update w.

- So the question is, is probit regression even harder to work with that logistic regression? Is it totally hopeless? We will see that the EM algorithm can be used to optimize $\ln p(\vec{y}, w|X)$ in a clever way that involves closed form updates and results in an algorithm not more complicated (and arguably cleaner and nicer) than gradient ascent for logistic regression.

Setting up the expectation-maximization (EM) algorithm

- In the following discussion of the EM algorithm, we return to the general model setting, where

$$X \sim p(X|\theta), \qquad \theta \sim p(\theta).$$

  Again, the goal is to do MAP inference, where we maximize $\ln p(X, \theta)$ over θ. We will then derive an EM algorithm for probit regression to make it more specific.

- EM starts with an assumption: Assume there is some other variable ϕ that we can *introduce* to the model so that the marginal distribution is unchanged. That is

$$p(X, \theta) = \int p(X, \theta, \phi) d\phi$$

  - Note that we can always manage to do this. The purpose for EM will be to pick a variable ϕ that is useful.
  - In the next lecture we will see that, when coming at it from the other direction, this comes directly out of the model itself. That is, in this lecture we are starting with $p(X, \theta)$ and finding that it's problematic, leading us down this EM path. Often we *start* with a more complex model definition $p(X, \theta, \phi)$ and then decide to optimize the marginal distribution of this $p(X, \theta)$. In this case we know ϕ because we defined it in the larger model.
  - Again, one case starts with $p(X, \theta)$ and expands it (the hard case), and one case starts with $p(X, \theta, \phi)$ and collapses it (the easy case). Traditionally an introduction of EM starts with the hard case.

- The next steps are all directed towards finding an equivalent representation for $\ln p(X, \theta)$ that involves $\ln p(X, \theta, \phi)$. The reason why this new representation should help us in any way to optimize $\ln p(X, \theta)$ more easily will come later. For now, we are only manipulating equations.

- The first step is to use a basic rule of probability

$$p(X, \theta) p(\phi|X, \theta) = p(X, \theta, \phi) \tag{7}$$

  and taking logarithms and reorganizing, we therefore have

$$\ln p(X, \theta) = \ln p(X, \theta, \phi) - \ln p(\phi|X, \theta) \tag{8}$$

  Notice that the LHS of this equation is the original objective function we want to maximize. We can now interpret this as the marginal log joint likelihood of the extended log joint likelihood on the RHS. Also subtracted on the RHS is the posterior distribution of ϕ. In practice we will find this with Bayes rule. From this we can guess (and be correct) that if $p(\phi|X, \theta)$ is not solvable, the EM algorithm we're heading towards isn't going to make optimizing $\ln p(X, \theta)$ easier.

- The next step toward the EM "master equation" is to introduce a probability distribution $q(\phi)$ such that q is defined on the same support as ϕ. That is, if $\phi \in \mathbb{R}$, then q is defined on $\mathbb{R}$. If $\phi \in \{1, 2, 3, 4, 5\}$ then q is a 5-dimensional probability vector on those integers. For now, this is the only requirement we make on $q(\phi)$. Later, EM will tell us what we should set $q(\phi)$ to.

- We then use $q(\phi)$ in the following sequence of equalities:

$$q(\phi) \ln p(X, \theta) = q(\phi) \ln p(X, \theta, \phi) - q(\phi) \ln p(\phi | X, \theta) \tag{9}$$

Comment: We simply multiply the LHS and RHS by the same thing. It doesn't matter that this thing is a function. For each value of ϕ the equality holds.

$$\int q(\phi) \ln p(X, \theta) d\phi = \int q(\phi) \ln p(X, \theta, \phi) d\phi - \int q(\phi) \ln p(\phi | X, \theta) d\phi \tag{10}$$

Comment: An integral can be thought of as an infinitesimal summation. For each infinitesimal bit we add on the LHS, we add an infinitesimal bit on the RHS that equals it.

$$\begin{aligned} \ln p(X, \theta) \quad = \quad & \int q(\phi) \ln p(X, \theta, \phi) d\phi - \int q(\phi) \ln q(\phi) d\phi \\ & - \int q(\phi) \ln p(\phi | X, \theta) d\phi + \int q(\phi) \ln q(\phi) d\phi \end{aligned} \tag{11}$$

Comment: On the LHS, we notice that $\ln p(X, \theta)$ doesn't involve ϕ in any way. Therefore it's a constant from the perspective of the integral and can be brought out front. The resulting integral is of a probability distribution. Even though we haven't defined $q(\phi)$ yet, we know it's integral has to equal on, hence the LHS. For the RHS we have added and subtracted the same thing, leaving the result unchanged. However, we do this because the result will give us insights as to how to design the EM algorithm and prove it works.

$$\ln p(X, \theta) = \int q(\phi) \ln \frac{p(X, \theta, \phi)}{q(\phi)} d\phi + \int q(\phi) \ln \frac{q(\phi)}{p(\phi | X, \theta)} d\phi \tag{12}$$

Comment: This last equality is the EM master equation we have been aiming to arrive at. This equation will tell us (1) what we should do, and (2) why it will work.

- Notice that the goal has been more than just to arrive at an equality for $\ln p(X, \theta)$. We already had that with $\ln p(X, \theta) = \ln p(X, \theta, \phi) - \ln p(\phi | X, \theta)$. However, this equality doesn't immediately help us. We will see that the final equation above does give us a potentially useful algorithm.

- Let's look at the EM equation term-by-term.

  1. $\ln p(X, \theta)$: This is simply the objective function we want to optimize.

  2. $\mathcal{L}(\theta) := \int q(\phi) \ln \frac{p(X, \theta, \phi)}{q(\phi)} d\phi$: This is a function only of θ since we integrate ϕ out. Of course, the function $\mathcal{L}$ we end up with does depend very significantly on what $q(\phi)$ is.

  3. $\mathrm{KL}(q \| p) := \int q(\phi) \ln \frac{q(\phi)}{p(\phi | X, \theta)} d\phi$: This term is called the Kullback-Leibler divergence. It is very important and worth looking at more closely.

Kullback-Leibler (KL) divergence

- The KL-divergence is a function of two probability distributions. That is, the input to $\mathrm{KL}(q \| p)$ includes probability distributions $q(x)$ and $p(x)$ such that they are defined on the same support (i.e., the same values for x can be input to both of them). The output is a number.

- The KL-divergence is a similarity measure between q and p in that $\text{KL}(q\|p) \geq 0$ for all q and p and $\text{KL}(q\|p) = 0$ only in the case where $q(x) = p(x)$ for all x—that is, when q and p are exactly the same distribution.

- The KL-divergence is not technically a distance measure because, for that to be the case, we would need that for any three distributions q, p and f, $\text{KL}(q\|f) + \text{KL}(f\|p) \geq \text{KL}(q\|p)$ (called the triangle inequality), and this can be shown to not always be true.

- We can show that $\text{KL}(q\|p) \geq 0$ by using the concavity of $\ln(\cdot)$. Because $\ln(\cdot)$ is concave, by Jensen's inequality we can say that for any distribution $q(x)$ on x, $\mathbb{E}_q[\ln x] \leq \ln \mathbb{E}_q[x]$. This inequality itself requires a proof, which will be given in any convex optimization class. For now we will only use this property of the $\ln$ function. First, notice that

$$0 = \ln 1 = \ln \int p(x)dx = \ln \int q(x)\frac{p(x)}{q(x)}dx \tag{13}$$

The last in this seemingly trivial sequence of equalities can be interpreted as $\ln \mathbb{E}_q[\frac{p(x)}{q(x)}]$. From Jensen's inequality and the concavity of $\ln$,

$$0 \;=\; \ln \mathbb{E}_q\Big[\frac{p(x)}{q(x)}\Big] \;\geq\; \mathbb{E}_q \ln \frac{p(x)}{q(x)} \;=\; \int q(x) \ln \frac{p(x)}{q(x)}dx \tag{14}$$

And we immediately have the non-negativity of KL because

$$0 \geq \int q(x) \ln \frac{p(x)}{q(x)}dx \qquad \Leftrightarrow \qquad 0 \leq \int q(x) \ln \frac{q(x)}{p(x)}dx \;:=\; \text{KL}(q\|p)$$

Discussion on the EM equality

- Before presenting the EM algorithm and then showing why it works, we make two observations about the equality

$$\ln p(X,\theta) = \underbrace{\int q(\phi) \ln \frac{p(X,\theta,\phi)}{q(\phi)}d\phi}_{=\,\mathcal{L}(\theta)} + \underbrace{\int q(\phi) \ln \frac{q(\phi)}{p(\phi|X,\theta)}d\phi}_{=\text{KL}(q\|p)}$$

  1. For any *fixed* value of θ, changing $q(\phi)$ doesn't change what the RHS adds up to because the LHS only changes with θ. All that is changing with $q(\phi)$ is the breakdown of how much let first and second term of the RHS contributes to the fixed total.

  2. Since $\text{KL}(q\|p) \geq 0$, the term $\mathcal{L}(\theta) \leq \ln p(X,\theta)$ and we only have $\mathcal{L}(\theta) = \ln p(X,\theta)$ when $q(\phi) = p(\phi|X,\theta)$.

  3. One might be tempted to ask: If we know $p(\phi|X,\theta)$, can we just define $q(\phi) := p(\phi|X,\theta)$ and plug back in? The answer is yes, but this will defeat the purpose of EM. Notice that if we plug in this value for $q(\phi)$, we will get out that

$$\ln p(X,\theta) = \int p(\phi|X,\theta) \ln \frac{p(X,\theta,\phi)}{p(\phi|X,\theta)}d\phi$$

     We've simply found another way to write $\ln p(X,\theta)$ as a function only of θ, and if there's no easy solution for θ for the LHS, there won't be one for the RHS either. The trick is to keep $q(\phi)$ as a separate object from θ, but use one of them when updating the other.

The EM algorithm

- Using RHS of the EM equality one can devise a means for iteratively updating θ, and then prove that it is monotonically increasing $\ln p(X, \theta)$. That is, in the iterative algorithm below, we will get out a sequence $\theta_1, \theta_2, \theta_3, \ldots$. We first focus on how to get this sequence, then we will show that, using this procedure, the generated sequence has the property that

$$\ln p(X, \theta_1) \leq \ln p(X, \theta_2) \leq \ln p(X, \theta_3) \leq \cdots \tag{15}$$

- The EM algorithm is traditionally broken into two steps, an "E" (expectation) step and an "M" (maximization) step. Each iteration of the algorithm consists of one E and one M step, which are then repeated in the next iteration.

  E-Step at iteration t: Set $q_t(\theta) = p(\phi|X, \theta_{t-1})$ and calculate

$$\mathcal{L}_t(\theta) = \int q_t(\phi) \ln p(X, \theta, \phi) d\phi - \int q_t(\phi) \ln q_t(\phi) d\phi \tag{16}$$

  Notice that $q_t(\phi)$ is the conditional posterior using the value of θ from the previous iteration. Also, notice that the second term in $\mathcal{L}_t(\theta)$ is simply a constant w.r.t. θ.

  M-Step at iteration t: Treat $\mathcal{L}_t(\theta)$ as a function of θ and find the value of θ that maximizes it,

$$\theta_t \;=\; \arg\max_{\theta} \; \mathcal{L}_t(\theta) \tag{17}$$

  Because the second term in $\mathcal{L}_t(\theta)$ doesn't factor in this maximization, in practice it is only necessary to calculate $\int q_t(\phi) \ln p(X, \theta, \phi) d\phi$ and maximize this over θ.

- Before we analyze what this is doing, there are two crucial assumptions underlying this algorithm that would make it preferable to directly optimizing $\ln p(X, \theta)$ using gradient methods:

  1. First, we assume we can calculate $p(\phi|X, \theta)$ in closed form using Bayes rule. If we can't then we're already stuck when trying to update $\mathcal{L}(\theta)$, which requires this distribution.

  2. Second, in the M-step we have to optimize over $\mathcal{L}_t(\theta)$, which is a function of θ. The original thing we want to optimize over, $\ln p(X, \theta)$, is also a function of θ. If optimizing $\mathcal{L}_t(\theta)$ is no easier than $\ln p(X, \theta)$, then there's not much point to this.

     That is, if we can't solve $\nabla_\theta \mathcal{L}_t(\theta) = 0$ for θ analytically, then we're back where we started. While the following statement might not be true 100% of the time (but I can't think of a counter-example offhand), in big-picture terms if you need to use gradient methods to optimize $\mathcal{L}_t(\theta)$, you might as well just use gradient methods to optimize $\ln p(X, \theta)$ instead.

- Therefore, just like Gibbs sampling required an additional assumption about the model for it to be useful (that the conditional posterior distributions of all variables are in closed form), EM also makes assumptions about the model before claiming to be useful (#1 and #2 above). Of course, EM will technically work if #2 above isn't satisfied, which is something to at least keep in mind.

Analysis of the EM algorithm

- The last part of this discussion on EM will show that the sequence $\theta_1, \theta_2, \ldots$ we get from this procedure is monotonically increasing $\ln p(X, \theta)$. First, recall that

$$\mathcal{L}_t(\theta) = \int q_t(\phi) \ln \frac{p(X, \theta, \phi)}{q_t(\phi)} d\phi, \qquad \text{KL}(q_t(\phi) \| p(\phi|X, \theta)) = \int q_t(\phi) \ln \frac{q_t(\phi)}{p(\phi|X, \theta)} d\phi$$

  The following sequence of inequalities shows that $\ln p(X, \theta_{t-1}) \leq \ln p(X, \theta_t)$, followed by our elaboration on each line. In the transition from iteration $t-1$ to t we have that

$$\begin{aligned} \ln p(X, \theta_{t-1}) &= \mathcal{L}_t(\theta_{t-1}) + \text{KL}(q_t(\phi) \| p(\phi|X, \theta_{t-1})) && (18) \\ &= \mathcal{L}_t(\theta_{t-1}) && (19) \\ &\leq \mathcal{L}_t(\theta_t) && (20) \\ &\leq \mathcal{L}_t(\theta_t) + \text{KL}(q_t(\phi) \| p(\phi|X, \theta_t)) && (21) \\ &= \ln p(X, \theta_t) && (22) \end{aligned}$$

  1. The first line is simply the EM master equation using $q(\phi) \leftarrow q_t(\phi) = p(\phi|X, \theta_{t-1})$ and evaluating the functions at $\theta = \theta_{t-1}$.
  2. Since $q_t(\phi) = p(\phi|X, \theta_{t-1})$, the KL-divergence equals zero, so we can remove it. These two lines constitute what we do for the E-step.
  3. We know that $\mathcal{L}_t(\theta_{t-1}) \leq \mathcal{L}_t(\theta_t)$ because $\theta_t = \arg\max_\theta \mathcal{L}_t(\theta)$. This is the M-step.
  4. The next question is how $\mathcal{L}_t(\theta_t)$ relates to $\ln p(X, \theta_t)$. In the discussion above we have already been able to show that $\mathcal{L}(\theta) \leq \ln p(X, \theta)$. However, to see this again explicitly, we add a strategically chosen non-negative number to $\mathcal{L}_t(\theta_t)$. Specifically, we add the KL-divergence $\text{KL}(q_t(\phi) \| p(\phi|X, \theta_t))$. When we do this, we see that the second to last line is again simply the EM equation for $\ln p(X, \theta_t)$.

- Also worth emphasizing here is that $\text{KL}(q_t(\phi) \| p(\phi|X, \theta_t)) > 0$ because $q_t(\phi) \neq p(\phi|X, \theta_t)$ since $q_t(\phi) = p(\phi|X, \theta_{t-1})$ and we can assume $\theta_{t-1} \neq \theta_t$ (otherwise the algorithm has converged). Therefore, we can return to the E-step by updating $q(\phi)$ at iteration $t+1$ to account for the new value of θ. Hence there is a natural loop we can iterate back and forth between, where we update $q(\phi)$ and then update θ. The above inequalities show that any single completion of this cycle will find a new θ that is better than the old one in terms of maximizing $\ln p(X, \theta)$.

- The final question is whether the sequence $\theta_1, \theta_2, \ldots$ is converging to a local optimal solution of $\ln p(X, \theta)$. It is easy to think how we could have $\ln p(X, \theta_t) \leq \ln p(X, \theta_{t+1})$ for all t, but also have θ_∞ not be at the top of one of the "hills" of $\ln p(X, \theta)$. It can be shown that EM does converge to one of these peaks, but the proof is significantly more complicated and so we skip it in this class. Suffice it to say that EM will give the MAP solution (or a local optimal of it) for any interesting model you will come across.

The EM algorithm for probit regression

- Next, we derive an EM algorithm for the probit regression model. In this model the weights w correspond to θ. We first need to find a good ϕ that will make things work out nicely. It's worth

dwelling on this for a while before jumping to the algorithm. Even though this qualifies as a straightforward application of EM, this is a good example of how the generic view taken above doesn't quite make deriving an EM algorithm for a specific problem an automatic procedure.

- For this model, the joint likelihood factorizes as

$$p(\vec{y}, w|X) = p(w)\prod_{i=1}^{N} p(y_i|w, x_i) \tag{23}$$

- With EM, our goal is to find a random variable, ϕ such that

$$\int p(\vec{y}, w, \phi|X)d\phi = p(\vec{y}, w|X) \tag{24}$$

- Actually, here we will see how our above simplified version of EM, where we had one θ and one ϕ, can be generalized. Instead, let $\phi = (\phi_1, \ldots, \phi_N)$. We pick these such that

$$\int p(\vec{y}, w, \phi|X)d\phi = \prod_{i=1}^{N} \int p(y_i, \phi_i, w|X)d\phi_i = p(\vec{y}, w|X) \tag{25}$$

  Therefore, we assume that the variables ϕ_i are independent from each other and each (x_i, y_i) pair has a ϕ_i associated with it in some way.

- Notice that by assuming that the ϕ_i are independent of each other and only dependent on (x_i, y_i) and w, the posterior on the vector $\phi = (\phi_1, \ldots, \phi_N)$ factorizes as well

$$\begin{aligned} p(\phi|\vec{y}, w, X) &= \frac{\prod_{i=1}^{N} p(y_i|\phi_i, w, x_i)p(\phi_i|w, x_i)}{\prod_{i=1}^{N} \int p(y_i|\phi_i, w, x_i)p(\phi_i|w, x_i)d\phi_i} \\ &= \prod_{i=1}^{N} \frac{p(y_i|\phi_i, w, x_i)p(\phi_i|w, x_i)}{\int p(y_i|\phi_i, w, x_i)p(\phi_i|w, x_i)d\phi_i} \\ &= \prod_{i=1}^{N} p(\phi_i|y_i, w, x_i) \end{aligned} \tag{26}$$

- How does this impact the EM equation? We have that

$$\ln p(\vec{y}, w|X) = \sum_{i=1}^{N} \ln p(y_i, w|x_i) \tag{27}$$

  For a single $\ln p(y_i, w|x_i)$ we have from the derivation above that

$$\ln p(y_i, w|x_i) = \int q(\phi_i) \ln \frac{p(y_i, w, \phi_i|x_i)}{q(\phi_i)}d\phi_i + \int q(\phi_i) \ln \frac{q(\phi_i)}{p(\phi_i|y_i, w, x_i)}d\phi_i \tag{28}$$

  and so summing the LHS and RHS over i we have that

$$\ln p(\vec{y}, w|X) = \sum_{i=1}^{N} \int q(\phi_i) \ln \frac{p(y_i, w, \phi_i|x_i)}{q(\phi_i)}d\phi_i + \sum_{i=1}^{N} \int q(\phi_i) \ln \frac{q(\phi_i)}{p(\phi_i|y_i, w, x_i)}d\phi_i \tag{29}$$

- To belabor the point, we could have gotten to this last EM equation from the other direction, and it can be worthwhile to see it from this perspective as well w.r.t. choosing q. Let ϕ be the latent variable we introduce—possibly a vector. From EM we know that

$$\ln p(\vec{y}, w|X) = \int q(\phi) \ln \frac{p(\vec{y}, \phi, w|X)}{q(\phi)} d\phi + \int q(\phi) \ln \frac{q(\phi)}{p(\phi|\vec{y}, w, X)} d\phi \tag{30}$$

- Next, assume that we pick $\phi = (\phi_1, \ldots, \phi_N)$ such that $p(\vec{y}, \phi, w|X) = \prod_{i=1}^N p(y_i, \phi_i, w|x_i)$. Then we showed above that $p(\phi|\vec{y}, w, X) = \prod_{i=1}^N p(\phi_i|y_i, w, x_i)$. From the EM algorithm, we know that we need to use this posterior to select q,

$$q(\phi) = p(\phi|\vec{y}, w, X) = \prod_{i=1}^N p(\phi_i|y_i, w, x_i) \tag{31}$$

  This implies a *factorization* of the $q(\phi)$ distribution (which results in the above EM equation)

$$q(\phi) = \prod_{i=1}^N q(\phi_i) \qquad \text{and} \qquad q(\phi_i) = p(\phi_i|y_i, w, x_i) \tag{32}$$

- Therefore, our goal is threefold:

  1. Define ϕ_i such that

$$\int p(y_i, \phi_i|w, x_i) d\phi_i = p(y_i|w, x_i) = \Phi\Big(\frac{x_i^T w}{\sigma}\Big)^{y_i} \left[1 - \Phi\Big(\frac{x_i^T w}{\sigma}\Big)\right]^{1-y_i}$$

  2. Derive a closed-form posterior distribution $p(\phi_i|y_i, x_i, w)$ using Bayes rule.

  3. Calculate the expectation $\mathcal{L}(w)$ and find that we can analytically find the value of w that maximizes it.

- In general, once #1 is done, #2 and #3 follow immediately (or on the flipside, we immediately realize that the ϕ we picked won't work and we should try again). It's step #1 that requires the insights that come from experience working with models, and that can't be reduced to a set of instructions like #2 and #3 can. Therefore, #1 is going to appear like it's coming out of nowhere below, while #2 and #3 will be more straightforward.

- Step 1: Consider the following expanded model

$$y_i = \mathbb{1}(\phi_i > 0), \quad \phi_i \sim \text{Normal}(x_i^T w, \sigma^2) \tag{33}$$

  Notice that even though we have an indicator function for y_i, we can still view this as a (deterministic) probability distribution: $p(y_i = 1|\phi_i) = \mathbb{1}(\phi_i > 0)$. In other words, y_i isn't random given ϕ_i, but it makes perfect mathematical sense to write the joint likelihood

$$\begin{aligned} p(y_i = 1, \phi_i|w, x_i) &= p(y_i = 1|\phi_i) p(\phi_i|w, x_i) \\ &= \mathbb{1}(\phi_i > 0)(2\pi\sigma^2)^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(\phi_i - x_i^T w)^2} \end{aligned} \tag{34}$$

For $y_i = 0$, we use the indicator $\mathbb{1}(\phi_i \leq 0)$ instead. Next we need to calculate the marginal distribution $p(y_i|w, x_i)$. Clearly

$$\begin{aligned}\int p(y_i = 1, \phi_i|w, x_i)d\phi_i &= \int_{-\infty}^{\infty} \mathbb{1}(\phi_i > 0)(2\pi\sigma^2)^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2\sigma^2}(\phi_i - x_i^T w)^2} \\ &= \int_0^{\infty} (2\pi\sigma^2)^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2\sigma^2}(\phi_i - x_i^T w)^2} \\ &= P(\phi_i > 0) \end{aligned} \tag{35}$$

However, remember that we need to show that this equals $\Phi(x_i^T w/\sigma) = \int_{-\infty}^{x_i^T w/\sigma}(2\pi)^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2}s^2}ds$. To do this, we first observe that we can draw a random variable $\phi_i \sim \text{Normal}(x_i^T w, \sigma^2)$ as follows

$$\phi_i = x_i^T w + \sigma s, \qquad s \sim \text{Normal}(0, 1) \tag{36}$$

Therefore, the probability

$$P(\phi_i > 0) = P(x_i^T w + \sigma s > 0) = P(s > -x_i^T w/\sigma)$$

The final step is to recognize that $P(s > -x_i^T w/\sigma) = P(s \leq x_i^T w/\sigma)$ since s is a standard normal distribution symmetric around zero. Therefore,

$$\int p(y_i = 1, \phi_i|w, x_i)d\phi_i = P(\phi_i > 0) = P(s \leq x_i^T w/\sigma) = \Phi(x_i^T w/\sigma) \tag{37}$$

- We have therefore found a hierarchical expansion of the probit regression model,

$$y_i = \mathbb{1}(\phi_i > 0), \quad \phi_i \sim \text{Normal}(x_i^T w, \sigma^2), \quad w \sim \text{Normal}(0, \lambda^{-1}I)$$

If we integrate out all the ϕ_i in this model, we return to the original probit regression model. Before moving on, it's worth discussing this a little more. First, there's nothing inherently "right" or "correct" about a probit model. Because we picked that as our desired model, we had to do some work to get to this hierarchical representation for EM. However, the probit model was only picked because it "makes sense" to do it.

The point I am making is that, in my opinion, the above model makes a lot of sense too—we could have made *this* model our starting point. In that case, we could have done MAP on w and the vector ϕ. Or, we could have decided to *integrate out* all uncertainty in ϕ and do MAP only for w. As shown below, EM would then give us conditional posterior distributions on the ϕ_i rather than a point estimate. In general, the more variables you can integrate out the better (one reason is that the model will avoid over-fitting). Coming at EM from this perspective, the problem is much easier since we already know the "hidden" variable ϕ to integrate out—after all, it was part of the definition of the model in this case.

- Step 2: We've found a latent variable ϕ_i that gives the correct marginal distribution. Next we need to calculate the posterior $p(\phi_i|y_i, x_i, w)$. By Bayes rule,

$$\begin{aligned} p(\phi_i|y_i, x_i, w) &= \frac{p(y_i|\phi_i)p(\phi_i|x_i, w)}{\int p(y_i|\phi_i)p(\phi_i|x_i, w)d\phi_i} \\ &= \frac{\mathbb{1}\{\text{sign}(\phi_i) = 2y_i - 1\}\mathrm{e}^{-\frac{1}{2}(\phi_i - x_i^T w)^2}}{\int_{-\infty}^{\infty} \mathbb{1}\{\text{sign}(\phi_i) = 2y_i - 1\}\mathrm{e}^{-\frac{1}{2}(\phi_i - x_i^T w)^2}d\phi_i} \end{aligned} \tag{38}$$

Unfortunately the indicator doesn't look nice, but all it is saying is that ϕ_i must be positive if $y_i = 1$ and must be negative if $y_i = 0$. This distribution is called a *truncated normal distribution*.

- If $y_i = 1$, then the truncated normal distribution $TN_1(x_i^T w, \sigma^2)$ is defined to be the part of Normal$(x_i^T w, \sigma^2)$ defined on $\mathbb{R}_+$ re-normalized to give a probability distribution. In other words, it's the distribution of a Gaussian random variable *conditioned on* knowledge that it is positive. The reverse holds when $y_i = 0$, in which case we can write $TN_0(x_i^T w, \sigma^2)$ defined on $\mathbb{R}_-$.

- Step 3: Finally, we need to calculate

$$\mathcal{L}(w) = \sum_{i=1}^{N} \mathbb{E}_q[\ln p(y_i, \phi_i, w|x_i)] + \text{constant} \tag{39}$$

Since the joint distribution including the extra ϕ variables is

$$p(\vec{y}, \phi, w|X) = p(w) \prod_{i=1}^{N} p(y_i|\phi_i) p(\phi_i|x_i, w) \tag{40}$$

we have that

$$\mathcal{L}(w) = -\frac{\lambda}{2} w^T w + \sum_{i=1}^{N} \underbrace{\mathbb{E}_q[\ln \mathbb{1}\{\text{sign}(\phi_i) = 2y_i - 1\}]}_{=0} - \frac{1}{2\sigma^2} \mathbb{E}_q[(\phi_i - x_i^T w)^2] + \text{const.} \tag{41}$$

One of the expectations always equals zero, since $q(\phi_i) = TN_{y_i}(x_i^T w, \sigma^2)$, and so it is only integrating over values of ϕ_i for which $\mathbb{1}\{\text{sign}(\phi_i) = 2y_i - 1\} = 1$. Also, if we expand the square of the rightmost term, we can put anything not involving w into the constant, since we want to maximize $\mathcal{L}(w)$ over w. As a result, we want to maximize

$$\mathcal{L}(w) = -\frac{\lambda}{2} w^T w - \sum_{i=1}^{N} \frac{1}{2\sigma^2} (w^T x_i x_i^T w - 2 w^T x_i \mathbb{E}_q[\phi_i]) + \text{constant} \tag{42}$$

Solving for $\nabla_w \mathcal{L}(w) = 0$, we find that

$$w = \arg\max_w \ \mathcal{L}(w) \qquad \Leftrightarrow \qquad w = \left(\lambda I + \sum_{i=1}^{N} x_i x_i^T / \sigma^2 \right)^{-1} \left(\sum_{i=1}^{N} x_i \mathbb{E}_q[\phi_i] / \sigma^2 \right) \tag{43}$$

We just need to know what $\mathbb{E}_q[\phi_i]$ is under the conditional posterior $q(\phi_i) = TN_{y_i}(x_i^T w, \sigma^2)$. Looking this up in a textbook or on Wikipedia, we find that

$$\mathbb{E}_q[\phi_i] = \begin{cases} x_i^T w + \sigma \times \dfrac{\Phi'(-x_i^T w/\sigma)}{1 - \Phi(-x_i^T w/\sigma)} & \text{if } \ y_i = 1 \\ \\ x_i^T w + \sigma \times \dfrac{-\Phi'(-x_i^T w/\sigma)}{\Phi(-x_i^T w/\sigma)} & \text{if } \ y_i = 0 \end{cases} \tag{44}$$

The function $\Phi'(s)$ is the probability density function (PDF) of a Normal$(0, 1)$ distribution evaluated at s. $\Phi(s)$, as before, is the CDF of a Normal$(0, 1)$ distribution evaluated at s. These can be quickly evaluated by calling a built-in function.

- Notice that, even though it took several steps to get to a final EM algorithm, we have everything we want:

    1. An expression for the conditional posterior distribution $p(\phi_i|y_i, w, x_i)$ as a known distribution (even though it's an atypical distribution, it's still one we know how to take the expectation with respect to, which we observe is all that matters here)

    2. A closed form expression for updating w that we can evaluate quickly in code without having to use iterative gradient methods.

- It might take a few readings of the above to appreciate all they nuances of the EM algorithm for probit regression and why it's correct (i.e., maximizes $\ln p(\vec{y}, w|X)$ over w). However, the final algorithm can be summarized very easily.

An EM algorithm for probit regression

1. Initialize w_0 to a vector of all zeros.

2. For iteration $t = 1, \ldots, T$

    (a) E-Step: Calculate the vector $\mathbb{E}_{q_t}[\phi] = \big(\mathbb{E}_{q_t}[\phi_1], \ldots, \mathbb{E}_{q_t}[\phi_N]\big)$, where

$$\mathbb{E}_{q_t}[\phi_i] = \begin{cases} x_i^T w_{t-1} + \sigma \times \dfrac{\Phi'(-x_i^T w_{t-1}/\sigma)}{1 - \Phi(-x_i^T w_{t-1}/\sigma)} & \text{if } y_i = 1 \\ x_i^T w_{t-1} + \sigma \times \dfrac{-\Phi'(-x_i^T w_{t-1}/\sigma)}{\Phi(-x_i^T w_{t-1}/\sigma)} & \text{if } y_i = 0 \end{cases}$$

    (b) M-Step: Update the vector w using the expectations above in the following equation

$$w_t = \left(\lambda I + \sum_{i=1}^{N} x_i x_i^T/\sigma^2\right)^{-1} \left(\sum_{i=1}^{N} x_i \mathbb{E}_{q_t}[\phi_i]/\sigma^2\right)$$

    (c) Calculate $\ln p(\vec{y}, w_t|X)$ using the equation

$$\begin{aligned} \ln p(\vec{y}, w_t|X) &= \frac{d}{2}\ln\left(\frac{\lambda}{2\pi}\right) - \frac{\lambda}{2} w_t^T w_t \\ &\quad + \sum_{i=1}^{N} y_i \ln \Phi(x_i^T w_t/\sigma) + \sum_{i=1}^{N} (1 - y_i) \ln(1 - \Phi(x_i^T w_t/\sigma)) \end{aligned}$$

- Part 2c can be used to assess convergence and therefore determine what T should be. Practically speaking, it can also be used to make sure your implementation is correct, since we know from the earlier proof that $\ln p(\vec{y}, w_t|X)$ must be monotonically increasing in t. If you find that this is not the case with your implementation, then you can be sure there is a bug somewhere.