Maximum Marginal Likelihood and Posterior Approximations with Monte Carlo Expectation Maximization

Bob Carpenter

9 August 2018

Contents

Maximum Marginal Likelihood 2
Estimating Missing Data 3
The Expectation Maximization Algorithm 4
Monte Carlo Expectation Maximization 5
Stochastic Averaging Expectation Maximization 7
Generalized Expectation Maximization 7
Gradient-Based Monte Carlo Expectation Maximization 8
Gradient-Based Marginal Optimization 8
Appendix: Maximum Marginal a Posteriori Approximations 9
Appendix: Laplace Approximation 11
Appendix: Convergence of Expectation Maximization 12
Appendix: Analytic Expectation Maximization 13
Historical Notes 15
References 15

Abstract

This note introduces classical and Bayesian application of the expecation maximization (EM) algorithm with full details required for implementations. EM calculates maximum marginal likelihood (MML) estimates in the presence of missing data, which is marginalized out of the complete data likelihood. The algorithm iteratively refines a current guess of the parameters by maximizing marginal likelihood with respect to the expectations for missing data with the current parameter guess. Standard errors are calculated with observed Fisher information.

In the Bayesian setting, the max marginal a posterior (MMAP) parameter values can be estimate by EM. These may then be used to locate a Laplace approximation of the posterior. Posterior predictive inference may be carried out using the Laplace approximation

directly, using sampling from the approximation, or by using the approximation as an importance sampling generator.

Monte Carlo (MC) methods enable general expectation calculations and may also be used to calculate derivatives of the expectations being maximized. Generalized expectation maximization (G-EM) replaces the maximization with a hill-climbing step. Stochastic averaging expectation maximization (SA-EM) smoothes expectation estimates over iterations.

Maximum Marginal Likelihood

The traditional application of the expectation maximization algorithm is to computing maximum marginal likelihood estimates. Maximum marginal estimates are required in situations where simple maximum likelihood estimates do not exist.

Complete data likelihood

Suppose we have a data sampling density $p(y, u \mid \theta)$ where

- y is observed data,
- u is missing data, and
- θ is a parameter vector.

Missing data in this context may be construed very broadly. It may literally be missing values in a survey, such as covariates or outcomes, or missing measurements from a broken monitor. The missing data may also be the censored times of death for animals in a mark-recapture study or patients in a survival analysis.[1] In other cases, the missing data is actually a latent item-level parameter, such as varying slopes in a multilevel regression or mixture-model responsibilities.[2]

If both the data y and the missing data u are known, then $p(y, u \mid \theta)$, considered as a function of the parameter vector θ, is known as the *complete data likelihood function*,[3]

$$\mathcal{L}(\theta) = p(y, u \mid \theta).$$

Because the missing data u will not be known in practice, the complete data likelihood is of limited use by itself. It is tempting to try to derive a maximum likelihood estimate for the missing data and parameters,[4]

$$\begin{aligned}(u^*, \theta^*) &= \text{argmax}_{(u,\,\theta)}\; p(y, u \mid \theta). \\ &= \text{argmax}_{(u,\,\theta)}\; p(y \mid u, \theta).\end{aligned}$$

For many problems of interest, such as hierarchical and multilevel models, there are no finite u and θ that jointly maximize $p(y, u \mid \theta)$.[5]

[1] The censoring is because a patient survived beyond the end of the study or an animal was last observed alive at a given time; in both cases, their times of death must be imputed.

[2] Item-level effects are often referred to as "random effects." The "fixed effects" (i.e., parameters not varying by level) are included in θ.

[3] Traditionally in frequentist statistics and probability theory, $p(y, u \mid \theta)$ would be written as $p(y, u; \theta)$ to distinguish the variables u and y (in Roman letters) as being random and the variable θ is a fixed, but unknown parameter, and hence note modeled as random. Here, we use Bayesian notation throughout despite its implication that θ is being treated as random when being used conditionally.

[4] The second step follows by Bayes's theorem, which applies in this frequentist setting because u is being modeled as random.

[5] An example is when u are item-level effects governed by group-level parameters in θ. As the group-level variance goes to zero and the item-level effects approach the group-level mean, the likelihood grows without bound.

Marginal data likelihood

To get around the technical problem of maximum likelihood estimation when there is no maximum, the unobserved data can be marginalized out, leaving the *marginal data likelihood*, which is a function of θ for fixed observed data y,[6]

$$\begin{aligned} \mathcal{L}(\theta) &= p(y \mid \theta) \\ &= \int_U p(y, u \mid \theta) \, \mathrm{d}u \\ &= \mathbb{E}_{p(u \mid \theta)}[\, p(y \mid u, \theta) \,]. \end{aligned}$$

[6] The integral over the domain U of u may involve summation if U has discrete components.

Maximum marginal likelihood estimator

The *maximum marginal likelihood* (MML) estimate[7] for θ is defined to be the value of the parameters that maximizes the marginal likelihood function,

$$\begin{aligned} \theta^* &= \mathrm{argmax}_\theta \; \mathcal{L}(\theta) \\ &= \mathrm{argmax}_\theta \; p(y \mid \theta). \\ &= \mathrm{argmax}_\theta \; \mathbb{E}_{p(u \mid \theta)}[\log p(y \mid u, \theta) \,]. \end{aligned}$$

The final formulation as an expectation points the way toward the EM algorithm.

[7] An *estimator* is just a function from data to a numerical value of a parameter, called an *estimate*. Representing the data as random variables, the estimate will be a derived random variable, the properties of which determine quantities of interest such as standard errors and confidence intervals.

Penalized Maximum Marginal Likelihood

Shrinking estimates toward zero can reduce the variance of the estimator at the cost of introducing some bias.[8] The goal is to improve predictive accuracy by reducing overfitting. More generally, regularization pulls estimators toward a predefined value; shrinkage regularizes to zero.

Regularization may be accomplished by adding a penalty term to the likelihood function. The *penalized likelihood* function is

$$\mathcal{L}_f(\theta) = \mathcal{L}(\theta) + f(\theta).$$

where $f(\theta)$ is a penalty function.[9]

Nothing changes for the EM algorithm if the likelihood function $\mathcal{L}(\theta)$ is replaced with a penalized likelihood function $\mathcal{L}_f(\theta)$.

[8] As a function of random data y, an estimator's *expectation* is

$$\mathbb{E}_{p(y|\theta)}[\hat{\theta}(y)] = \int_Y \hat{\theta}(y) \cdot p(y \mid \theta) \, \mathrm{d}y.$$

The *mean square error* of an estimator is

$$\begin{aligned} \mathrm{mse}(\hat{\theta}(y)) &= \mathbb{E}_{p(y|\theta)}\left[(\hat{\theta}(y) - \theta)^2\right] \\ &= \int_Y (\hat{\theta}(y) - \theta)^2 \cdot p(y \mid \theta) \, \mathrm{d}y. \end{aligned}$$

The *variance* of an estimator is its variance,

$$\begin{aligned} \mathrm{var}(\hat{\theta}(y)) &= \mathbb{E}_{p(y|\theta)}\left[\mathrm{var}(\hat{\theta}(y))\right]. \\ &= \int_Y \left(\hat{\theta}(y) - \mathbb{E}_{p(y|\theta)}[\hat{\theta}(y)]\right)^2 \cdot p(y \mid \theta) \, \mathrm{d}y. \end{aligned}$$

The *bias* of an estimator is its expected error,

$$\begin{aligned} \mathrm{bias}(\hat{\theta}) &= \mathbb{E}_{p(y|\theta)}\left[\hat{\theta}(y) - \theta\right]. \\ &= \int_Y \left(\hat{\theta}(y) - \theta\right) \cdot p(y \mid \theta) \, \mathrm{d}y. \end{aligned}$$

[9] A common penalty function is the quadratic,

$$\begin{aligned} f_\lambda(\theta) &= -\lambda \cdot \theta^\top \cdot \theta \\ &= -\lambda \cdot \textstyle\sum_{k=1}^K \theta_k{}^2, \end{aligned}$$

where $\lambda > 0$ is a tuning parameter controlling the amount of shrinkage. This penalty is equivalent to the log density of a normal prior on θ, but frequentists use the term "penalty" to avoid treating θ as random.

Estimating Missing Data

Often the missing data is of interest itself. Given the maximum marginal likelihood estimate for the parameters,

$$\begin{aligned}\theta^* &= \text{argmax}_\theta \; p(y \mid \theta) \\ &= \int_U p(y, u \mid \theta) \; \mathrm{d}u.\end{aligned}$$

the missing data may be estimated by fixing the parameter value and maximizing the joint likelihood for u,

$$u^* = \text{argmax}_u \; p(y, u \mid \theta^*).$$

Uncertainty in missing data

The observed Fisher information matrix may be used to estimate standard errors using the curvature of the log density for the missing data u at the maximum marginal likelihood estimate θ^*. The estimated *standard error* for the estimate u_n^* is given by

$$\text{se}(u_n^*) \approx \left(-\frac{\partial^2}{\partial {u_n}^2} \log p(y, u \mid \theta^*) \right)^{-\frac{1}{2}}.$$

This standard error estimate is biased in that it systematically underestimates the true error due to the use of a fixed value of θ^*.[10]

[10] This is the same way that the maximum likelihood estimator for variance is biased to underestimate because it is based on the sample mean, not the true mean. The unbiased estimate (known as *sample variance*), divides the sum of square differences from the sample mean by $N - 1$ rather than N as is done for the MLE.

The Expectation Maximization Algorithm

The *expectation maximization* (EM) algorithm calculates maximum marginal likelihood estimates by iteratively performing maximizations over an expectation.[11] The algorithm starts with a randomly initialized parameter value, then alternatively calculates the expectation of the missing data given the current parameter value, then maximizes the parameter value given the expectation of the missing data.

[11] It may also be used to calculate maximum marginal penalized likelihood estimates and find posterior modes of a Bayesian model.

The Algorithm

The EM algorithm involves the following steps.[12]

- *Inputs:* (a) complete data likelihood function $p(y, u \mid \theta)$ for observed data y, missing data u, and parameters θ, and (b) the observed data y.

1. Initialize $\theta^{(0)}$ such that $p\left(y \mid \theta^{(0)}\right) > 0$.
2. While the sequence has not converged,[13]

$$\begin{aligned}\theta^{(t+1)} &= \text{argmax}_\theta \; Q(\theta \mid \theta^{(t)}) \\ &= \text{argmax}_\theta \; \mathbb{E}_{p\left(u \mid y, \theta^{(t)}\right)}[\log \; p(y, u \mid \theta)] \\ &= \text{argmax}_\theta \; \int_U \log p(y, u \mid \theta) \cdot p(u \mid y, \theta^{(t)}) \, \mathrm{d}u.\end{aligned}$$

[12] Statistics notation is heavily overloaded. The y in $p(y, u \mid \theta)$ is a bound variable the name of which is chosen to disambiguate $p()$, which is overloaded for every density. The observed data y is a bona fide value. To make matters even more confusing, the observed data y will be plugged into the likelihood function $p(y, u \mid \theta)$, setting $y = y$ in notation that should be near and dear to users of R.

[13] Convergence is determined when the difference between successive updates falls below an absolute threshold,

$$\left| \theta^{(t+1)} - \theta^{(t)} \right| < \epsilon^{\text{abs}},$$

or relative threshold,

$$\frac{\left| \theta^{(t+1)} - \theta^{t} \right|}{\frac{1}{2}\left(\left| \theta^{(t)} \right| + \left| \theta^{(t+1)} \right| \right)} < \epsilon^{\text{rel}}.$$

3. Return the final value $\theta^{(t)}$.

The quantity $Q(\theta \mid \theta^{(t)})$ is the *expected log complete data likelihood* for parameters θ where the expectation is taken over the missing data u given the previous parameters $\theta^{(t)}$.

This is just a framework for an algorithm—it does not specify how to calculate the innner expectation or perform the maximization. Traditionally, these were calculated analytically; see the appendix on analytic EM for an example.

Monte Carlo Expectation Maximization

Monte Carlo expectation maximization (MC-EM) uses Monte Carlo methods to evaluate the integral in the definition of

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= \mathbb{E}_{p\left(u \mid y, \theta^{(t)}\right)}[\log\, p(y, u \mid \theta)] \\ &= \int_U \log p(y, u \mid \theta) \cdot p(u \mid y, \theta^{(t)})\, \mathrm{d}u. \end{aligned}$$

All we need for a Monte Carlo solution to this integral is a way to simulate independent draws for u conditioned on y and $\theta^{(t)}$, i.e.,

$$u^{(1)}, \ldots u^{(M)} \sim p\left(u \mid y, \theta^{(t)}\right).$$

Given these draws, the Monte Carlo estimator for $Q(\theta \mid \theta^{(t)})$ using M simulation draws is

$$\hat{Q}_M\left(\theta \mid \theta^{(t)}\right) = \frac{1}{M} \cdot \sum_{m=1}^{M} \log p\left(y, u \mid \theta\right)$$

The central limit theorem ensures that the Monte Carlo estimate converges to the true value,

$$\lim_{M \to \infty} \hat{Q}_M\left(\theta \mid \theta^{(t)}\right) = Q\left(\theta \mid \theta^{(t)}\right).$$

The rate of convergence, i.e., rate of reduction in expected error, is $\mathcal{O}\left(1/\sqrt{M}\right)$.[14]

[14] For example, an estimate based on 100 draws has ten times the expected error of one based on 10,000 draws.

For simple models, like a mixture model, it is relatively straightforward to take independent draws of u conditioned on y and $\theta^{(t)}$. For more general models, we need either an approximation from which it is easier to sample (and perhaps adjust via importance sampling) or we need to move to Markov chain Monte Carlo methods. For the time being, we will simply assume we can draw the required $u^{(1)}, \ldots, u^{(M)}$ in order to sketch out the Monte Carlo EM algorithm.

Given the estimator $\hat{Q}_M\left(\theta \mid \theta^{(t)}\right)$ for $Q\left(\theta \mid \theta^{(t)}\right)$, the M-step of the expectation-maximization algorithm becomes

$$\begin{aligned} \theta^{(t+1)} &= \text{argmax}_\theta\, \hat{Q}_M\left(\theta \mid \theta^{(t)}\right) \\ &= \text{argmax}_\theta\, \tfrac{1}{M} \cdot \textstyle\sum_{m=1}^{M} \log p\left(y \mid u^{(m)}, \theta\right) \end{aligned}$$

Putting this all together gives us the *Monte Carlo expectation maximization* (MC-EM) algorithm for maximum marginal likelihood estimates,

1. Set $\theta^{(0)}$ to a value such that $p\left(y \mid \theta^{(0)}\right) > 0$.
2. While the sequence has not converged,
   i. draw
   $$u^{(t,1)}, \ldots, u^{(t,M)} \sim p\left(u \mid y, \theta^{(t)}\right)$$
   ii. set
   $$\theta^{(t+1)} \;=\; \text{argmax}_\theta \; \frac{1}{M} \cdot \sum_{m=1}^{M} \log p\left(y \mid u^{(t,m)}, \theta\right).$$
3. Return the last $\theta^{(t)}$.

Number of Monte Carlo simulations and precision

In early iterations, the value of $\theta^{(t)}$ is far from the true value. In these situations, a small number of Monte Carlo draws $u^{(m)}$ may be more computationally effective than trying to estimate $Q\left(\theta \mid \theta^{(t)}\right)$ more precisely. During later iterations when $\theta^{(t)}$ is close to the true value of θ, it will be necessary to increase M in order to increase precision in $\theta^{(t)}$. How many steps are necessary will depend on the precision required in the final answer.

The Monte Carlo EM algorithm is easily modified to accomodate a schedule $M_1, \ldots, M_t, \ldots$ of draw sizes. These may even be chosen adaptively based on the history $\theta^{(1)}, \ldots \theta^{(t)}$.

Markov Chain Monte Carlo Expectation Maximization

The Monte Carlo expectation maximization algorithm assumes the $u^{(t,m)}$ can be drawn indepenedently. In black-box situations, it is often possible to take draws using Markov chain Monte Carlo (MCMC) when there is no easy way to take independent draws. Rather than being independent, the draws $u^{(t,1)}, \ldots, u^{(t,M)}$ form a Markov chain.[15]

The EM algoirthm works the same way if the draws of $u^{(t,m)}$ are made using Markov chain Monte Carlo (MCMC) instead of being independent Monte Carlo draws. Convergence follows from the Markov chain Monte Carlo central limit theorem (MCMC CLT) at a rate of at least $\mathcal{O}\left(1/\sqrt{M_{\text{eff}}}\right)$, where M_{eff} is the minimum effective sample size for MCMC draws.[16] The resulting algorithm is the *Markov chain Monte Carlo expectation maximization* (MCMC-EM) algorithm.

[15] A Markov chain $y_1, \ldots, y_N$ is a sequence of variables such that

$$p(y_{n+1} \mid y_1, \ldots, y_n) = p(y_{n+1} \mid y_n).$$

The variables in a Markov chain are most often autocorrelated rather than independent or anticorrelated.

[16] The *effective sample size* M_{eff} for a sequence of draws indicates the number of independent draws that would provide the same error bounds.

Stochastic Averaging Expectation Maximization

The *stochastic averaging expectation maximization* (SA-EM) algorithm replaces the independent estimates $\hat{Q}_M\left(\theta \mid \theta^{(t)}\right)$ with a rolling average, the value for which at step t will be written with a subscript as $\hat{Q}_M^{(t)}\left(\theta \mid \theta^{(t)}\right)$.

At step zero, the estimate is set the same way as before,

$$\hat{Q}_M^{(0)}\left(\theta \mid \theta^{(t)}\right) \;=\; \hat{Q}_M\left(\theta \mid \theta^{(t)}\right).$$

At each iteration, the estimate is updated by averaging the estimate based on the current parameters and the previous estimate,

$$\hat{Q}_M^{(t+1)}\left(\theta \mid \theta^{(t)}\right) \;=\; \lambda \cdot \hat{Q}_M\left(\theta \mid \theta^{(t)}\right) \;+\; (1-\lambda) \cdot \hat{Q}_M^{(t)}\left(\theta \mid \theta^{(t)}\right).$$

The (MC)MC estimate $\hat{Q}_M$ is weighted by λ and the previous estimate $\hat{Q}_M^{(t)}$ is weighted by $(1-\lambda)$. This has the effect of regularizing the estimate of $\hat{Q}_M^{(t)}$. The intended goal is, as usual with regularization, to reduce the expected error of $\hat{Q}_M^{(t)}$ relative to $\hat{Q}_M$ by reducing the variance.

The update rate λ may vary with the iteration; just replace λ with λ_t and provide some schedule for the values of λ_t rather than a fixed value of λ. This may be useful, for example, in conjunction with schedules for increasing M over iterations.

This regularization strategy may be used with any of the other EM algorithms described in this note.

Generalized Expectation Maximization

It turns out that replacing the maximization step of EM with a simple hill-climbing step does not affect convergence of the algorithm (other than rate). The resulting algorithm where maximization is replaced with a a hill-climbing step is called *generalized expectatino maximization* (G-EM). This M-step modification can be applied independently of any of the E-step variations discussed in this note.

Stochastic Generalized Expectation Maximization

The hill-climibing step in the M-step of the EM algorithm may be done stochastically. That is, it may be done using a subsample of the data so that it is only hill climbing in expectation.

Usually the update rate with a stochastic hill-climbing algorithm must be carefully scheduled over time to ensure convergence.[17]

[17] The schedule for stochastic maximization is usually required to meet the Robbins-Monro conditions, i.e.,

$$\sum \epsilon_t = \infty,$$

and

$$\sum {\epsilon_t}^2 < \infty.$$

Gradient-Based Monte Carlo Expectation Maximization

For generalized models where the maximization in the M-step must be done with generic optimization techniques, gradients can be used to accelerate optimization.

Because expectations and derivatives are linear operators, we can distribute derivatives through them,

$$\frac{\partial}{\partial\theta}\,\mathbb{E}[f(\theta)] \;=\; \mathbb{E}\left[\frac{\partial}{\partial\theta}\,f(\theta)\right].$$

The case of interest is where the function is the log density with respect to the draw $u^{(t,m)}$,

$$\begin{aligned}
\frac{\partial}{\partial\theta}\,\hat{Q}_M\Big(\theta \mid \theta^{(t)}\Big) &= \frac{\partial}{\partial\theta}\int p(y,u,\mid\theta)\cdot p(u \mid y,\theta^{(t)})\;\mathrm{d}u \\
&\approx \frac{\partial}{\partial\theta}\frac{1}{M}\cdot\sum_{m=1}^{M}\log p(y,u^{(t,m)} \mid \theta) \\
&= \frac{1}{M}\cdot\sum_{m=1}^{M}\frac{\partial}{\partial\theta}\log p(y,u^{(t,m)} \mid \theta)
\end{aligned}$$

Being able to compute these derivatives for any value of θ means that optimization algorithms based on gradients, such as gradient descent or limited-memory quasi-Newton methods like L-BFGS, may be used for the M-step.

Automatic differentiation

Automatic differentiation may be used to differentiate a model in cases where the derivatives are too time comsuming or error prone to code by hand. All that is needed is a program to evaluate $\log p(y \mid u,\theta)$ and the values of y, u, and θ at which derivatives are required.

Gradient-Based Marginal Optimization

Gradient-based marginal optimization (GMO) combines several ideas:

- Monte Carlo expectation maximization,
- Laplace approximations for the inner Monte Carlo step,
- importance sampling from Laplace approximation to compute the expectation,
- automatic differentiation for derivatives of expectation, and
- generalized expectation maximization.

Putting this all together, the *gradient-based marginal optimization* (GMO) algorithm is as follows.

1. Initialize $\theta^{(0)}$ such that $p(y \mid \theta_0) > 0$.

2. While the sequence has not converged,

   (i) simulate M draws from the Laplace approximation

   $$u^{(1)}, ..., u^{(M)} \sim \text{Normal}(\mu_u, \Sigma_u)$$

   with location vector

   $$\mu_u = \text{argmax}_u \; p\Big(u \mid y, \theta^{(t)}\Big)$$

   and the covariance matrix

   $$\Sigma_u \;=\; \left(\frac{\partial^2}{\partial u^2} \log p\Big(u \mid y, \theta^{(t)}\Big)\Big|_{u=\mu_u} \right)^{-1}$$

   (ii) approximate the gradient by

   $$\begin{aligned} \frac{\partial}{\partial \theta} Q\left(\theta \mid \theta^{(t)}\right) &\approx \frac{\partial}{\partial \theta} \hat{Q}_M\left(\theta \mid \theta^{(t)}\right) \\ &\approx \frac{1}{\sum_{m=1}^M \rho_m} \cdot \sum_{m=1}^M \rho_m \cdot \frac{\partial}{\partial \theta} \log p(y, u^{(m)} \mid \theta) \\ &= G\Big(\theta^{(t)}\Big) \end{aligned}$$

   with importance weights

   $$\rho_m = \frac{p(u^{(m)} \mid y, \theta)}{\text{Normal}(u^{(m)} \mid \mu_u, \Sigma_u)}$$

   (iii) step along the approximate gradient,

   $$\theta^{(t+1)} = \theta^{(t)} + \epsilon_t \cdot G\Big(\theta^{(t)}\Big)$$

   with stepsize schedule ϵ_t for $t = 1, 2, \ldots$.

3. Return the last $\theta^{(t)}$.

This algorithm sketch leaves open the stepsize schedule ϵ_t. Typically some kind of decreasing schedule is required for asymptotic convergence (in the number of iterations) to the correct value with stochastic gradient descent algorithms like these.

Appendix: Maximum Marginal a Posteriori Approximations

Bayesian models

A Bayesian model provides a joint density $p(y, u, \theta)$ of the observed data y, unobserved data u and parameters θ. In a Bayesian model, there is nothing special about the unobserved data—it is treated the same way as an unknown parameter.

Bayesian posteriors

The usual quantity of interest for a Bayesian model is the posterior distribution

$$p(\theta, u \mid y) \ = \ \frac{p(y \mid u, \theta) \cdot p(u, \theta)}{p(y)}$$

or one of the marginal posteriors $p(\theta \mid y)$ if only the parameters are of interest or $p(u \mid y)$ if only the missing data is of interest.

Fully Bayesian Posterior Predictive Inference

We usually don't care about estimates other than as a very rough indication of the location of the posterior. Posterior predictive inference requires integrating a function of the parameters weighted by posterior density.[18]

[18] For example, to compute event probabilities for an indicator function $\phi : \Theta \to \{0, 1\}$,

$$\begin{aligned} \Pr[\,\phi(\theta) \mid y\,] &= \mathbb{E}[\phi(\theta) \mid y] \\ &= \int_{\Theta} \phi(\theta) \cdot p(\theta \mid y) \, \mathrm{d}\theta \end{aligned}$$

With a Monte Carlo sample $\theta^{(1)}, \ldots, \theta^{(M)}$ from the posterior $p(\theta \mid y)$, the sample mean is the estimator,

$$\Pr[\phi(\theta) \mid y] \ \approx \ \frac{1}{M} \cdot \sum_{m=1}^{M} \phi\left(\theta^{(m)}\right).$$

Bayesian estimators

One form of Bayesian posterior predictive analysis is Bayesian parameter estimation. For example, the standard Bayesian point estimate for a parameter is the posterior mean, or expected value conditioned on the observed data,

$$\begin{aligned} \hat{\theta}, \hat{u} &= \mathbb{E}\,[\theta, u \mid y] \\ &= \int_{\Theta, U} (\theta, u) \cdot p(\theta, u \mid y) \, \mathrm{d}\theta \, \mathrm{d}u. \end{aligned}$$

The *error* in an estimate $\hat{\theta}$ is defined to be the difference from the true value, i.e., $\theta - \hat{\theta}$. The posterior mean estimates enjoy the pleasant property of being the estimates that minimize the expected squared error, or equivalently, mean square error.[19] The posterior median minimizes expected absolute error.[20]

[19] Squared error is defined between a parameter θ and its estimate $\hat{\theta}$, as $\left(\theta - \hat{\theta}\right)^2$.

[20] Absolute error is $\left|\theta - \hat{\theta}\right|$.

Maximum marginal a posteriori estimates

When it is too costly to calculate posterior means or medians, which usually requires Markov chain Monte Carlo (MCMC) methods, it is often possible to find the maximum marginal a posteriori (MMAP) estimates of the parameters θ as

$$\begin{aligned} \theta^* &= \mathrm{argmax}_{\theta} \ p(\theta \mid y) \\ &= \mathrm{argmax}_{\theta} \ \int_u p(u, \theta \mid y) \, \mathrm{d}u. \\ &= \mathrm{argmax}_{\theta} \ \mathbb{E}_{p(u \mid y, \theta)}[\, p(\theta, u \mid y)]\,. \end{aligned}$$

These estimates may then be used themselves, or as the basis of a Laplace approximation of the posterior.

Expectation Maximization for MMAP

The expectation maximization algorithm may be used to calculate the MMAP estimate in exactly the same way as it calculates the MML estimate. The only difference is that Q is defined in terms of the posterior instead of a likelihood.[21]

[21] If the prior is uniform, the two definitions are equivalent up to a constant.

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= \mathbb{E}_{p\left(u \mid y,\, \theta^{(t)}\right)}[\log p\left(\theta, u \mid y\right)] \\ &= \int_U p\left(u, \theta \mid y\right) \cdot p\left(u \mid y, \theta^{(t)}\right) \mathrm{d}u \end{aligned}$$

The result computes posterior modes.

Appendix: Laplace Approximation

Two concepts traffic under the heading "Laplace approximation", one to approximate a density and another to approximate an integral based on the approximate desnity.

Approximating a distribution

The Laplace approximation is a multivariate normal approximation to a density with location given by the density's mode and covariance by the inverse Hessian at the mode. That is, for a general distribution $p(\alpha)$, the Laplace approximation

$$p(\alpha) \approx \mathsf{MultiNormal}(\mu_\alpha, \Sigma_\alpha)$$

where

$$\mu_\alpha \;=\; \mathrm{argmax}_\alpha \; p(\alpha).$$

and

$$\Sigma_\alpha \;=\; \left(-\frac{\partial^2}{\partial \alpha^2} \log p(\alpha) \Big|_{\alpha = \mu_\alpha} \right)^{-1}$$

Approximating an expectation

The Laplace approximation is an approximation of a general expectation of a smooth function f and density p,[22]

[22] The smoothness required will be second order.

$$\mathbb{E}_{p(\theta)}[f(\theta)] = \int_\Theta f(\theta) \cdot p(\theta) \, \mathrm{d}\theta$$

where θ has D dimensions.

Let $\theta^\star$ be the point that maximizes the expression being integrated,[23]

[23] Thus $f(\theta) \cdot p(\theta)$ must have a maximum in order for this approximation to succeed.

$$\theta^\star = \mathrm{argmax}_\theta \; f(\theta) \cdot p(\theta).$$

Then the Laplace approximation is

$$\mathbb{E}_{p(\theta)}[f(\theta)] \;\approx\; f(\theta^\star) \cdot p(\theta^\star) \cdot \sqrt{\det\left(-\frac{\partial^2}{\partial \theta^2} \log\, (f(\theta) \cdot p(\theta)) \Big|_{\theta=\theta^\star} \right)}$$

The inner term is the determinant of the negative Hessian (matrix of second derivatives) of $f(\theta) \cdot p(\theta)$, evaluated at θ^*.

Appendix: Convergence of Expectation Maximization

The EM algorithm will converge if the iterations converge to the true value,

$$\lim_{t \to \infty} \theta^{(t)} \to \theta.$$

The proof of convergence hinges on each $\theta^{(t+1)}$ improving the approximation over the previous $\theta^{(t)}$, or more specifically by showing that the expected log complete data likelihood relative to $\theta^{(t)}$ improves in each iteration. Specifically, if $\theta^{(t)}$ hasn't already converged to the true value, i.e., if $\theta^{(t)} \neq \theta$, then

$$Q\Big(\theta \mid \theta^{(t+1)}\Big) \;>\; Q\Big(\theta \mid \theta^{(t)}\Big).$$

The rest of this section expands on why this condition is sufficient.

The marginal log likelihood for parameters θ may be defined on the log scale for observed data y by marginalizing the missing data u,

$$\begin{aligned} \log p(y \mid \theta) &= \log \frac{p(y, u \mid \theta)}{p(u \mid \theta)} \\ &= \log p(y, u \mid \theta) - \log p(u \mid \theta) \end{aligned}$$

During the E-step of the EM algorithm, we need to take the expectation of the log marginal likelihood with respect to the distribution of missing values governed by the current parameter values $\theta^{(t)}$,

$$\begin{aligned} \mathbb{E}_{p\left(u \mid y, \theta^{(t)}\right)} \left[\log p(y \mid \theta) \right] &= \mathbb{E}_{p\left(u \mid y, \theta^{(t)}\right)} \left[\log p(y, u \mid \theta) - \log p(u \mid \theta) \right] \\ &= \mathbb{E}_{p\left(u \mid y, \theta^{(t)}\right)} \left[\log p(y, u \mid \theta) \right] \;-\; \mathbb{E}_{p\left(u \mid y, \theta^{(t)}\right)} \left[\log p(u \mid \theta) \right] \\ &= Q\Big(\theta \mid \theta^{(t)}\Big) + \mathrm{H}\big[\, p_\theta \mid p_{\theta^{(t)}} \big], \end{aligned}$$

where the $Q(\cdots)$ term is as before and the $\mathrm{H}[\cdots]$ term is the cross-entropy between the missing data posterior with parameters θ and $\theta^{(t)}$,[24]

$$\begin{aligned} p_\theta(u) &= p(u \mid y, \theta) \\ p_{\theta^{(t)}}(u) &= p(u \mid y, \theta^{(t)}) \end{aligned}$$

[24] In general, cross-entropy is defined between from a density $p_1(u)$ to a density $p_2(u)$ as

$$\begin{aligned} \mathrm{H}\left[\, p_1(u) \mid p_2(u) \right] &= -\mathbb{E}_{p_1(u)} \left[\log p_2(u) \right] \\ &= -\textstyle\int_U \log p_2(u) \cdot p_1(u) \, \mathrm{d}u. \end{aligned}$$

We use "from" and "to" in the definition to distinguish argument postions because, in most cases of interest, $\mathrm{H}[p_1, p_2] \neq \mathrm{H}[p_2, p_1]$. Information theoretically, cross entropy is the expected cost in nats, which are like bits, only in (like bits, only in the natural logarithm base e, to encode a u drawn from distribution $p_1(u)$ using $p_2(u)$ as the basis of the code.

The left-hand side of the equation is constant inside the expectation, so the result reduces to

$$\log p(y \mid \theta) \;=\; Q(\theta \mid \theta^{(t)}) \;+\; \mathrm{H}[\, p_\theta \mid p_{\theta^{(t)}}]$$

Substituting in for the above,

$$\begin{aligned}
\log p(y \mid \theta) - \log p(y \mid \theta^{(t)}) \;&=\; \Big(Q(\theta \mid \theta^{(t)}) + \mathrm{H}[\, p_\theta \mid p_{\theta^{(t)}}]\Big) \\
&\quad - \Big(Q(\theta^{(t)} \mid \theta^{(t)}) + \mathrm{H}[\, p_{\theta^{(t)}} \mid p_{\theta^{(t)}}]\Big) \\
&=\; \Big(Q(\theta \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)})\Big) \\
&\quad + \; \big(\mathrm{H}[\, p_\theta \mid p_{\theta^{(t)}}] - \mathrm{H}[\, p_{\theta^{(t)}} \mid p_{\theta^{(t)}}]\big)
\end{aligned}$$

By Gibbs' inequality,[25]

$$\mathrm{H}[\, p_\theta \mid p_{\theta^{(t)}}] \geq \mathrm{H}[\, p_{\theta^{(t)}} \mid p_{\theta^{(t)}}]$$

That means $\mathrm{H}[\, p_\theta \mid p_{\theta^{(t)}}] - \mathrm{H}[\, p_{\theta^{(t)}} \mid p_{\theta^{(t)}}]$ is positive, and thus

$$\log p(y \mid \theta) - \log p(y \mid \theta^{(t)}) \;\geq\; Q(\theta \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}).$$

Substituting $\theta^{(t+1)}$ in for θ yields

$$\log p(y \mid \theta^{(t+1)}) - \log p(y \mid \theta^{(t)}) \;\geq\; Q(\theta^{(t+1)} \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}).$$

Finally, because

$$\theta^{(t+1)} = \mathrm{argmax}_\theta \; Q(\theta \mid \theta^{(t)}),$$

it follows that the right-hand side is non-negative, and thus

$$\log p(y \mid \theta^{(t+1)}) \geq \log p(y \mid \theta^{(t)}).$$

[25] Gibbs' inequality states that for distributions p_1 and p_2,

$$\mathrm{H}[p_1 \mid p_2] \geq \mathrm{H}[p_2 \mid p_2]\,.$$

It follows that cross entropy from a fixed distribution p_1 to another distribution p_2 is minimized at $p_2 = p_1$.

Appendix: Analytic Expectation Maximization

In the traditional applications of EM to mixture models, hidden Markov models, or missing data problems with sufficient statistics, both the marginal expectation calculations and maximization would be computed analytically. This is particularly straightforward when the component distributions are drawn from exponential families and may be optimized as functions of sufficient statistics rather than a large data set.

For example, consider a mixture of two normals, with unknown locations μ_0, μ_1 and scales σ_0, σ_1 for the components and mixing proportion λ for the proportion of items drawn from component 1. The model is thus[26]

$$\begin{aligned}
z_n \;&\sim\; \mathsf{Bernoulli}(\lambda) \\
y_n \;&\sim\; \mathsf{Normal}(\mu_{z[n]}, \sigma_{z[n]})
\end{aligned}$$

[26] We write $z[n]$ instead of z_n to avoid the small size of doubley nested subscripts.

This yields a log density for a given pair of observed data y_n and mixture responsibility of z_n for parameters $\theta = (\lambda, \mu, \sigma)$ of

$$\log p(y_n, z_n \mid \theta) = \log \mathsf{Bernoulli}(z_n \mid \lambda) + \log \mathsf{Normal}(y_n \mid \mu_{z[n]}, \sigma_{z[n]}).$$

Given values of the parameters $\theta^{(t)} = (\lambda^{(t)}, \mu^{(t)}, \sigma^{(t)})$, it is just a matter of algebra to derive the required expectations for each $n \in 1:N$. Rather than an integral, there is a summation over the discrete domain $Z = \{0, 1\}$ of the z_n. Expanding out the definitions yields

$$\mathbb{E}_{p\left(z_n \mid y_n, \theta^{(t)}\right)}[\log p\left(y_n,\, z_n \mid \theta\right)] \;=\; \sum_{z_n \in Z} p\left(z_n \mid y_n, \theta^{(t)}\right) \cdot \log p\left(z_n, y_n \mid \theta\right).$$

Next, the conditional distribution of the missing z_n given $\theta^{(t)}$ and y_n may be addressed by Bayes's rule,

$$\begin{aligned} p\left(z_n \mid y_n, \theta^{(t)}\right) &\propto p\left(z_n, y_n \mid \theta^{(t)}\right) \\ &= \mathsf{Bernoulli}\left(z_n \mid \lambda^{(t)}\right) \cdot \mathsf{Normal}\left(y_n \mid \mu_{z_n}^{(t)}, \sigma_{z_n}^{(t)}\right) \end{aligned}$$

Discrete distributions can be normalized by dividing over the sum of the proportional densities of the possible values for $z_n \in Z = \{0, 1\}$,

$$\begin{aligned} p\left(z_n \mid y_n, \theta^{(t)}\right) &= \frac{p\left(z_n \mid y_n, \theta^{(t)}\right)}{\sum_{z \in Z} p\left(z \mid y_n, \theta^{(t)}\right)} \\ &= \frac{\mathsf{Bernoulli}\left(z_n \mid \lambda^{(t)}\right) \cdot \mathsf{Normal}\left(y_n \mid \mu_{z_n}^{(t)}, \sigma_{z_n}^{(t)}\right)}{\mathsf{Bernoulli}\left(0 \mid \lambda^{(t)}\right) \cdot \mathsf{Normal}\left(y_n \mid \mu_0^{(t)}, \sigma_0^{(t)}\right) \;+\; \mathsf{Bernoulli}\left(1 \mid \lambda^{(t)}\right) \cdot \mathsf{Normal}\left(y_n \mid \mu_1^{(t)}, \sigma_1^{(t)}\right)} \end{aligned}$$

This needs to be summed over all $n \in 1:N$ and then plugged back into the expression for the expected complete data likelihood under the previous parameters, $Q(\theta \mid \theta^{(t)})$.

In this particular case, it is also easy to do the estimation step. Because the responsibilities are known in expectation, these may be used as weights on the observations y_n to calculate the maximum likelihood estimates for μ_0, μ_1 and σ_0, σ_1. These are calculated analytically from the weighted sufficient statistics. For example, μ_1^* is just the average of y weighted by the expected responsibility,

$$\mu_1^* \;=\; \left(\frac{1}{\sum_{n=1}^N \mathbb{E}[z_n \mid y]}\right) \cdot \sum_{n=1}^N y_n \cdot \mathbb{E}[z_n \mid y].$$

The estimate μ_0^* is defined similarly, with flipped weighting

$$\mu_1^* \;=\; \left(\frac{1}{\sum_{n=1}^N 1 - \mathbb{E}[z_n \mid y]}\right) \cdot \sum_{n=1}^N y_n \cdot (1 - \mathbb{E}[z_n \mid y])\,.$$

Given the estimates μ_0^* the scale estimates are just the square root of the sum of squares weighted again by expectation of responsibility,

$$\sigma_0^* = \sqrt{\left(\frac{1}{\sum_{n=1}^N \mathbb{E}[z_n \mid y]}\right) \cdot \sum_{n=1}^N (y_n - \mu_0^*)^2 \;\cdot\; \mathbb{E}[z_n \mid y]}$$

The calculation for σ_1 is similar.

The maximum likelihood estimates σ_0^* and σ_*^1 are biased[27] because they systematically underestimate true variation, i.e.,

$$\mathbb{E}_{p(y)}[\,\theta^*\,] < \theta.$$

The underestimate arises because the weighted sum of squared differences $(y_n - \mu_0^*)^2$ is based on the estimate μ_0^* of μ_0 rather than the true μ_0. This can be adjusted by slightly modifying the denominator to subtract 1 from the weighted, total, changing the leading multiplier to

$$\frac{1}{1 - \sum_{n=1}^{N} \mathbb{E}[z_n \mid y]}.$$

[27] An estimator θ^* is said to be *biased* if its expected value is not the true value, i.e., if $\mathbb{E}[\theta^*] \neq \theta$, where the expectation is taken over the distribution $p(y)$ on which the estimator is based. This would be clearer if the estimate was written treating the estimator as a function, e.g., $\mathbb{E}_{p(y)}[\theta^*(y)]$

Historical Notes

Depmster, Laird, and Rubin (1977) introduced the expectation maximization algorithm and proved that the algorithm converged to the maximum marginal likelihood estimate.

Wei and Tanner (1990) introduced the Monte Carlo expectation maximization (MC-EM) algorithm for both maximum marginal likelihood and maximum marginal a posteriorio estimation and established convergence. They also discussed approximate integrals and Monte Carlo gradient calculations.

Neal and Hinton (1998) introduced several generalized forms of the EM algorithm.

Deylon, Lavielle, and Moulines (1999) introduced the stochastic averaging expectation maximization (SA-EM) algorithm and established convergence.

Kucukelbir, Tran, Ranganath, Gelman, and Blei (2017) introduced gradient-based maximization of expectations using automatic differentiation, with applications to variational inference.

Tran, Gelman, and Vehtari (2016) introduced gradient-based marginal optimization. The stochastic gradient descent approach they employ was developed by Robbins and Monro (1951).

Rue, Martino, and Chopin (2009) introduced Laplace approximations for the marginal posterior distributions of parameters (which they then integrate by quadrature) and the marginal posterior distribution of missing data conditioned on the parameters.

References

- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, 1–38.

- Delyon, B., Lavielle, M., & Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, 94–128.
- Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., & Blei, D. M. (2017). Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1), 430-474.
- Neal, R. M., & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M. E. (Ed.) *Learning in Graphical Models* (pp. 355-368), MIT Press.
- Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 400-407.
- Rue, H., Martino, S., & Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, 71(2), 319-392.
- Tran, D., Gelman, A., & Vehtari, A. (2016) Gradient-based marginal optimization. Unpublished manuscript.
- Wei, G. C., & Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411), 699–704.