Week 8: Oct 17 - Oct 21

Estimation vs. Approximation

There are a few key elements of a Bayesian data analysis:

1. *Model Specification:* a sampling model $p(y_1, \ldots, y_n|\phi)$ is specified. Where ϕ could be a high dimensional parameter set.

2. *Prior Specification:* a probability distribution $p(\phi)$ is specified.
   Once these are specified and the data have been gathered, the posterior distribution $p(\phi|y_1, \ldots, y_n)$ is fully determined. It is exactly:
   $$p(\phi|y_1, \ldots, y_n) = \frac{p(\phi)p(y_1, \ldots, y_n|\phi)}{p(y_1, \ldots, y_n)}$$
   Excluding posterior model checking, all that remains is to summarize the posterior.

3. *Posterior Summary:* A description of the posterior distribution $p(\phi|y_1, \ldots, y_n)$, typically using intervals, posterior means, and predictive probabilities.

For most models we have discussed thus far, $p(\phi|y_1, \ldots, y_n)$ is known in closed form or easy to sample from using Monte Carlo procedures. However, in more sophisticated settings, $p(\phi|y_1, \ldots, y_n)$ is complicated, and hard to write down or sample from. In these cases, we study $p(\phi|y_1, \ldots, y_n)$ by looking at MCMC samples. Thus, Monte Carlo and MCMC sampling algorithms:

-
-
-

For example, if we have Monte Carlo samples $\phi^{(1)}, \ldots, \phi^{(J)}$ that are approximate draws from $p(\theta|y_1, dots, y_n)$, then these sample help describe $p(\phi|y_1, \ldots, y_n)$, for example:

-

•

To keep the distinction between and clear, commonly *estimation* is used do describe how we use $p(\phi|y_1, \ldots, y_n)$ to make inferences about ϕ and use *approximation* to describe the use of Monte Carlo (including MCMC) procedures to approximate integrals.

MCMC Diagnostics

A useful way to think about an MCMC sampler is that there is a *particle* moving through and exploring the parameter space. For each region, or set, A the particle needs to spend time proportional to the target probability, $\int_A p(\phi)d\phi$ Consider the three modes from the exercise conducted in class and denote these three modes as A_1, A_2, A_3. Given the weights on the mixture components the particle should spend more time in A_1 and A_3 than A_2. However, if the particle was initialized in A_2 we'd hope that the number of iterations are large enough that:

1.

2.

The technical term associated with item 1 is which means the chain has conveged to the target distribution. For the models we have seen thus far, convergence happens quite rapidly, but we will look at this in more depth later on. The second item is focused on the speed the particle moves through the target distribution, this is referred to as . An independent sampler like the Monte Carlo procedures we have seen have perfect mixing as each sample is independently drawn from the target distribution. The MCMC samples can be highly correlated and tend to get stuck in certain regions of the space.

People often quantify mixing properties of MCMC samples using the idea of effective sample size. To understand this, first consider the variance of independent Monte Carlo samples:

$$Var_{MC}[\bar{\phi}] =$$

where $\bar{\phi} = \sum_{j=1}^{J} \phi^{(j)}/J$. The Monte Carlo variance is controlled by the number of samples obtained from

the algorithm. In a MCMC setting, consecutive samples $\phi^{(j)}$ and $\phi^{(j+1)}$ are not independent, rather they are usually positively correlated. Once stationarity has been acheived, the variance of the MCMC algorithm can be expressed as:

$$Var_{MCMC}[\phi] = \cdots = Var_{MC}[\bar{\phi}] +$$

where ϕ_0 is the true value of the integral, typically $E[\phi]$. Now if two consecutive samples are highly correlated the variance of the estimator will be much larger than that of an Monte Carlo procedure with the same number of iterations. This is captured in the idea of the **effective sample**. The effective sample size is computed such that:

$$Var_{MCMC}[\bar{\phi}] =$$

where S_{eff} can be interpreted as the number of independent Monte Carlo samples necessary to give the same precision as the MCMC samples. Note that the R function `effectiveSize` in the "coda" package will calculate the effective sample size of MCMC output.

We will talk more about MCMC diagnostics after introducing the Metropolis-Hastings algorithm later in class, but the general procedure is:

1.

2.

An easy solution, especially in the context of Gibbs Sampling is to look at trace plots and histograms of marginal posterior distributions. In conjunction with ESS (Effective Sample Size) calculations this usually gives a good sense of convergence. In other situations, combining visuals displays (trace plots) with other statisics Gelman's R statistic or QDE is a good strategy.

The big picture idea with MCMC, is that we want to guarantee that our algorithm has:

1.

2.

Posterior Model Checks on Normal Model

Exercise. Consider a similar scenario to the code for the first Gibbs sampler. Again 100 data points have been generated.

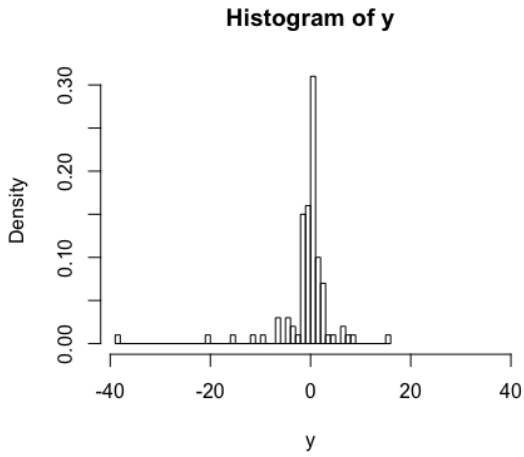
1. A histogram of the data is shown later as figure (a). What are your thoughts about this data?

2. Now assume you used your MCMC code and came up with figures (b) - (e). Comment on the convergence and the marginal posterior distributions.

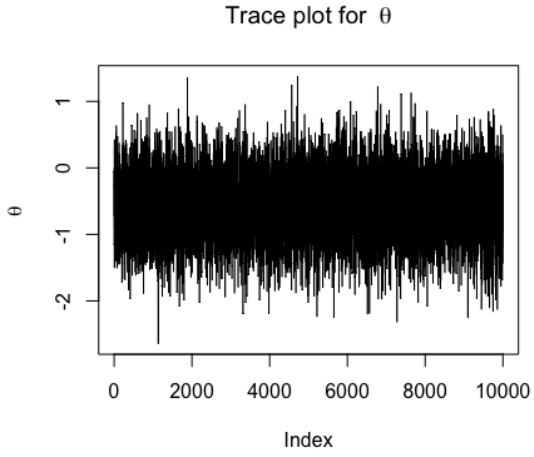
3. As a final check you decide to use your MCMC samples to compute the posterior predictive distribution, $p(y^*|y_1, \ldots, y_n)$. Computationally this can be acheived by using each pair $\{\theta^{(i)}, \sigma^{2(i)}\}$ and then simulating from $N(y^*; \theta^{(i)}, \sigma^{2(i)})$. In R this can be done with one line of code:

   ```
   post.pred <- rnorm(num.sims,mean=Phi[,1],sd = sqrt(1/Phi[,2]))
   ```

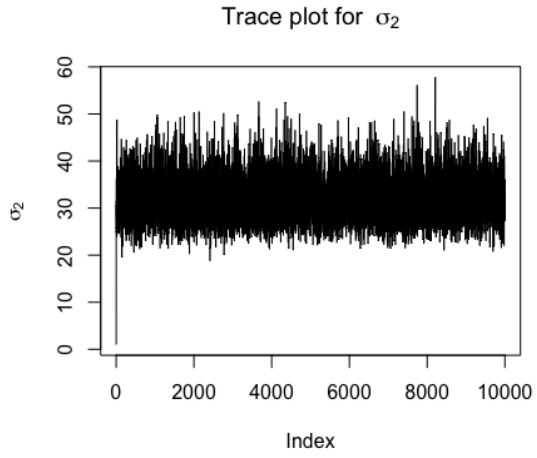
   Now compare your posterior predictive distribution with the observed data. Are you satisfied that the posterior predictive distribution represents the actual data? Why or why not?



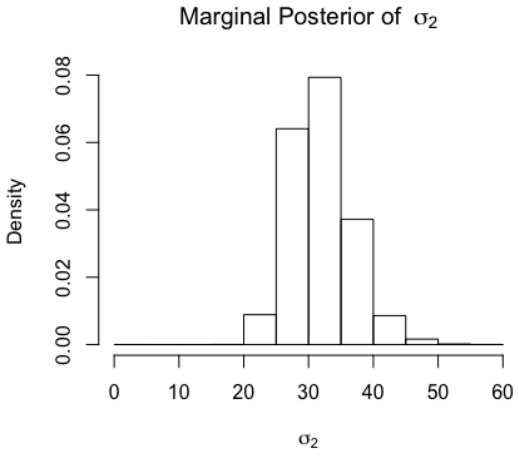
(a) Histogram of the Data



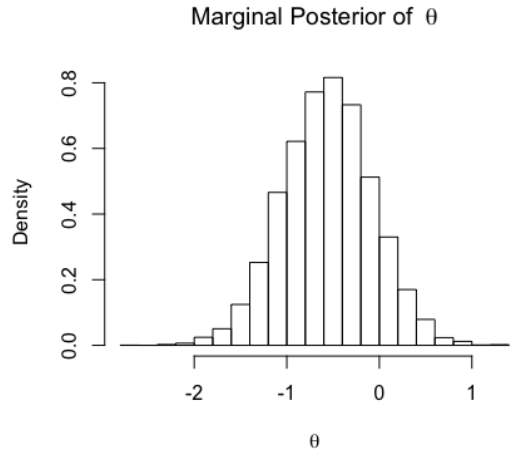
(b) Trace plot for θ

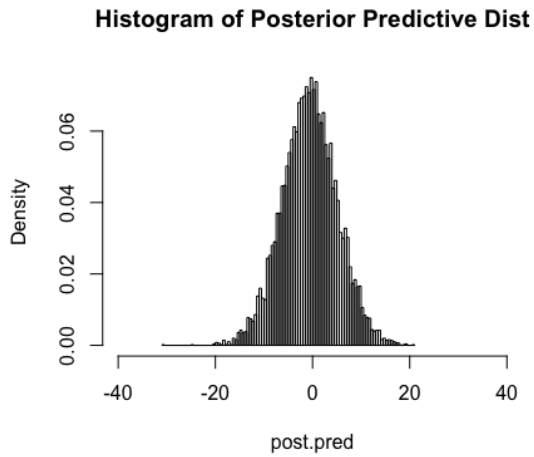


(c) Trace plot for σ^2



(d) Histogram of marginal posterior for σ^2





(e) Histogram of marginal posterior for θ

Multivariate Normal Distribution

While the normal distribution has two parameters, up to now we have focused on univariate data. Now we will consider multivariate responses from a multivariate normal distribution.

Example.

Multivariate Normal Distribution: The multivariate normal distribution has the following sampling distribution:

$$p(\tilde{y}|\tilde{\theta}, \Sigma) = \qquad ,$$

where

$$\tilde{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}, \qquad \tilde{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{1,2} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1,p} & \sigma_{2,p} & \cdots & \sigma_p^2 \end{pmatrix}$$

The vector $\tilde{\theta}$ is the mean vector and the matrix Σ is the covariance matrix, where the diagonal elements are the variance terms for observation i and the off diagonal elements are the covariance terms between observation i and j. Marginally, each $y_i \sim$

Linear Algebra Review

- Let A be a matrix, then $|A|$ is the

- Let A be a matrix, then A^{-1} is the

- Let $\tilde{\theta}$ be a vector of dimension $p \times 1$, $\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}$, then the transpose of θ, denoted θ^T is

- Let A be a $n \times p$ matrix and B be a $p \times m$ matrix, then the matrix product, AB will be

- Let A be a matrix, then the **trace** of A is

- The `mnormt` package in R includes functions `rmnorm, dmnorm` which are multivariate analogs of functions used for a univariate normal distribution.

Priors for multivariate normal distribution

In the univariate normal setting, a normal prior for the mean term was *semiconjugate*. Does the same hold for a multivariate setting? Let $p(\tilde{\theta}) \sim N_p(\tilde{\mu}_0, \Lambda_0)$.

$$\begin{aligned} p(\tilde{\theta}) &= (2\pi)^{-p/2}|\Lambda_0|^{-1/2} \exp\left[-\frac{1}{2}(\tilde{\theta} - \tilde{\mu}_0)^T \Lambda_0^{-1}(\tilde{\theta} - \tilde{\mu_0})\right] \\ &\propto \exp\left[-\frac{1}{2}(\tilde{\theta} - \tilde{\mu}_0)^T \Lambda_0^{-1}(\tilde{\theta} - \tilde{\mu_0})\right] \\ &\propto \exp \end{aligned}$$

Now combine this with the sampling model, only retaining the elements that contain θ.

$$\begin{aligned} p(\tilde{y}_1, \ldots, \tilde{y}_n | \tilde{\theta}, \Sigma) &\propto \prod_{i=1}^{n} \exp\left[-\frac{1}{2}(\tilde{y}_i - \tilde{\theta})^T \Sigma^{-1} (\tilde{y}_i - \tilde{\theta})\right] \\ &\propto \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(\tilde{y}_i - \tilde{\theta})^T \Sigma^{-1} (\tilde{y}_i - \tilde{\theta})\right] \\ &\propto \exp \end{aligned}$$

Next we find the full conditional distribution for θ, $p(\tilde{\theta}|\Sigma, \tilde{y}_1, \ldots, \tilde{y}_n)$.

$$\begin{aligned} p(\tilde{\theta}|\Sigma, \tilde{y}_1, \ldots, \tilde{y}_n) &\propto \exp\left[-\frac{1}{2}\left(\tilde{\theta}^T n\Sigma^{-1}\tilde{\theta} - \tilde{\theta}^T\Sigma^{-1}\sum_{i=1}^{n}\tilde{y}_i - \sum_{i=1}^{n}\tilde{y}_i^T\Sigma^{-1}\tilde{\theta} + \tilde{\theta}^T\Lambda_0^{-1}\tilde{\theta} - \tilde{\theta}^T\Lambda_0^{-1}\tilde{\mu_0} - \tilde{\mu_0}^T\Lambda_0^{-1}\tilde{\theta}\right)\right] \\ &\propto \exp\left[-\frac{1}{2}\left(\tilde{\theta}^T(n\Sigma^{-1} + \Lambda_0^{-1})\tilde{\theta} - \tilde{\theta}^T(\Sigma^{-1}\sum_{i=1}^{n}\tilde{y}_i + \Lambda_0^{-1}\tilde{\mu}_0) - c\tilde{\theta}\right)\right] \end{aligned}$$

it turns out we can drop the term $c\tilde{\theta}$

$$\propto \exp$$

and we have a similar result to that found earlier for a univariate normal

The variance (matrix) is A^{-1} and the expectation is $A^{-1}B$. Hence the full conditional follows a multivariate normal distribution with variance $\Lambda_n =$ and expectation $= \tilde{\mu}_n =$. Sometimes a uniform prior $p(\tilde{\theta}) \propto \tilde{1}$ is used. In this case the variance and expectation simplify to $V = \Sigma/n$ and $E = \bar{\tilde{y}}$.

Using this semiconjugate prior in a Gibbs sampler we can make draws from the full conditional distribution using `rmnorm(.)` in R. However, we still need to be able to take samples of the covariance matrix Σ to get draws from the joint posterior distribution.

Inverse-Wishart Distribution

A covariance matrix $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{1,2} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1,p} & \sigma_{2,p} & \cdots & \sigma_p^2 \end{pmatrix}$ has the variance terms on the diagonal and covariance terms for off diagonal elements. Similar to the requirement that σ^2 be positive, a covariance matrix, Σ must be

positive definite, such that: $\tilde{x}^T \Sigma \tilde{x}$ for all vectors $\tilde{x}$. With a positive definite matrix, the diagonal elements (which correspond the marginal variances σ_j^2) are greater than zero and it also constrains the correlation terms to be between -1 and 1. A covariance matrix also requires symmetry, so that $Cov(y_i, y_j) = Cov(y_j, y_i)$.

The covariance matrix is closely related to the sum of squares matrix with is given by:

$$\sum_{i=1}^{N} \tilde{z}_i \tilde{z}_i^T = Z^T Z,$$

where $z_1, ..., z_n$ are $p \times 1$ vectors containing the multivariate response. Thus $\tilde{z}_i \tilde{z}_i^T$ results in a $p \times p$ matrix, where

$$\tilde{z}_i \tilde{z}_i^T = \begin{pmatrix} z_{i,1}^2 & z_{i,1} z_{i,2} & \dots & z_{i,1} z_{i,p} \\ z_{i,2} z_{i,1} & z_{i,2}^2 & \dots & z_{i,2} z_{i,p} \\ \vdots & & & \vdots \\ z_{i,p} z_{i,1} & z_{i,p} z_{i,2} & \dots & z_{i,p}^2 \end{pmatrix}$$

Now let the $\tilde{z}_i' s$ have zero mean (are centered). Recall that the sample variance is computed as $S^2 =$. Similarly the matrix $\tilde{z}_i \tilde{z}_i^T / n$ is the contribution of the i^{th} observation to the sample covariance. In this case:

- $\frac{1}{n}[Z^T Z]_{j,j}$

- $\frac{1}{n}[Z^T Z]_{j,k} =$

If $n > p$ and the $\tilde{z}_i' s$ are linearly independent then $Z^T Z$ will be positive definite and symmetric.

Consider the following procedure with a positive integer, ν_0, and a $p \times p$ covariance matrix Φ_0:

1. sample

2. calculate

then the matrix $Z^T Z$ is a random draw from a with parameters ν_0 and Φ_0. The expectation of $Z^T Z$ is $\nu_0 \Phi_0$. The Wishart distribution can be thought of as a multivariate analogue of the gamma distribution. Accordingly,

The density of the inverse-Wishart distribution with parameters S_0^{-1}, a $p \times p$ matrix and ν_0: $IW(\nu_0, S_0^{-1})$ is:

$$p(\Sigma) = \left[2^{\nu_0 p/2} \pi^{\binom{p}{n}/2} |S_0|^{-\nu_0/2} \prod_{j=1}^{P} \Gamma([\nu_0 + 1 - j]/2) \right]^{-1} \times$$

Inverse Wishart Full Conditional Calculations

$$\begin{aligned} p(\Sigma|\tilde{y}_1, \ldots, \tilde{y}_n, \tilde{\theta}) &\propto p(\Sigma) \times p(\tilde{y}_1, \ldots, \tilde{y}_n|\Sigma, \tilde{\theta}) \\ &\propto \left(|\Sigma|^{-(\nu_0+p+1)/2} \exp[-tr(S_0\Sigma^{-1})/2] \right) \times \left(|\Sigma|^{-n/2} \exp[-\frac{1}{2}\sum_{i=1}^{n} (\tilde{y}_i - \tilde{\theta})^T \Sigma^{-1} (\tilde{y}_i - \tilde{\theta})] \right) \end{aligned}$$

$$\begin{aligned} \text{so } p(\Sigma|\tilde{y}_1, \ldots, \tilde{y}_n, \tilde{\theta}) &\propto \left(|\Sigma|^{-(\nu_0+p+1)/2} \exp[-tr(S_0\Sigma^{-1})/2] \right) \times \\ &\propto \left(|\Sigma|^{-(\nu_0+n+p+1)/2} \exp[-tr([S_0 + S_\theta]\Sigma^{-1})/2] \right) \\ \text{thus } \Sigma|\tilde{y}_1, \ldots, \tilde{y}_n, \tilde{\theta} &\sim IW(\end{aligned}$$

Thinking about the parameters in the prior distribution, is the prior sample size and is the prior residual sum of squares.

Gibbs Sampling for Σ and $\tilde{\theta}$

We now that the full conditional distributions follow as:

$$\begin{aligned} \tilde{\theta}|\Sigma, \tilde{y}_1, \ldots, \tilde{y}_n &\sim \\ \Sigma|\tilde{\theta}, \tilde{y}_1, \ldots, \tilde{y}_n &\sim \end{aligned}$$

Given these full conditional distributions the Gibbs sampler can be implemented as:

1. Sample $\tilde{\theta}^{(j+1)}$ from the full conditional distribution

   (a)

   (b)

2. Sample $\Sigma^{(j+1)}$ from its full conditional distribution

   (a)

   (b)

As $\tilde{\mu}_n$ and Λ_n depend on Σ they must be calculated every iteration. Similarly, S_n depends on $\tilde{\theta}$ and needs to be calculated every iteration as well.