Chapter 3.2

Markov chain Monte Carlo

Monte Carlo sampling

- Monte Carlo (MC) sampling is the predominant method of Bayesian inference because it can be used for high-dimensional models (i.e., with many parameters)
- The main idea is to approximate posterior summaries by drawing samples from the posterior distribution, and then using these samples to approximate posterior summaries of interest
- This requires drawing samples from non-standard distributions
- It also requires careful analysis to be sure the approximation is sufficiently accurate

Monte Carlo sampling

- Notation: Let $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$ be the collection of all parameters in the model
- Notation: Let $\mathbf{Y} = (Y_1, ..., Y_n)$ be the entire dataset
- The posterior $f(\boldsymbol{\theta}|\mathbf{Y})$ is a distribution
- If $\theta^{(1)}, ..., \theta^{(S)}$ are samples from $f(\boldsymbol{\theta}|\mathbf{Y})$, then the mean of the S samples approximates the posterior mean
- This only provides approximations of the posterior summaries of interest.
- But how to draw samples from some arbitrary distribution $p(\boldsymbol{\theta}|\mathbf{Y})$?

Software optioms

- There are now many software options for performing MC sampling
- There are SAS procs and R functions for particular analyses (e.g., the function `BLR` for linear regression)
- There are also all-purpose programs that work for virtually any user-specified model: OpenBUGS; JAGS; Proc MCMC; STAN; INLA (not MC)
- We will use JAGS, but they are all similar

MCMC

We will study the algorithms behind these programs, which is important because it helps:

- Select models and priors conducive to MC sampling
- Anticipate bottlenecks
- Understand error messages and output
- Design your own sampler if these off-the-shelf programs are too slow

The most common algorithms are **Gibbs** and **Metropolis** sampling

Gibbs sampling

- Gibbs sampling is attractive because it can sample from high-dimensional posteriors
- The main idea is to break the problem of sampling from the high-dimensional joint distribution into a series of samples from low-dimensional conditional distributions
- Updates can also be done in blocks (groups of parameters)
- Because the low-dimensional updates are done in a loop, samples are not independent
- The dependence turns out to be a Markov distribution, leading to the name Markov chain Monte Carlo (MCMC)

MCMC for the Bayesian t test

- Say $Y_i \sim \text{Normal}(\mu, \sigma^2)$ with $\mu \sim \text{Normal}(0, \sigma_0^2)$ and $\sigma^2 \sim \text{InvGamma}(a, b)$

- In Chapter 2 we saw that if we knew either μ or σ^2, we can sample from the other parameter

- $\mu|\sigma^2, \mathbf{Y} \sim \text{Normal}\left[\frac{n\bar{Y}\sigma^{-2}+\mu_0\sigma_0^{-2}}{n\sigma^{-2}+\sigma_0^{-2}}, \frac{1}{n\sigma^{-2}+\sigma_0^{-2}}\right]$

- $\sigma^2|\mu, \mathbf{Y} \sim \text{InvGamma}\left[\frac{n}{2}+a, \frac{1}{2}\sum_{i-1}^{n}(Y_i-\mu)^2+b\right]$

- But how to draw from the joint distribution?

Gibbs sampling for the Gaussian model

- The full conditional (FC) distribution is the distribution of one parameter taking all other as fixed and known

- FC1: $\mu|\sigma^2, \mathbf{Y} \sim \text{Normal}\left[\frac{n\bar{Y}\sigma^{-2}+\mu_0\sigma_0^{-2}}{n\sigma^{-2}+\sigma_0^{-2}}, \frac{1}{n\sigma^{-2}+\sigma_0^{-2}}\right]$

- FC2: $\sigma^2|\mu, \mathbf{Y} \sim \text{InvGamma}\left[\frac{n}{2}+a, \frac{1}{2}\sum_{i-1}^{n}(Y_i-\mu)^2+b\right]$

Gibbs sampling

- In the Gaussian model $\boldsymbol{\theta} = (\mu, \sigma^2)$ so $\theta_1 = \mu$ and $\theta_2 = \sigma^2$
- The algorithm begins by setting initial values for all parameters, $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, ..., \theta_p^{(0)})$.
- Variables are then sampled one at a time from their full conditional distributions,

$$p(\theta_j | \theta_1, ..., \theta_{j-1}, \theta_{j+1}, ..., \theta_p, \mathbf{Y})$$

- Rather than 1 p-dimensional joint sample, we make p 1-dimensional samples.
- The process is repeated until the required number of samples have been generated.

Gibbs sampling

A Set initial value $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, ..., \theta_p^{(0)})$

B For iteration t,

FC1 Draw $\theta_1^{(t)} | \theta_2^{(t-1)}, ..., \theta_p^{(t-1)}, \mathbf{Y}$

FC2 Draw $\theta_2^{(t)} | \theta_1^{(t)}, \theta_3^{(t-1)}, ..., \theta_p^{(t-1)}, \mathbf{Y}$

...

FCp Draw $\theta_p^{(t)} | \theta_1^{(t)}, ..., \theta_{p-1}^{(t)}, \mathbf{Y}$

We repeat step B S times giving posterior draws

$$\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(S)}$$

Why does this work?

- $\theta^{(0)}$ isn't a sample from the posterior, it is an arbitrarily chosen initial value
- $\theta^{(1)}$ likely isn't from the posterior either. Its distribution depends on $\theta^{(0)}$
- $\theta^{(2)}$ likely isn't from the posterior either. Its distribution depends on $\theta^{(0)}$ and $\theta^{(1)}$
- **Theorem**: For any initial values, the chain will eventually converge to the posterior
- **Theorem**: If $\theta^{(s)}$ is a sample from the posterior, then $\theta^{(s+1)}$ is too

Convergence

- We need to decide:
  1. When has it converged?
  2. When have we taken enough samples to approximate the posterior?
- Once we decide the chain has converged at iteration T, we discard the first T samples as "burn-in"
- We use the remaining $S - T$ to approximate the posterior
- For example, the posterior mean (marginal over all other parameters) of θ_j is

$$E(\theta_j|\mathbf{Y}) \approx \frac{1}{S-T} \sum_{s=S-T+1}^{S} \theta_j^{(s)}$$

Practice problem

- Implementing Gibbs sampling requires deriving the full conditional distribution of each parameter

- Work out the full conditionals for λ and b for the following model:

$$Y|\lambda, b \sim \text{Poisson}(\lambda)$$
$$\lambda|b \sim \text{Gamma}(1, b)$$
$$b \sim \text{Gamma}(1, 1)$$

Practice problem

$Y|\lambda, b \sim \text{Poisson}(\lambda)$, $\lambda|b \sim \text{Gamma}(1, b)$, $b \sim \text{Gamma}(1, 1)$

- The full conditional for λ is

$$\begin{aligned}
p(\lambda|b, Y) &\propto \frac{f(Y, \lambda, b)}{f(Y, b)} \propto f(Y, \lambda, b) \\
&\propto f(Y|\lambda, b)\pi(\lambda|b)\pi(b) \\
&\propto f(Y|\lambda)\pi(\lambda|b) \\
&\propto \left[\exp(-\lambda)\lambda^{Y}\right]\left[\exp(-b\lambda)\lambda^{1-1}\right] \\
&\propto \exp[-(b+1)\lambda]\lambda^{(Y+1-1)}
\end{aligned}$$

- Therefore, $\lambda|b, Y \sim \text{Gamma}(Y + 1, b + 1)$

Practice problem

$Y|\lambda, b \sim \text{Poisson}(\lambda)$, $\lambda|b \sim \text{Gamma}(1, b)$, $b \sim \text{Gamma}(1, 1)$

- The full conditional for b is

$$
\begin{aligned}
p(\lambda|b, Y) &\propto \frac{f(Y, \lambda, b)}{f(Y, \lambda)} \propto f(Y, \lambda, b) \\
&\propto f(Y|\lambda)\pi(\lambda|b)\pi(b) \\
&\propto \pi(\lambda|b)\pi(b) \\
&\propto \left[b^1 \exp(-b\lambda)\right]\left[\exp(-b)b^{1-1}\right] \\
&\propto \exp[-(\lambda+1)b]b^{(2-1)}
\end{aligned}
$$

- Therefore, $b|\lambda, Y \sim \text{Gamma}(2, \lambda + 1)$

Examples

- `http://www4.stat.ncsu.edu/~reich/ABA/code/NN2`
- `http://www4.stat.ncsu.edu/~reich/ABA/code/SLR`
- `http://www4.stat.ncsu.edu/~reich/ABA/code/ttest`
- All derivations of full conditionals are in the online derivations

Metropolis sampling

- In Gibbs sampling each parameter is updated by sampling from its full conditional distribution
- This is possible with conjugate priors
- However, if the prior is not conjugate it is not obvious how to make a draw from the full conditional
- For example, if $Y \sim \text{Normal}(\mu, 1)$ and $\mu \sim \text{Beta}(a, b)$ then

$$p(\mu|Y) \propto \exp\left[-\frac{1}{2}(Y-\mu)^2\right]\mu^{(a-1)}(1-\mu)^{b-1}$$

- For some likelihoods there is no known conjugate prior, e.g., logistic regression
- In these cases we use Metropolis sampling

Metropolis sampling

- Metropolis sampling is a version of rejection sampling
- Let θ_j^* be the current value of the parameter being updated and $\theta_{(j)}$ be the current value of all other parameters
- You propose a random candidate based on the current value, e.g.,

$$\theta_j^c \sim \text{Normal}(\theta_j^*, s_j^2)$$

- The candidate is accepted with probability

$$R = \min\left\{1, \frac{p(\theta_j^c|\theta_{(j)}, \mathbf{Y})}{p(\theta_j^*|\theta_{(j)}, \mathbf{Y})}\right\}$$

- If the candidate is not accepted then you simply retain the previous value and move to the next step

Metropolis sampling

- The candidate standard deviation s_j is a tuning parameter
- Ideally s_j is tuned to give acceptance probability around 0.3-0.4
- If s_j is too small:
- If s_j is too large:
- Off-the-shelf programs have default values, and many allow you to change the value if the results are unsatisfactory

Metropolis-Hastings sampling

- Denote $\theta_j^c \sim q(\theta|\theta^*)$ as the candidate distribution

- The candidate distribution is symmetric if

$$q(\theta^*|\theta_j^c) = q(\theta_j^c|\theta^*)$$

- For example, if $\theta_j^c \sim \text{Normal}(\theta_j^*, s_j^2)$ then

$$q(\theta_j^c|\theta^*) = \frac{1}{\sqrt{2\pi}s_j}\exp\left[-\frac{(\theta_j^c - \theta_j^*)^2}{2s_j^2}\right] = q(\theta^*|\theta_j^c).$$

Metropolis-Hastings sampling

- Metropolis-Hastings (MH) sampling generalizes Metropolis sampling to allow for asymmetric candidate distributions

- For example, if $\theta_j \in [0, 1]$ then a reasonable candidate is

$$\theta_j^c | \theta_j^* \sim \text{Beta}[10\theta_j^*, 10(1 - \theta_j^*)]$$

- Then $q(\theta_j^* | \theta_j^c)$ and $q(\theta_j^c | \theta^*)$ are both beta PDFs

- MH proceeds exactly like Metropolis except the acceptance probability is

$$R = \min \left\{ 1, \frac{p(\theta_j^c | \theta_{(j)}, \mathbf{Y}) q(\theta_j^* | \theta_j^c)}{p(\theta_j^* | \theta_{(j)}, \mathbf{Y}) q(\theta_j^c | \theta_j^*)} \right\}$$

Metropolis-Hastings sampling

- What if we take the candidate distribution to be the full conditional distribution

$$\theta_j^c \sim p(\theta_j^c|\theta_{(j)}, \mathbf{Y})$$

- What is the acceptance ratio?

$$\frac{p(\theta_j^c|\theta_{(j)}, \mathbf{Y})q(\theta_j^*|\theta_j^c)}{p(\theta_j^*|\theta_{(j)}, \mathbf{Y})q(\theta_j^c|\theta_j^*)} = \frac{p(\theta_j^c|\theta_{(j)}, \mathbf{Y})p(\theta_j^*|\theta_{(j)}, \mathbf{Y})}{p(\theta_j^*|\theta_{(j)}, \mathbf{Y})p(\theta_j^c|\theta_{(j)}, \mathbf{Y})} = 1$$

- What does this say about the relationship between Gibbs and Metropolis Hastings sampling?

- Gibbs is a special case of MH with the full conditional as the candidate

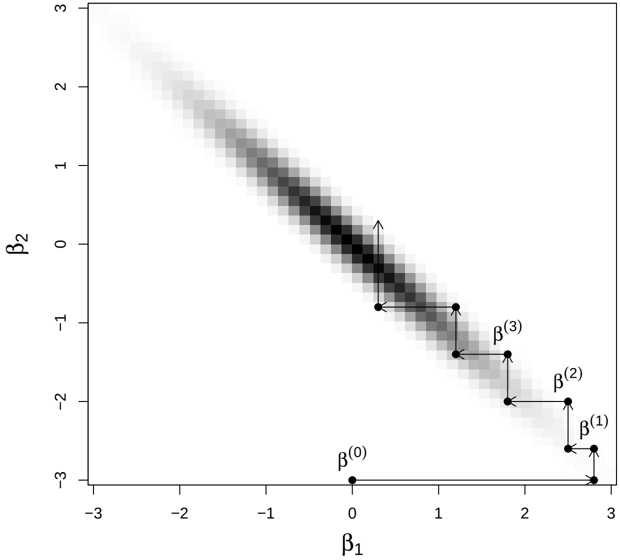
Variants

- You can combine Gibbs and Metropolis in the obvious way, sampling directly from full conditional when possible and Metropolis otherwise
- Adaptive MCMC varies the candidate distribution throughout the chain
- Hamiltonian MCMC uses the gradient of the posterior in the candidate distribution and is used in STAN

Blocked Gibbs/Metropolis

- If a group of parameters are highly correlated convergence can be slow
- One way to improve Gibbs sampling is a block update
- For example, in linear regression might iterate between sampling the block $(\beta_1, ..., \beta_p)$ and σ^2
- Blocked Metropolis is possible too
- For example, the candidate for $(\beta_1, ..., \beta_p)$ could be a multivariate normal

Posterior correlation leads to slow convergence



Summary

- With the combination of Gibbs and Metropolis-Hastings sampling we can fit virtually any model
- In some cases Bayesian computing is actually preferable to maximum likelihood analysis
- In most cases Bayesian computing is slower
- However, in the opinion of many it is worth the wait for improved uncertainty quantification and interpretability
- In all cases it is important to carefully monitor convergence

Options for coding MCMC

- Writing your own code
- Bayesian options in SAS procedures
- R packages for specific models
- All-purpose software like JAGS, BUGS, PROC MCMC, and STAN

Bayes in SAS procedures and R functions

- Here is a SAS proc

```
proc phreg data=VALung;
    class PTherapy(ref='no') Cell(ref='large')
    Therapy(ref='standard');
    model Time*Status(0) = KPS Duration;
    bayes seed=1 outpost=cout coeffprior=uniform
    plots=density;
run;
```

- In `R` you can use `BLR` for linear regression, `MCMClogit` for logistic regression, etc.

Why Just Another Gibbs Sampler (JAGS)?

- You can fit virtually any model
- You can call JAGS from R which allows for plotting and data manipulation in R
- It runs on all platforms: LINUX, Mac, Windows
- There is a lot of help online
- R has many built in packages for convergence diagnostics

How does JAGS work?

- You specify the model by declaring the likelihood and priors
- JAGS then sets up the MCMC sampler, e.g., works out the full conditional distributions for all parameters
- It returns MCMC samples in a matrix or array
- It also automatically produces posterior summaries like means, credible sets, and convergence diagnostics
- User's manual: `http://blue.for.msu.edu/CSTAT_13/jags_user_manual.pdf`

Running JAGS from `R` has the following steps

1. Install JAGS: `https://sourceforge.net/projects/mcmc-jags/files/JAGS/4.x/Windows/`
2. Download `rjags` from CRAN and load the library
3. Specify the model as a string
4. Compile the model using the function `jags.model`
5. Draw burn-in samples using the function `update`
6. Draw posterior samples using the function `coda.samples`
7. Inspect the results using the `plot` and `summary` functions

Examples

- The course website has many example of Bayesian analyses using JAGS
- There are also comparisons with other software
- For moderately-sized problems JAGS is competitive with these methods
- For really big and/or complex analyses STAN is preferred
- JAGS is easier to code and so we will use it through the course, but you should be familiar with other software
- Once you understand JAGS, switching to the others is straightforward