(5) Multi-parameter models - Gibbs sampling

ST440/540: Applied Bayesian Analysis
Summarizing a posterior

- Given the data and prior the posterior is determined

- Summarizing the posterior gives parameter estimates, intervals, and hypothesis tests

- Most of these computations are integrals over the posterior

- For simple problems we can do these integrals with pencil and paper, or google

- For medium problems can be solved with numerical integration

- For hard problems we usually use MCMC
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
Monte Carlo sampling

- Notation: Let $\theta = (\theta_1, ..., \theta_p)$ be the collection of all parameters in the model

- Notation: Let $Y = (Y_1, ..., Y_n)$ be entire dataset

- The posterior $f(\theta | Y)$ is a distribution

- If $\theta^{(1)}, ..., \theta^{(S)}$ are samples from $f(\theta | Y)$, 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 $p(\theta | Y)$?
Software options

- 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
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 was proposed in the early 1990s (Geman and Geman, 1984; Gelfand and Smith, 1990) and fundamentally changed Bayesian computing.

- 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).
Gibbs sampling for the Gaussian model

- Likelihood: $Y_i | \mu, \sigma \sim N(\mu, \sigma^2)$ independent over $i = 1, \ldots, n$

- Priors: $\mu \sim N(\mu_0, \sigma_0^2)$ independent of $\sigma^2 \sim \text{InvGamma}(a, b)$

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

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

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

- In the Gaussian model $\theta = (\mu, \sigma^2)$ so $\theta_1 = \mu$ and $\theta_2 = \sigma^2$

- The algorithm begins by setting initial values for all parameters, $\theta^{(0)} = (\theta_1^{(0)}, ..., \theta_p^{(0)})$.

- Variables are then sampled one at a time from their full conditional distributions,

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

- 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 $\theta^{(0)} = (\theta_1^{(0)}, ..., \theta_p^{(0)})$

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

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

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

▶ Monitoring convergence is the topic of the next lecture
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 http://www4.stat.ncsu.edu/~reich/ABA/derivations5.pdf
Practice problem

Work out the full conditionals for $\lambda$ and $b$ for the following model:

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