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.
- Because the low-dimensional updates are done in a loop, samples are not independent as in rejection sampling.
- The dependence of the samples turns out to follow a Markov distribution, leading to the name Markov chain Monte Carlo (MCMC).

Gibbs sampling

- 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 sample from p-dimensional joint, we make p 1-dimensional samples.
- The process is repeated until the required number of samples have been generated.
- Formally, the algorithm is:

Gibbs sampling

- Simple linear regression: $Y_i \sim \text{N}(\beta_0 + X_i\beta_1, \sigma^2)$, $\beta_j \sim \text{N}(\mu_j, \tau_j^2)$ and $\sigma^2 \sim \text{InvGamma}(a, b)$.
- **Initial values**: A reasonable choice would be to set β_0, β_1, and σ^2 at their MLEs.
- The **full conditionals** are all conjugate:
  - $\beta_0|\beta_1, \sigma^2, \mathbf{y} \sim$
  - $\beta_1|\beta_0, \sigma^2, \mathbf{y} \sim$
  - $\sigma^2|\beta_0, \beta_1, \mathbf{y} \sim$

Code

```
 n <- length(y)

#intial values:

 ols     <- lm(y~X)
 sigma2  <- var(ols$residuals)
 beta    <- ols$coef

#Initialize matrix to store the results:

 samples           <- matrix(0,n.samples,3)
 colnames(samples) <- c("beta1","beta2","sigma^2")

#Start the MCMC sampler:

 for(i in 1:n.samples){

     #update sigma^2:

      SSE    <- sum((y-beta[1]-X*beta[2])^2)
      sigma2 <- 1/rgamma(1,n/2+a,SSE/2+b)

     #update beta1:

      VVV     <- n/sigma2 + 1/tau[1]^2
      MMM     <- sum(y-X*beta[2])/sigma2 + mu[1]/tau[1]^2
      beta[1] <- rnorm(1,MMM/VVV,1/sqrt(VVV))

     #update beta2:

      VVV     <- sum(X^2)/sigma2 + 1/tau[2]^2
      MMM     <- sum(X*(y-beta[1]))/sigma2 + mu[2]/tau[2]^2
      beta[2] <- rnorm(1,MMM/VVV,1/sqrt(VVV))

     #store results:

      samples[i,]  <- c(beta,sigma2)
 }
```

Code is online at

http://www4.stat.ncsu.edu/~reich/ST740/code/BayesSLM.R.

Gibbs sampling

- Why does this work? $\boldsymbol{\theta}^{(0)}$ isn't a sample from the posterior. $\boldsymbol{\theta}^{(1)}$ likely isn't either.
- However, once we get one sample from the posterior (i.e., convergence) then the next one is also from the posterior:

Gibbs sampling

- When does the chain converge? That is, for which t can we assume that $\boldsymbol{\theta}^{(t)} \sim p(\boldsymbol{\theta}|\mathbf{y})$?
- Proving that the chain will eventually converge requires many stochastic process theorems.
- The Markov chain will converge to $p(\boldsymbol{\theta}|\mathbf{y})$ if it satisfies the detailed balance condition

$$p(\boldsymbol{\theta}^{(t)}|\mathbf{y})P(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) = p(\boldsymbol{\theta}^{(t+1)}|\mathbf{y})P(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t+1)}),$$

  where $P(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)})$ is the PDF of the transition from one iteration to the next.
- Geman and Geman showed this holds for Gibbs sampling.
- This is an asymptotic result, in practice we need a finite time to mark convergence.
- Typically, we sample say $S = 50,000$ draws and discard the first say $B = 5,000$ as burn-in, and then use the last $S - B$ samples for posterior inference.
- Ideally it looks like this:

- These samples are not independent! They follow a first-order Markov chain, meaning that $\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t-1)}$ is independent of samples before $t - 1$.
- However, the mean (or other summaries) of the samples remains an unbiased estimate of the posterior mean.

Variants of Gibbs sampling

- Updating parameters one at a time can lead to high autocorrelation (the lag h autocorrelation function (ACF) is $\rho(h) = \text{Cor}[\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t+h)}]$.
- Autocorrelation leads to slow convergence and imprecise estimates.
- This is especially true with the posterior correlation between parameters is high.

- Many variants including blocked and collapsed Gibbs have been proposed for this.

Blocked Gibbs sampling

- Rather than update parameters one at a time, we update blocks of parameters one at a time.
- Putting highly-correlated parameters in the block can improve convergence and mixing.
- Consider the simple linear regression case and define $\boldsymbol{\theta}_1 = (\beta_0, \beta_1)$ and $\boldsymbol{\theta}_2 = \sigma^2$.
- Blocked Gibbs alternates between
  1. $\boldsymbol{\theta}_1^{(t)} \sim p(\boldsymbol{\theta}_1 | \boldsymbol{\theta}_2^{(t-1)}, \mathbf{y})$
  2. $\boldsymbol{\theta}_2^{(t)} \sim p(\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1^{(t)}, \mathbf{y})$
- The full conditionals are

Code

```
n <- length(y)
p <- ncol(X)

# Intial values:
sigma2 <- rgamma(1,1,1)
beta   <- rnorm(p,0,10)

# Initialize vectors to store the results:
keep.sigma2 <- rep(0,n.samples)
keep.beta   <- matrix(0,n.samples,p)

# Pre-compute some matrices that will be used at each iteration
tXXinv  <- solve(t(X)%*%X)
betahat <- tXXinv%*%t(X)%*%y

#Start the MCMC sampler!
for(i in 1:n.samples){

  # Update beta:
  beta   <- rmvnorm(1,betahat,sigma2*tXXinv)
  beta   <- as.vector(beta)

  # Update sigma2:
  sigma2 <- 1/rgamma(1,n/2+a,sum((y-X%*%beta)^2)/2+b)

  # Store results:
  keep.sigma2[i] <- sigma2
  keep.beta[i,]  <- beta
}
```

Code is online at

http://www4.stat.ncsu.edu/∼reich/ST740/code/BayesLM.R.

Collapsed Gibbs sampling

- In some cases it is possible to marginalize out parameters in closed form.
- For example, in the linear regression case the marginal posterior distribution of $\boldsymbol{\theta}_2$ is

- In this case, we can just do Gibbs sampling to obtain draws from $\boldsymbol{\theta}_2^{(t)} \sim p(\boldsymbol{\theta}_2|\mathbf{y})$.
- After this MCMC is computed, we can then sample the values of $\boldsymbol{\theta}_1$ from $\boldsymbol{\theta}_1^{(t)} \sim p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2^{(t)}, \mathbf{y})$.
- The samples $\boldsymbol{\theta}^{(t)} = (\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)})$ have the correct joint posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$.
- MCMC is now performed on a lower dimensional problem which may be easier code and faster to converge.