BSTA 670:

Bayesian Computation: MCMC Sampling, Integration, and Approximate Inference

Arman Oganisian
aoganisi@upenn.edu
April 23, 2019

Overview of Bayesian Inference

- Parameter vector $\theta \in \mathbb{R}^p$ and data D.
- $\mathcal{L}(\theta|D) = p(D|\theta)$ with prior $p(\theta)$ over parameters space.

$$\begin{aligned} p(\theta|D) &= C \cdot p(D|\theta)p(\theta) \\ &\propto p(D|\theta)p(\theta) \end{aligned}$$

- Inference engines:
  - Frequentist: **optimization methods** for maximizing $p(D|\theta)$.
  - Bayesian: **sampling methods** for drawing from $p(\theta|D)$.
  - Difficult since C unknown.

Gibbs Sampler for Linear Regression

- Data $D = (y_i, x_i)_{1:n}$ and $\theta = (\beta, \phi)$, where $x_i, \beta \in \mathbb{R}^{p+1}$.
- $p(D|\theta) = \prod_i p(y_i|x_i, \theta) \overset{d}{=} \prod_i N(y_i\ ;\ x_i'\beta, \phi)$.
- If we use joint prior $p(\theta) = p(\beta)p(\phi) = N_{p+1}(0, I)IG(\alpha, \lambda)$, then
  - $p(\beta|\phi, D) = N_{p+1}\Big((I + \frac{1}{\phi}X'X)^{-1}(\frac{1}{\phi}X'y), (I + \frac{1}{\phi}X'X)^{-1}\Big)$.
  - $p(\phi|\beta, D) = IG(\alpha + n/2, \lambda + \frac{1}{2}(y - X\beta)'(y - X\beta))$
- **Gibbs Sampling**: sample from these two conditionals in alternating fashion
  - $\beta^{(t)}|\phi^{(t-1)} \sim N_{p+1}\Big((I + \frac{1}{\phi^{(t-1)}}X'X)^{-1}(\frac{1}{\phi^{(t-1)}}X'y), (I + \frac{1}{\phi^{(t-1)}}X'X)^{-1}\Big)$
    .
  - $\phi^{(t)}|\beta^{(t)} \sim IG(\alpha + n/2, \lambda + \frac{1}{2}(y - X\beta^{(t)})'(y - X\beta^{(t)}))$.
- Claim: The samples $\{\beta^{(t)}, \phi^{(t)}\}_{1:T}$ converge to draws from the posterior $p(\beta, \phi|D)$.

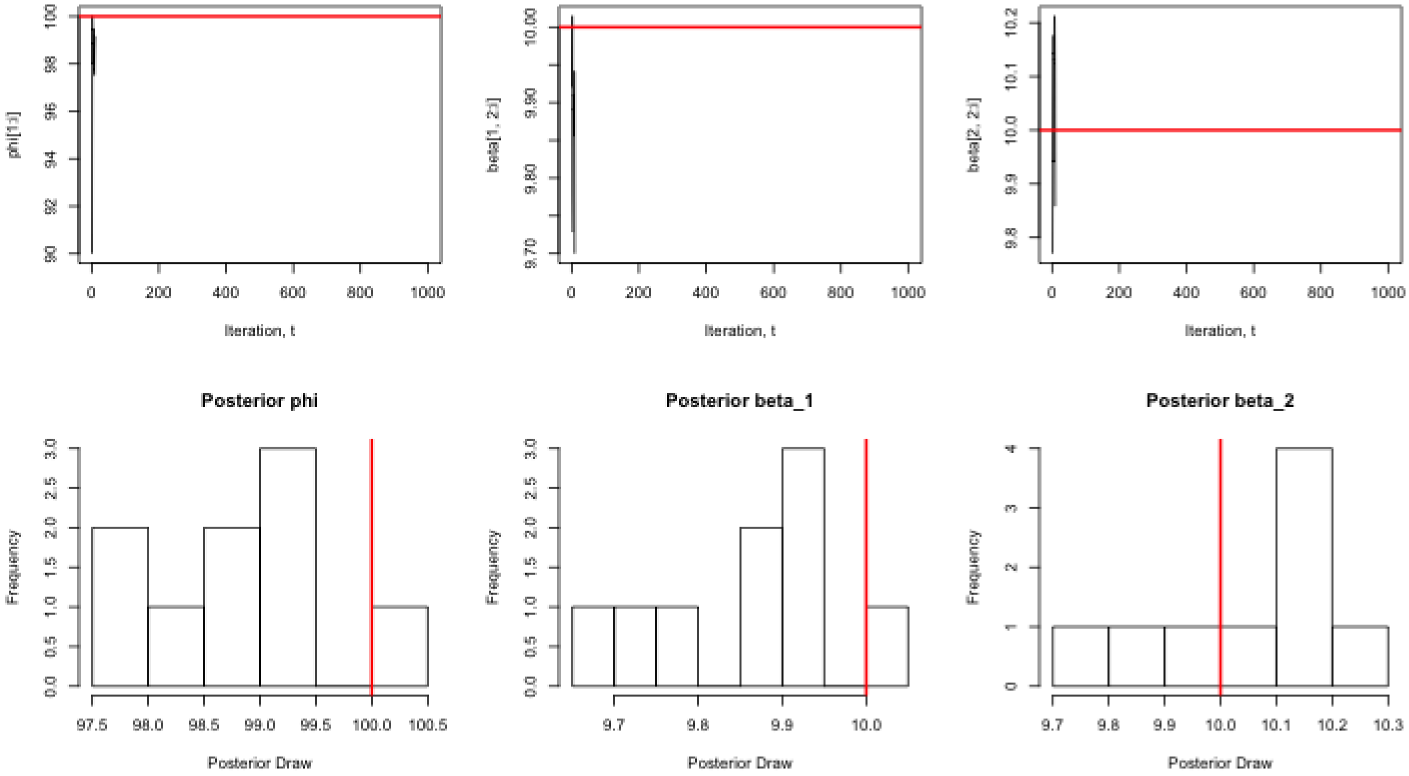
Gibbs Sampling

```
for( i in 2:iter){
  post_cov <- solve(Imat + (1/phi[i-1]) * xtx)
  post_mean <- post_cov %*% ((1/phi[i-1]) * t(X) %*% y)
  beta[, i]  <- MASS::mvrnorm(1, post_mean , post_cov )

  post_rate <- 100 + .5*sum((y - X %*% beta[, i, drop=F])^2)
  post_shape <- 5 + n/2
  phi[i] <- invgamma::rinvgamma(1, post_shape, rate = post_rate)
}
```

- We can plot the sequences or "chains": $\{\beta^{(t)}\}_{1:T}$ and $\{\phi^{(t)}\}_{1:T}$.
- These are the **Monte Carlo Markov Chains**.
  - **Monte Carlo**: each element of the chain is randomly drawn/simulated.
  - **Markov**: $\theta^{(t)}$ only depends on the previous element $\theta^{(t-1)}$.

Gibbs Sampling



MCMC - Checks and Limitations of Gibbs

- After sampling, must conduct visual and formal checks for
  - Convergence.
  - Autocorrelation.
  - Sensitivity to initial values.
- Gibbs requires known conditional posteriors: $p(\beta|\phi, D), p(\phi|\beta, D)$.
- In models without conjugacy, these are unknown - all we know is the form of $p(\theta|D)$ up to a proportionality constant.

Sampling for a Logistic Regression

- Data $D = (y_i, x_i)_{1:n}$, where $x_i \in \mathbb{R}^{p+1}$ and $y_i \in \{0, 1\}$.

$$p(D|\theta) \overset{d}{=} \prod_i Ber\big(\, y_i \;;\; expit(x_i'\theta)\big)$$

$$p(\theta) \overset{d}{=} N_{p+1}(0, I)$$

- Posterior is unknown:

$$p(\theta|D) \propto N_{p+1}(0, I) \prod_i Ber\big(\, y_i \;;\; expit(x_i'\theta)\big)$$

- Gibbs can't be used here.

The Metropolis-Hastings Sampler

Along with initial value, $\theta^{(0)}$, MH algorithm requires two inputs:

- **Unnormalized target density**, $\tilde{p}(\theta|D)$:

$$p(\theta|D) = C \cdot \tilde{p}(\theta|D) = C \cdot Ber\big(expit(x_i'\theta)\big)N_{p+1}(0, I)$$

- **Jumping Distribution**:

$$Q(\theta^*|\theta) = N_{p+1}(\theta, \tau)$$

for t = *1 to* T **do**

$\theta^* \sim N(\theta^{(t-1)}, \tau)$

$\alpha = \frac{\tilde{p}(\theta^*|D)}{\tilde{p}(\theta^{(t-1)}|D)}$

$U \sim Ber(p = min(1, \alpha))$

if $U == 1$ **then**

$\theta^{(t)} \leftarrow \theta^*$

else

$\theta^{(t)} \leftarrow \theta^{(t-1)}$

end

end

MH for Univariate Logit Model

```r
# target log density
p_tilde <- function(y, x, theta){
  p <- invlogit( x %*% theta)
  lik <- sum(dbinom(y, 1, p, log = T))
  pr <- sum(dnorm( theta, 0, 100, log = T))
  eval <- lik + pr
  return(eval)
}

iter <- 1000 # number of iterations
tau <- .1 # proposal sd
theta <- matrix(NA, nrow = 2, ncol = iter) # for storing draws
theta[,1] <- c(0,0) # initialize

for(i in 2:iter){
  # propose theta
  theta_star <- MASS::mvrnorm(1, theta[,i-1] , tau*diag(2) )

  # accept/reject
  prop_eval <- p_tilde(y, x, theta_star)
  curr_eval <- p_tilde(y, x, theta[,i-1, drop=F])
  ratio <- exp( prop_eval - curr_eval )
  U <- rbinom(n = 1, size = 1, prob = min( c(1, ratio  ) ) )
  theta[, i] <- U*theta_star + (1-U)*theta[, i-1]
}
```

MH for Univariate Logit Model

Extensions

- Proposal distributions for constrained variables.
  - Using non-Gaussian proposals or $log()$ transform.
- Sensitivity to proposal variance τ.
  - **Adaptive Metropolis-Hastings**.
  - Tunes τ periodically to target a desired acceptance rate.
- MH often fails in high-dimensions.
  - **Hamiltonian Monte Carlo**: leverages the gradient of $\tilde{p}$.
- Other MCMC algorithms (all similar to MH):
  - **Reversible Jump MCMC** for model selection.
  - **Split-Merge MCMC** for clustering analysis.
  - **Data Augmentation** for missing data problems.

Monte Carlo Integration

- We covered methods for obtaining draws $\{\theta^{(t)}\}_{1:T}$ from $p(\theta|D)$
  - Often we need summary quantities:

$$E[\theta|D] = \int_{\Theta} \theta p(\theta|D) d\theta$$

$$V[\theta|D] = \int_{\Theta} (\theta - E[\theta|D])^2 p(\theta|D) d\theta$$

$$E[\tilde{y}|\tilde{x}, D] = \int_{\Theta} E[\tilde{y}|\tilde{x}, \theta] p(\theta|D) d\theta$$

  - Computing useful quantities requires integration - hard if $dim(\theta)$ is big.

Monte Carlo Integration

Recall Monte Carlo (MC) integration. Given $i.i.d$ samples $\{\theta^{(t)}\}_{1:T} \sim p(\theta|D)$,

$$E[g(\theta)|D] = \int_\Theta g(\theta)p(\theta|D)d\theta \approx \frac{1}{T}\sum_{t=1}^{T} g(\theta^{(t)})$$

- For posterior expectation: $g(\theta^{(t)}) = 1$
- For posterior variance: $g(\theta^{(t)}) = (\theta^{(t)} - \frac{1}{T}\sum_t \theta^{(t)})^2$
- For posterior prediction: $g(\theta^{(t)}) = E[\tilde{y}|\tilde{x}, \theta^{(t)}]$

Some properties:

- Convergence rate, $\sqrt{T}$, independent of $dim(\theta)$.
- Converges to posterior mean exactly based on LLN.
- We have $\{\theta^{(t)}\}_{1:T}$ from MCMC, but they are not exactly $i.i.d$.
    - Effective number of draws is less than T.

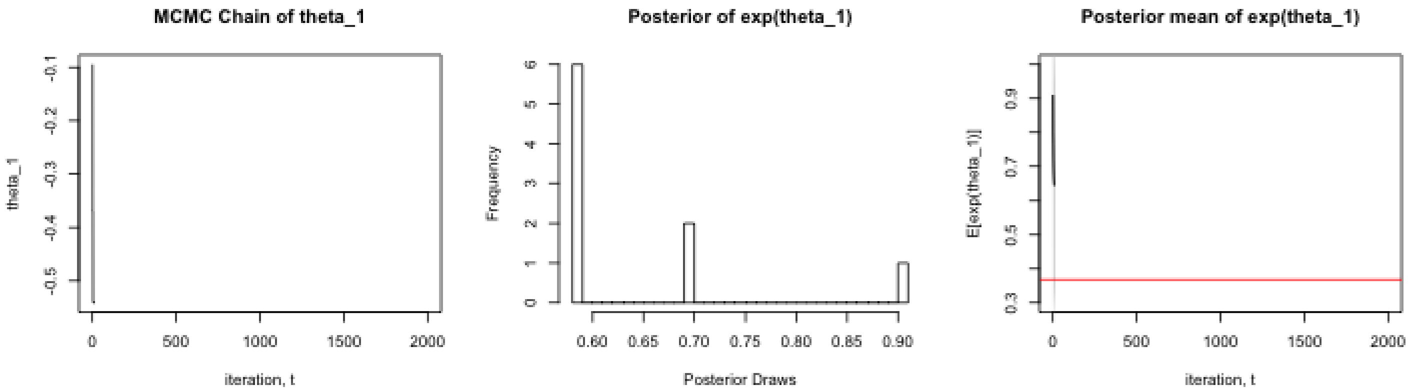
Variable Transformation

Earlier we used MH to get $\{\theta_1^{(t)}\}_{1:T} \sim p(\theta|D)$ from model

$$y_i \sim Ber\Big(p_i = expit(\theta_0 + \theta_1 x_i)\Big)$$

Suppose we want estimate $\hat{OR} = E[exp(\theta)|D]$

$$E[exp(\theta)|D] = \int_\Theta e^\theta \, p(\theta|D)d\theta \approx \frac{1}{T}\sum_{t=1}^{T} exp\big(\theta^{(t)}\big)$$



Bayesian Prediction

Suppose we sample parameters of this model with some prior

$$y_i|x_i, \theta_0, \theta_1, \phi \sim N(\ \theta_0 + \theta_1 x_i,\ \phi)$$

And get posterior draws $\{\theta_0^{(t)}, \theta_1^{(t)}, \phi^{(t)}\}_{1:T}$. Given new $\tilde{x}_i$, form out-of-sample prediction

$$\begin{aligned} E[\tilde{y}|\tilde{x}, D] &= \int_\Theta E[\ \tilde{y} \mid \tilde{x}, \theta_0, \theta_1, \phi] \cdot p(\theta_0, \theta_1, \phi|D) \\ &\approx \frac{1}{T}\sum_{i=1}^{T} E[\ \tilde{y} \mid \tilde{x}, \theta_0^{(t)}, \theta_1^{(t)}, \phi^{(t)}] = \frac{1}{T}\sum_{i=1}^{T} \theta_0^{(t)} + \theta_1^{(t)}\tilde{x}_i \end{aligned}$$

Approximation Methods

Upside of MCMC

- **Asymptotically exacty**: chains guaranteed to converge to exact posterior.
- **Can bound errors**: easy to measure autocorrelation in chains and error in integration.

Downside of MCMC:

- **Slow**: in complicated samplers, may take an hour to get a single draw!

Motivates the need for approximation methods: find some $q(\theta)$ such that

$$q(\theta) \approx p(\theta|D)$$

Variational Bayes

Find approximation $q^*(\theta)$ to $p(\theta|D)$ such that Kullback–Leibler divergence is minimized:

$$
\begin{aligned}
q* &= \underset{q}{\operatorname{argmin}}\, KL(q||p) \\
&= \underset{q}{\operatorname{argmin}} - \int_{\Theta} q(\theta)\ log\Big[\frac{p(\theta|D)}{q(\theta)}\Big] d\theta
\end{aligned}
$$

This is too hard of a search problem - space of $q(\theta)$ is too large.

Restrict $q(\theta)$ to the "mean-field family",

$$
q(\theta) = \prod_{i=1}^{p} q_i(\theta_i)
$$

Mean-Field Variational Bayes

Then the solution for each $q_j(\theta_j)$ is

$$\log q_j^*(\theta_j) \propto E_{\theta_{-j} \sim q_{-j}(\theta_{-j})}\Big[\log p(D|\theta)p(\theta)\Big]$$

These updating equations define a **Coordinate Ascent** algorithm:

- **Initialize**: $q_j(\theta_j)^{(0)} = q_j^{(0)}$ for $j = 1, \ldots, p$.
- **Update**: for $t = 1, \ldots, T$.
  - $q_1^{(t)}$ conditional on $q_2^{(t-1)}, \ldots, q_p^{(t-1)}$.
  - $q_2^{(t)}$ conditional on $q_1^{(t)}, q_3^{(t-1)}, \ldots, q_p^{(t-1)}$
  - $q_3^{(t)}$ conditional on $q_1^{(t)}, q_2^{(t)}, q_4^{(t-1)}, \ldots, q_p^{(t-1)}$
  - ...
  - $q_p^{(t)}$ conditional on $q_1^{(t)}, q_2^{(t)}, q_3^{(t)}, \ldots, q_p - 1^{(t-1)}$

Regression with Variational Bayes

Consider model for $D = (y_i, x_{0i}, x_{1i})_{1:n}$ with known ϕ

$$y_i|\theta \sim N\big(x_{0i}\theta_0 + x_{1i}\theta_1, \phi\big)$$

With prior $p(\theta_0)p(\theta_1) = N(0, \tau)N(0, \tau)$,

$$\log p(D|\theta)p(\theta) \propto \log \Big[\prod_{n=1}^{N} N(y_i; x_0\theta_0 + x_1\theta_1, \phi)\Big] + \log N(\theta_0; 0, \tau) + \log N(\theta_1; 0, \tau)$$

Regression with Variational Bayes

We assume mean-field family $q(\theta) = q_0(\theta_0)q_1(\theta_1)$.

Using the solution expression for $\log q_j^*$,

$$\log q_0^*(\theta_0) \propto E_{\theta_1 \sim q_1}\Big[\log p(D|\theta)p(\theta)\Big]$$

$$\propto -\frac{1}{2\big(\frac{\phi}{\sum_i x_{0i}^2 + \frac{\phi}{\tau}}\big)}\Big\{\theta_0^2 - 2\theta_0\Big[\frac{\sum_i x_{0i}y_i - E_{\theta_1 \sim q_1}[\theta_1]\sum_i x_{0i}x_{1i}}{\sum_i x_{0i}^2 + \frac{\phi}{\tau}}\Big]\Big\}$$

$$\Rightarrow q_0^*(\theta_0) \overset{d}{=} N(\mu_0, \lambda_0)$$

Where $\mu_0 = \frac{\sum_i x_{0i}y_i - E_{\theta_1 \sim q_1}[\theta_1]\sum_i x_{0i}x_{1i}}{\sum_i x_{0i}^2 - \frac{\tau}{2}}$ and $\lambda_0 = \frac{\phi}{\sum_i x_{0i}^2 - \frac{\tau}{2}}$.

- Note dependence on $E_{\theta_1 \sim q_1}[\theta_1]$
- Math is same for $q_1^*(\theta_1) \overset{d}{=} N(\mu_1, \lambda_1)$, with
  - $\mu_1 = \frac{\sum_i x_{1i}y_i - E_{\theta_0 \sim q_0}[\theta_0]\sum_i x_{0i}x_{1i}}{\sum_i x_{1i}^2 + \frac{\phi}{\tau}}$ and $\lambda_1 = \frac{\phi}{\sum_i x_{1i}^2 + \frac{\phi}{\tau}}$

Coordinate Ascent for VB

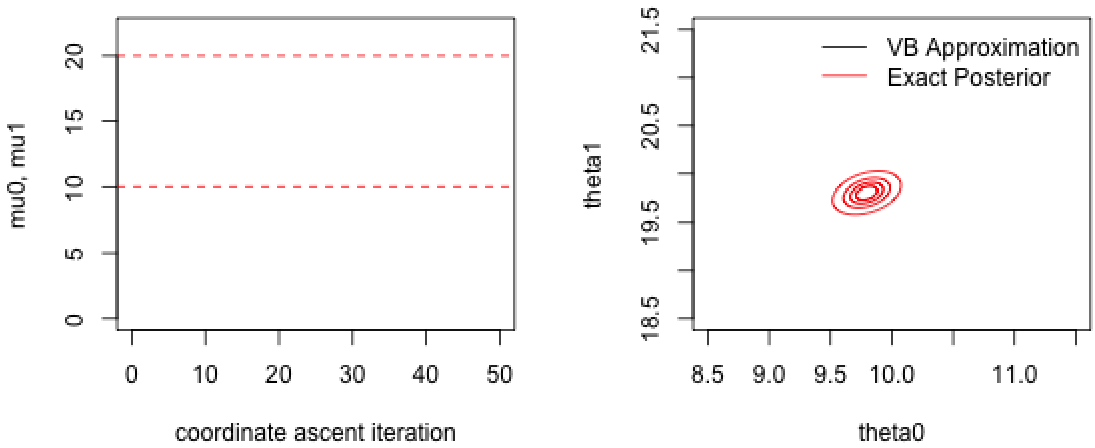
Note λ_1, λ_0 are known. They don't need to be updated.

Coordinate ascent with updating equations:

- Initialize $\mu_0^{(0)}, \mu_1^{(0)}$
- for $t = 1, \ldots, T$,
  - $\mu_0^{(t)} = \frac{\sum_i x_{0i}y_i - \mu_1^{(t-1)} \sum_i x_{0i}x_{1i}}{\sum_i x_{0i}^2 + \frac{\phi}{\tau}}$
  - $\mu_1^{(t)} = \frac{\sum_i x_{1i}y_i - \mu_0^{(t)} \sum_i x_{0i}x_{1i}}{\sum_i x_{1i}^2 + \frac{\phi}{\tau}}$

$$p(\theta|D) \approx q(\theta_0)q(\theta_1) = N\left\{ \begin{bmatrix} \mu_0^{(T)} \\ \mu_1^{(T)} \end{bmatrix}, \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \right\}$$

Coordinate Ascent for VB



Summary

- MCMC Sampling
  - Gibbs Sampling
  - Metropolis-Hastings
- Monte Carlo Integration
  - Computing posterior summaries
- Posterior Approximation with VB