

Bayesian Computing

STATISTICAL COMPUTING WORKSHOP, ISI DELHI

DAY 2



Why do we need Bayesian computing?

- Consider the simple linear regression model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, \dots, n$$

- Our parameters are then $\theta = (\beta_0, \beta_1, \sigma^2)'$
- To do Bayesian inference, first we need to decide on priors
- If we choose a Normal-Inverse-Gamma prior for θ , then the posterior distribution for θ would also be Normal-Inverse-Gamma.
- I can use the mean and variance of this distribution to infer
- What if I want to use any other prior?

Contd...

- I may want to use the following prior specification

$$\beta_0 \sim N(0, \sigma_b^2)$$

$$\beta_1 | \sigma^2 \sim DE(2\lambda)$$

$$\sigma^2 \sim IG(a, b)$$

- In this case, the posterior looks like:

$$\log \pi(\theta | X) \propto -\frac{n+a}{2} \log \sigma^2 - \frac{\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 + 2b}{2\sigma^2} - \lambda |\beta_1| - \frac{\beta_0^2}{2\sigma_b^2}$$

- This is not any known (log) pdf !
- How can I infer about the parameters still?

Markov Chain Monte Carlo (MCMC)

- Suppose $\{\theta_i\}_{i=1}^{\infty}$ is a Markov chain that is Harris recurrent with stationary distribution π
- We are interested in computing $E_{\pi}(f(\theta)) = \int f(u)\pi(u|X)du$
- We consider its sample analogue $S_n(f) = \frac{1}{n}\sum_{i=1}^n f(\theta_i)$
- If $f, g \in L^1(\pi)$, $\lim_{n \rightarrow \infty} \frac{S_n(f)}{S_n(g)} = \frac{\int f(u)\pi(u|X)du}{\int g(u)\pi(u|X)du}$ when $\int g(u)\pi(u|X)du \neq 0$
- In particular, if π is proper, and we take $g \equiv 1$, we see that the sample analogue will converge to the desired quantity

Will any Markov Chain work?

- NO!
- Posterior distribution must be a proper one for this to work
- Prior selection is therefore important
- The transition probabilities for the Markov Chain must have the **Detailed Balance Condition**:
$$\pi(s)P(s'|s) = \pi(s')P(s|s')$$
- We must choose the sampling algorithm carefully to ensure the DBC holds

Metropolis-Hastings Algorithm

1. Pick an initial state θ_0 ; Set $t \leftarrow 0$
 2. Repeat till $t = T$
 - a. Propose a candidate θ' following the conditional probability $g(\theta'|\theta_t)$
 - b. Calculate the acceptance probability $a(\theta', \theta_t) = \min \left\{ 1, \frac{\pi(\theta'|X)g(\theta_t|\theta')}{\pi(\theta_t|X)g(\theta'|\theta_t)} \right\}$
 - c. Generate $u \sim U(0,1)$ and set $\theta_{t+1} = \theta'$ if $u \leq a(\theta', \theta_t)$, set $\theta_{t+1} = \theta_t$ otherwise
 - d. Set $t \leftarrow t + 1$
- Samples are $\theta_0, \dots, \theta_T$
 - Usually, we discard initial few samples (burn-in period)

Metropolis-Hastings Example

- Consider the initial example of SLR where $\beta_0 = 0$ and $\sigma^2 = 1$ are known and $\beta_1 \sim DE(2\lambda)$
- Then, $\log \pi(\beta_1 | X) = -\frac{1}{2} \sum_{i=1}^n (y_i - \beta_1 x_i)^2 - \lambda |\beta_1|$
- We start with $\beta_1^{(0)} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}$
- We choose $g(\beta'_1 | \beta_1^{(t)}) = \frac{1}{\sqrt{2\pi\tau}} \exp - \frac{(\beta'_1 - \beta_1^{(t)})^2}{2\tau}$
- Then $g(\beta'_1 | \beta_1^{(t)}) = g(\beta_1^{(t)} | \beta'_1)$ and so, the acceptance ratio is simply the ratio of the posterior at β'_1 and $\beta_1^{(t)}$

Tuning and Thinning

- The tuning parameter τ can be tuned to improve acceptance
- But tuning must be done before the burn-in period ends
- Thinning is the process of selecting a subsample out of the obtained sample, often at a prescribed thinning interval, say every 5th sample is stored
- This is done to avoid any residual autocorrelation in the posterior sample

Metropolis Adjusted Langevin Algorithm

- RW Metropolis: $\theta' = \theta_t + \sqrt{\tau}N(0, I)$
- Instead, we can guide the proposal along the gradient for a better convergence
- MALA: $\theta' = \theta_t + \tau \nabla \log \pi(\theta_t|X) + \sqrt{2\tau}N(0, I)$

- The acceptance probability is

$$a(\theta', \theta_t) = \min \left\{ 1, \frac{\pi(\theta'|X)q(\theta_t|\theta')}{\pi(\theta_t|X)q(\theta'|\theta_t)} \right\},$$

with $q(\theta'|\theta) \propto \exp \left(-\frac{1}{4\tau} \left\| \theta' - \theta - \tau \nabla \log \pi(\theta|X) \right\|_2^2 \right)$

- Pro: Faster Convergence, Con: Computing gradient may be expensive

MALA with Preconditioner

- In reality, to capture the variation in the posterior distribution surface quickly, we should use a preconditioning matrix $P \in \mathbb{R}^{d \times d}$
- The proposal: $\theta' = \theta_t + \tau P \nabla \log \pi(\theta_t | X) + \sqrt{2\tau} P^{\frac{1}{2}} N(0, I)$
- The conditional probability then looks like:
$$q(\theta' | \theta) \propto \exp \left(-\frac{1}{4\tau} (\theta' - \theta - \tau P \nabla \log \pi(\theta | X))^T A^{-1} (\theta' - \theta - \tau P \nabla \log \pi(\theta | X)) \right)$$
- Ideally, P should be close to the negative Hessian: $-\nabla_2 \log \pi(\theta | X)$, and should remain unchanged throughout iterations ($P^{(t)} \equiv P$)

Gibbs Sampler

- Instead of drawing samples for all parameter together from the posterior sample, draw samples for each parameter individually, from their full conditionals.
 - Assume $\theta = (\theta_1, \theta_2, \dots, \theta_d)^T$
1. Get initial values of $\theta^{(0)}$; Set $t \leftarrow 0$
 2. Repeat till $t = T$
 - a. For k in $1, 2, \dots, d$
 - i. Compute full conditional distribution of $\theta_k := \pi(\theta_k | \theta_{1,\dots,k-1}, \theta_{k+1,\dots,d}, X)$
 - ii. Generate $\theta_k^{(t+1)} \sim \pi(\theta_k | \theta_{1,\dots,k-1}^{(t+1)}, \theta_{k+1,\dots,d}^{(t)}, X)$
 - iii. Set $k \leftarrow k + 1$
 - b. Set $t \leftarrow t + 1$

Example

- SLR problem with Gaussian priors for β_0, β_1 and Inverse-Gamma prior for σ^2
- Check that the full conditionals are:

$$\beta_0^{(t+1)} | \beta_1^{(t)}, \sigma^{2(t)}, X, y \sim N \left(\frac{n\sigma_0^2 (\bar{y} - \beta_1^{(t)} \bar{x})}{n\sigma_0^2 + \sigma^{2(t)}}, \frac{\sigma_0^2 \sigma^{2(t)}}{n\sigma_0^2 + \sigma^{2(t)}} \right)$$

$$\beta_1^{(t+1)} | \beta_0^{(t+1)}, \sigma^{2(t)}, X, y \sim N \left(\frac{\sigma_1^2 \sum_{i=1}^n (y_i - \beta_0^{(t)}) x_i}{\sigma^{2(t)} + \sigma_1^2 \sum_{i=1}^n x_i^2}, \frac{\sigma_1^2 \sigma^{2(t)}}{\sigma^{2(t)} + \sigma_1^2 \sum_{i=1}^n x_i^2} \right)$$

$$\sigma^{2(t+1)} | \beta_0^{(t+1)}, \beta_1^{(t+1)} \sim IG \left(a + \frac{n}{2}, b + \frac{1}{2} \sum_{i=1}^n (y_i - \beta_0^{(t+1)} - \beta_1^{(t+1)} x_i)^2 \right)$$

Block Gibbs Sampler

- Instead of updating all the parameters individually, I could update them in blocks
- In the last example, I can update $\beta = (\beta_0, \beta_1)^T$ simultaneously and σ^2 separately
- The full conditionals are:

$$\beta^{(t+1)} | \sigma^{2(t)}, X, y \sim N \left(\frac{1}{\sigma^{2(t)}} \left(\frac{1}{\sigma^{2(t)}} X^T X + D_\beta \right)^{-1} X^T y, \left(\frac{1}{\sigma^{2(t)}} X^T X + D_\beta \right)^{-1} \right)$$

$$\sigma^{2(t+1)} | \beta^{(t+1)}, X, y \sim IG \left(a + \frac{n}{2}, b + \frac{1}{2} (y - X\beta^{(t+1)})^T (y - X\beta^{(t+1)}) \right)$$

with D_β being a diagonal matrix with diagonal entries σ_0^2 and σ_1^2 .

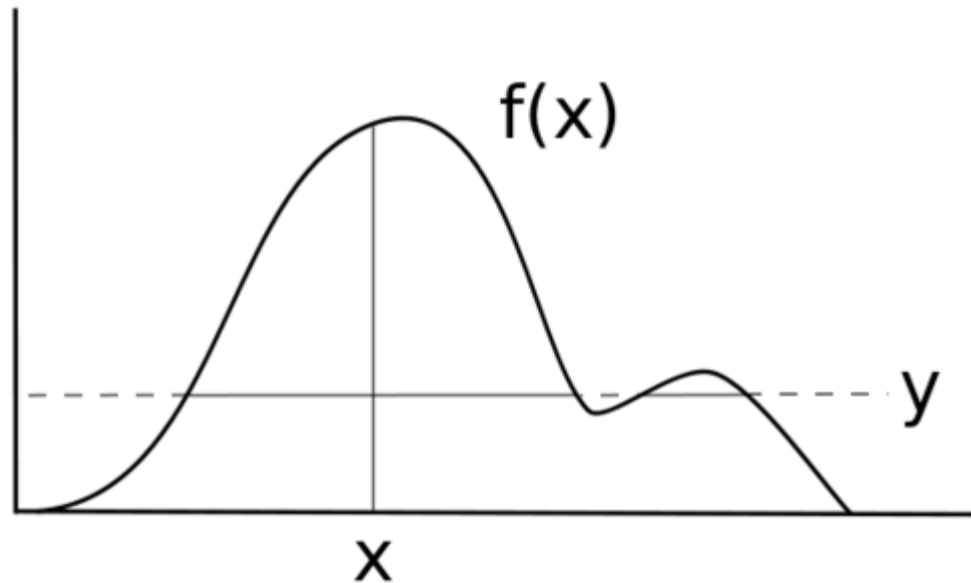
Metropolis within Gibbs Sampler

- Use Gibbs sampler and draw from the full conditional distribution of the parameters
- If the full conditional is not a standard distribution, we use MH to draw from the full conditional.
- SLR example with Laplace prior for β_1
- We have normal and inverse-gamma full conditionals for β_0 and σ^2
- For β_1 we use an MH (or MALA) draw from the full conditional distribution

Slice Sampler (Neal, 2003)

- a)* Draw a real value, y , uniformly from $(0, f(x_0))$, thereby defining a horizontal ‘slice’:
 $S = \{ x : y < f(x) \}$. Note that x_0 is always within S .
 - b)* Find an interval, $I = (L, R)$, around x_0 that contains at least a big part of the slice.
 - c)* Draw the new point, x_1 , from the part of the slice within this interval (ie, from $S \cap I$).
- Finding the interval and sampling from it is not straightforward, but has algorithms for it.

Slice Sampler



- Can be used in conjunction with Gibbs sampler, Slice within Gibbs sampler

Elliptical Slice Sampler (Murray *et al.*, 2010)

- Can be only used if the posterior/full-conditional is Gaussian

Input: current state $\mathbf{f}$, a routine that samples from $\mathcal{N}(0, \Sigma)$, log-likelihood function $\log L$.

Output: a new state $\mathbf{f}'$. When $\mathbf{f}$ is drawn from $p^*(\mathbf{f}) \propto \mathcal{N}(\mathbf{f}; 0, \Sigma) L(\mathbf{f})$, the marginal distribution of $\mathbf{f}'$ is also p^* .

1. Choose ellipse: $\boldsymbol{\nu} \sim \mathcal{N}(0, \Sigma)$

2. Log-likelihood threshold:

$$u \sim \text{Uniform}[0, 1]$$

$$\log y \leftarrow \log L(\mathbf{f}) + \log u$$

3. Draw an initial proposal, also defining a bracket:

$$\theta \sim \text{Uniform}[0, 2\pi]$$

$$[\theta_{\min}, \theta_{\max}] \leftarrow [\theta - 2\pi, \theta]$$

4. $\mathbf{f}' \leftarrow \mathbf{f} \cos \theta + \boldsymbol{\nu} \sin \theta$

5. **if** $\log L(\mathbf{f}') > \log y$ **then:**

6. Accept: **return** $\mathbf{f}'$

7. **else:**

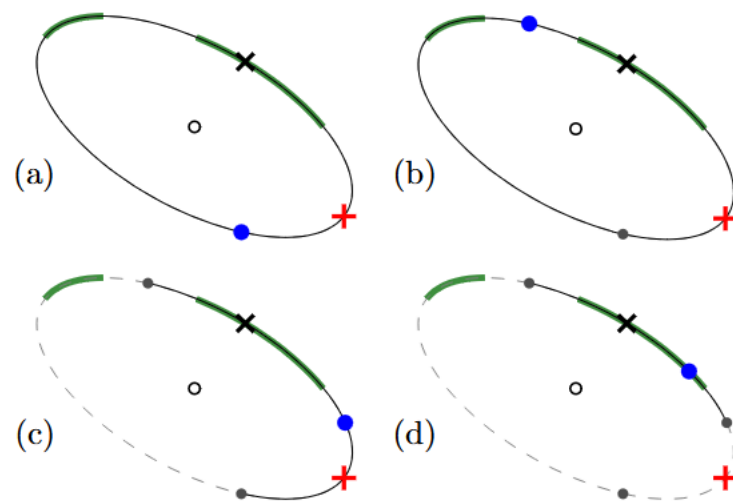
 Shrink the bracket and try a new point:

8. **if** $\theta < 0$ **then:** $\theta_{\min} \leftarrow \theta$ **else:** $\theta_{\max} \leftarrow \theta$

9. $\theta \sim \text{Uniform}[\theta_{\min}, \theta_{\max}]$

10. **GoTo** 4.

Elliptical Slice Sampler



Hamiltonian Monte Carlo (Neal, 2011)

- Generates from a better proposal distribution and improves acceptance rate
- Proposes momentum $\omega_0 \sim p(\omega) \propto \exp(-V(\omega))$
- Runs the Hamiltonian dynamics for T time, deterministically updating (θ, ω)
- Compute acceptance probability using Hamiltonian energies $H(\theta_T, \omega_T)$ and $H(\theta_0, \omega_0)$, where

$$H(\theta, \omega) = -\log \pi(\theta|X) - V(\omega)$$

- Classically, $p(\omega)$ is taken to be Gaussian

HMC & Leapfrog Method

1. Generate the initial momentum $\omega_0 \sim N(0, M^{-1})$.
2. Set $x^{(0)} = x_0$.
3. For the momentum, make a half-step update:

$$\omega^{(0)} = \omega_0 - \frac{\epsilon}{2} \nabla \log r(x^{(0)}).$$

4. For $\ell = 1, \dots, L - 1$, do the followings:
 - (a) Update position: $x^{(\ell)} = x^{(\ell-1)} + \epsilon \cdot \omega^{(\ell-1)}$.
 - (b) Update momentum: $\omega^{(\ell)} = \omega^{(\ell-1)} - \epsilon \cdot \nabla \log r(x^{(\ell)})$.
5. Make one last update on the position: $x^{(L)} = x^{(L-1)} + \epsilon \cdot \omega^{(L-1)}$.
6. Make another half-step update of the momentum:

$$\omega^{(L)} = \omega^{(L-1)} - \frac{\epsilon}{2} \nabla \log r(x^{(L)}).$$

7. Compute the acceptance probability:

$$a(x_0, \omega_0, x^{(L)}, \omega^{(L)}) = \min \left\{ 1, \frac{\exp(-H(x^{(L)}, \omega^{(L)}))}{\exp(-H(x_0, \omega_0))} \right\}.$$

8. Accept $x_{\text{new}} = x^{(L)}$ with a probability of $a(x_0, \omega_0, x^{(L)}, \omega^{(L)})$. If we reject, then $x_{\text{new}} = x_0$.
9. Return x_{new} .

No U-Turn Sampler (Hoffman & Gelman, 2014)

- Runs the HMC leapfrog algorithm with autotuning the length of the leapfrog distance L
- Loosely speaking, it runs a leapfrog method forward and backward simultaneously while increasing the size of the leapfrog distance
- Stops when a U-turn is reached or the Hamiltonian becomes unacceptable
- The path is tracked using a binary tree and finally a sample from this path is selected randomly as the new proposal

Reversible Jump MCMC (Green, 1995)

- Required when the model parameter space itself can change between iterations.
- In a variable selection method, the parameter space can be either (β_0, σ^2) or $(\beta_0, \beta_1, \sigma^2)$
- In this setting, the DBC for standard MCMC may not be satisfied.
- This is handled by the Markov chain itself being reversible which allows “dimension matching” and which in turn allows for the DBC being satisfied
- Bayesian variable selection methods usually do not do variable selection, only introduces shrinkage, and hence do usually not require RJMCMC

Autocorrelation in Posterior Samples

- Consecutive samples almost always have some correlation between them
- The autocorrelation for estimating $\mu_f = \int f(\theta)\pi(\theta|X)d\theta$ using $\hat{\mu}_f = \frac{1}{N} \sum_{i=1}^N f(\theta_i)$ with θ_i being the samples, is given as $\hat{\rho}(\tau) = \hat{c}_f(\tau)/\hat{c}_f(0)$ where

$$\hat{c}_f(\tau) = \frac{1}{N - \tau} \sum_{i=1}^{N-\tau} (f(\theta_i) - \hat{\mu}_f)(f(\theta_{i+\tau}) - \hat{\mu}_f)$$

- The integrated autocorrelation is defined as $\tau_f = \sum_{i=-\infty}^{\infty} \rho(\tau)$ as estimated as

$$\hat{\tau}_f = 1 + 2 \sum_{\tau=1}^M \hat{\rho}_f(\tau),$$

with $M \ll N$.

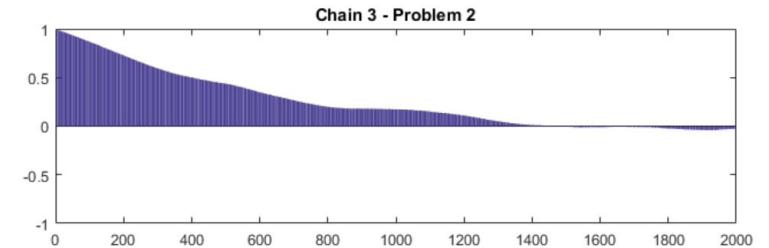
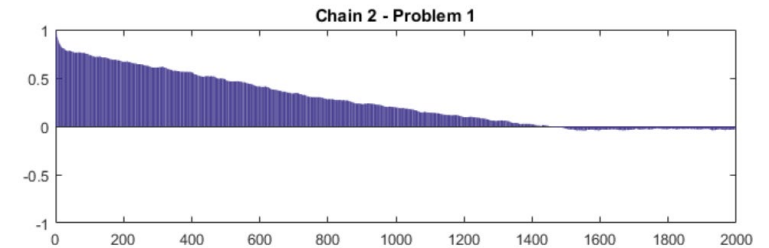
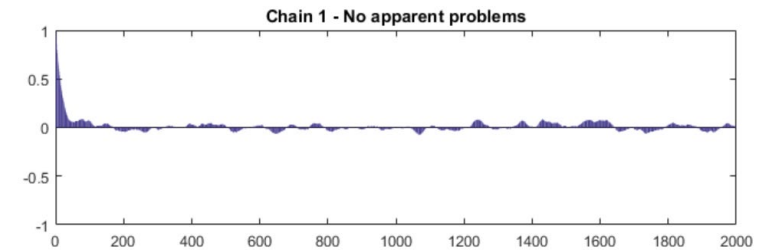
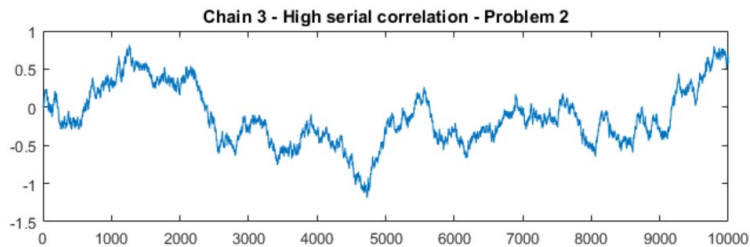
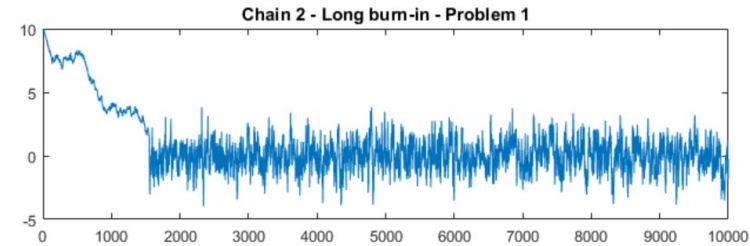
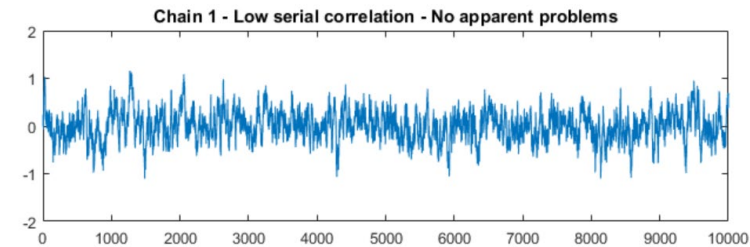
Effective Sample Size

- Effective sample size (N_{eff}) is the number of independent samples drawn
- The error rate for MCMC based inference is $\frac{1}{\sqrt{N_{eff}}}$ and not $\frac{1}{\sqrt{N}}$
- It is the number of independent samples from the posterior that would give the same quality inference as N dependent samples.
- It is calculated as $N_{eff} = \frac{N}{\tau_f}$ and estimated as $\hat{N}_{eff} = \frac{N}{\hat{\tau}_f} = \frac{N}{1 + 2 \sum_{\tau=1}^M \rho(\tau)}$

Diagnostic Checks

- There are multiple metrics for quantifying convergence quality, such as Gelman and Rubin's R (Gelman and Rubin, 1992), or Geweke's diagnostic (Geweke, 1992)
- All of these depend on ratios of between and within chain variations (for multiple chains) and the said ratio being small denoting convergence
- Usually, if Gelman's R is below 1.05 or 1.01 we say it has converged

Diagnostics using Trace and ACF Plots



References

1. Neal, R. M. (2003). Slice sampling. *The annals of statistics*, 31(3), 705-767.
2. Murray, I., Adams, R., & MacKay, D. (2010, March). Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 541-548). JMLR Workshop and Conference Proceedings.
3. Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11), 2.
4. Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1), 1593-1623.
5. Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4), 711-732.
6. Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4), 457-472.
7. Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments. *Bayesian statistics*, 4, 641-649.