DSL Seminar: MCMC (7)

Kyung-han Kim

Data Science Lab

August, 2023

목차

- Gibbs Sampler
- Bayesian Regression with M-H algorithm

Bayesian Regression

- Bayesian regression의 핵심은, 우리가 알아내고자 하는 모수인 회귀계수 (β_i)들과 σ^2를 확률변수 취급하는 것입니다.

$$\{y_i\}_{i=1}^n | \beta_0, \beta_1, \sigma^2 \sim^{iid} N(\beta_0 + \beta_1 x_i, \sigma^2)$$

$$\beta_0 \sim N(\mu, \tau)$$

$$\beta_1 \sim N(\mu, \tau)$$

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

- Posterior는 Prior와 Likelihood의 곱에 비례하므로, $\pi(\beta_0, \beta_1, \sigma^2 | X, Y) \propto \prod_{i=1}^n L(\beta_0, \beta_1, \sigma^2 | X_i, Y_i) \times p(\beta_0)p(\beta_1)p(\sigma^2)$.
- Likelihood: Normal, β Prior: Normal, σ^2 Prior: Inverse-Gamma
- 모든 parameter는 서로 독립이라고 가정합니다.

Bayesian Regression vs. Bayesian GLM

- Bayesian GLM 역시 회귀계수에 확률분포를 가정합니다.

$$\pi(\beta_0, \beta_1 | X, Y) \propto L(\beta_0, \beta_1 | X, Y) p(\beta_0) p(\beta_1)$$

- Bayesian Regression과의 차이점은 Likelihood입니다!
- 일반 선형회귀에서 $Y \sim N(E[Y|X], \sigma^2)$이지만, GLM에서는 Y가 더이상 실수 전체를 범위로 가지지 않으므로, 정규분포가 아닌 다른 분포를 따르게 됩니다.
- 즉, Y의 분포가 바뀐다는 것인데 이는 Bayesian Regression에서 Likelihood 부분의 함수식에 영향을 미칩니다.

Bayesian GLM Likelihood Setting

- Logistic Regression ($Y \sim$ Bernoulli$(\mu(X))$):

$$L(\beta_0, \beta_1 | X, Y) = \prod_{i=1}^{n} \mu(X_i)^{Y_i}(1-\mu(X_i))^{1-Y_i}, \;\; \mu(X_i) = \frac{\exp(X_i^T \beta)}{1+\exp(X_i^T \beta)}$$

$$(X_i = (1, x_{1i}), \;\; \beta = (\beta_0, \beta_1))$$

- Poisson Regression ($Y \sim$ Pois$(\mu(X))$):

$$L(\beta_0, \beta_1 | X, Y) = \prod_{i=1}^{n} \frac{\mu(X_i)^{Y_i} \exp(-\mu(X_i))}{Y_i!}, \;\; \mu(X_i) = \exp(X_i^T \beta)$$

- 추정하는 회귀계수의 개수에 따라 X_i와 β의 차원은 달라진다. ($X_i, \beta \in \mathbb{R}^p$)

Example] Bayesian Statistics HW3

In this assignment, we will implement MCMC algorithm for a Bayesian GLM as follows:

$$\mathbf{Y} \sim \text{Poisson}(\mu(\mathbf{X}))$$
$$\log(\mu(\mathbf{X})) = \mathbf{X}\beta$$
$$\beta_j \sim N(0, 10) \text{ for } j = 1, 2, 3, 4$$

1) Simulate the dataset as follows.
(a) Let $X_i \in \mathbb{R}^4$ be the predictors for ith observation. For $i = 1, ..., 1000$, simulate $X_i \sim N(0, \mathbf{I})$ independently. Here $\mathbf{I}$ is an identity matrix.
(b) For $i = 1, ..., 1000$, simulate $Y_i \sim \text{Poisson}(\exp(X_i^T \beta))$ independetly. Set the true regression coefficient value as $\beta = (0.5, -0.5, 0, 1)$.

2) Implement the MCMC algorithm using the simulated dataset in Problem 1). Here you should write down the code without using any packages. Report the trace plots, density plots, 95% HPD intervals, posterior mean, acceptance probability, and effective sample size for all parameters.

Example] Bayesian Statistics HW3 (1)

1) Simulate the dataset as follows.
(a) Let $X_i \in \mathbb{R}^4$ be the predictors for ith observation. For $i = 1, ..., 1000$, simulate $X_i \sim N(0, \mathbf{I})$ independently. Here $\mathbf{I}$ is an identity matrix.

X_i follows multivariate normal distribution. However, its covariance matrix is an identity matrix. So, 1000 iid samples from $N_4(0, \mathbf{I})$ is equivalent with 4000 iid samples from univariate standard normal distribution.

Example] Bayesian Statistics HW3 (2)

1) Simulate the dataset as follows.
(b) For $i = 1, ..., 1000$, simulate $Y_i \sim$ Poisson$(\exp(X_i^T \beta))$ independetly.
Set the true regression coefficient value as $\beta = (0.5, -0.5, 0, 1)$.

Now, you have 1000×4 data matrix X. Let $\beta = (0.5, -0.5, 0, 1)$.
Multiply them, and we can get $X^T\beta$ matrix (or vector), which has a size of 1000×1. Then you can also easily get $\exp(X_i^T \beta)$.

Let $M = \exp(X^T \beta)$, then this means
$Y_1 \sim$ Pois(M_1), $Y_2 \sim$ Pois(M_2), $\cdots$, $Y_{1000} \sim$ Pois(M_{1000}).
You can use `for` loop and `rpois` function in order to simulate $Y_1 \sim Y_{1000}$.

Example] Bayesian Statistics HW3 (3)

$$\mathbf{Y} \sim \text{Poisson}(\mu(\mathbf{X}))$$

$$\log(\mu(\mathbf{X})) = \mathbf{X}\beta$$

$$\beta_j \sim N(0, 10) \text{ for } j = 1, 2, 3, 4$$

2) Implement the MCMC algorithm using the simulated dataset.

Our goal is to find the posterior distribution of $\beta_1 \sim \beta_4$.
Let $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$. Now, the posterior distribution is,

$$\pi(\beta_1, \beta_2, \beta_3, \beta_4 | X, Y) \propto L(\beta_1, \beta_2, \beta_3, \beta_4 | X, Y) p(\beta_1) p(\beta_2) p(\beta_3) p(\beta_4)$$

$$\text{where } L(\beta) = \prod_{i=1}^{n} \frac{\mu(X_i)^{Y_i} \exp(-\mu(X_i))}{Y_i!}, \ \mu(X_i) = \exp(X_i^T \beta)$$

We already have X, Y and β, so we can calculate the value.

Example] Bayesian Statistics HW3 (4)

Now, let's use the Metropolis-Hastings algorithm to find the posterior distribution of $\beta_1 \sim \beta_4$.
One-by-one update and All-at-once update are both possible, but I'm going to update regression coefficients one-by-one (in order to control the acceptance probability more easily).
After setting the initial value, $\beta_1^{(0)}, \beta_2^{(0)}, \beta_3^{(0)}, \beta_4^{(0)}$, update each regression coefficient as follows:

1) Generate $\beta_1' \sim N(\beta_1^{(t)}, \sigma_1^2)$. (Normal proposal)
2) Find $\log(\alpha) = \log(\pi(\beta_1', \beta_2^{(t)}, \beta_3^{(t)}, \beta_4^{(t)})) - \log(\pi(\beta_1^{(t)}, \beta_2^{(t)}, \beta_3^{(t)}, \beta_4^{(t)}))$.
Here, you can ignore the prior of $\beta_2, \beta_3, \beta_4$. (kernel)
3) Generate $U \sim U(0,1)$, and compare $\log(U)$ and $\log(\alpha)$.
4) If $\log(U) < \log(\alpha)$, $\beta_1^{(t+1)} = \beta_1'$. Else, $\beta_1^{(t+1)} = \beta_1^{(t)}$.
5) Repeat 1) $\sim$ 4) for β_2, β_3, and β_4.
6) Repeat 1) $\sim$ 5) for some sufficient number of iterations.

Gibbs Sampler

- Gibbs Sampler는 특수한 형태의 Metropolis-Hastings algorithm입니다.
- 우리의 target distribution이 multivariate이면서, **fully conditional distribution**을 알아낼 수 있다면 사용할 수 있는 방법입니다.
- 즉, 앞서 예시에서처럼 target distribution π가, $\pi(\beta_1, \beta_2, \beta_3, \beta_4|X, Y)$처럼 다변량 함수로 주어져 있을 때, 그것들의 fully conditional distribution인 $\pi(\beta_1|\beta_2, \beta_3, \beta_4, X, Y)$, $\pi(\beta_2|\beta_1, \beta_3, \beta_4, X, Y)$, $\pi(\beta_3|\beta_1, \beta_2, \beta_4, X, Y)$, $\pi(\beta_4|\beta_1, \beta_2, \beta_3, X, Y)$을 모두 찾아낼 수 있다면 이 함수들을 이용해 Gibbs sampler를 사용할 수 있습니다.
- Gibbs sampler의 장점은, acceptance rate가 **1**이라는 점입니다.
- 따라서 Gibbs sampler에서는 threshold value를 계산하는 단계나, 해당 값과 U를 비교해 proposed value(x')를 사용할지 말지를 결정하는 단계도 필요가 없습니다.
- Gibbs sampler를 Gibbs sampling이라고 하기도 합니다.

Gibbs Sampler Algorithm

1. Initialize $t = 0$ and $\mathbf{X}_0 = (X_1^{(0)}, ..., X_d^{(0)})$.
2. Sample in turn:

$$X_1^{(t+1)} \sim f_{1|-1}(x_1 | X_2^{(t)}, X_3^{(t)}, ..., X_d^{(t)})$$

$$X_2^{(t+1)} \sim f_{2|-2}(x_2 | X_1^{(t+1)}, X_3^{(t)}, ..., X_d^{(t)})$$

$$...$$

$$X_d^{(t+1)} \sim f_{d|-d}(x_d | X_1^{(t+1)}, X_2^{(t+1)}, ..., X_{d-1}^{(t+1)})$$

3. Set $\mathbf{X}_{t+1} = (X_1^{(t+1)}, ..., X_d^{(t+1)})$.
4. Set $t = t + 1$ and go to step 2.

Gibbs sampler의 acceptance rate가 항상 1인 이유

$$\text{Let } x_o = (x_1, ..., x_i, ..., x_d), x' = (x_1, ..., x', ..., x_d),$$

$$\text{and } q(x'|x_o) = \frac{1}{d} f_{i|-i}(x'|x_j, j \neq i) = \frac{1}{d} \frac{f(x')}{f(x_1, ..., x_{i-1}, x_{i+1}, ..., x_d)}.$$

$$\text{Then, } \frac{f(x')q(x_o|x')}{f(x_o)q(x'|x_o)} = \frac{f(x')\frac{1}{d}\frac{f(x_o)}{f(x_1,...,x_{i-1},x_{i+1},...,x_d)}}{f(x_o)\frac{1}{d}\frac{f(x')}{f(x_1,...,x_{i-1},x_{i+1},...,x_d)}} = \frac{f(x')f(x_o)}{f(x_o)f(x')} = 1.$$

$$\text{Therefore, } \alpha = \min(\frac{f(x')q(x_o|x')}{f(x_o)q(x'|x_o)}, 1) = \min(1, 1) = 1.$$

위의 증명과정을 통해 Gibbs sampler는 Metropolis-Hastings algorithm의 특수한 경우라는 것과, Gibbs sampler는 항상 acceptance rate가 1임을 알 수 있습니다.