STA3105-01 Bayesian Statistics
Homework 5
DUE Friday, November 18

Copying homework solutions from others lead to a 0 score. No late submission is allowed. Your solution should contain both code and a corresponding explanation for the answer. Submit your HW through LearnUs. You should submit (1) a report file (pdf) and (2) relevant code files. We will implement MCMC algorithm for a Bayesian logistic regression as follows:

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

$$\log\left(\frac{\mu(\mathbf{X})}{1-\mu(\mathbf{X})}\right) = \mathbf{X}\beta$$

$$\beta_j \sim N(0,10) \; for \; j = 1, \cdots, 4$$

1. (10 points) Simulate the dataset as follows.

   (a) Let $\mathbf{X}_i \in \mathbb{R}^4$ be the predictors for ith observation. For $i = 1, \cdots, n$, simulate $\mathbf{X}_i \sim N(0, \mathbf{I})$ independently. Here $n = 300,000$ and $\mathbf{I}$ is an identity matrix.

   (b) For $i = 1, \cdots, n$, simulate $Y_i \sim \text{Bernoulli}(\mu(\mathbf{X}_i))$ independently. Set the true regression coefficient value as $\beta = (0.5, -0.5, 0, 1)$.

2. (90 points) Implement the MCMC algorithm for the simulated dataset in Problem 1. Here, you should write down the `C++` code using `RcppArmadillo` library as follows.

   (a) Write down the `Rcpp` function for evaluating joint likelihood function for given $\beta_1, \beta_2, \beta_3, \beta_4$, $\mathbf{Y}, \mathbf{X}$. Compare the log-likelihood values calculated from your `Rcpp` function with `dbinom` function in `R`.

   (b) Write down the `Rcpp` function for evaluating prior for given $\beta_1, \beta_2, \beta_3, \beta_4$.

   (c) Using the functions defined in (a),(b), write down the `Rcpp` function to generate posterior samples from the above specified model.

   (d) Report the trace plots, density plots, 95% HPD intervals, posterior mean, acceptance probability, and effective sample size for all parameters. Check whether your MCMC samples can recover the true $\beta = (0.5, -0.5, 0, 1)$ well.

3. (Extra credit: 20 points) Now implement the divide-and-conquer MCMC algorithm to the simulated dataset. Here, you should write down the `C++` code using `RcppArmadillo` and `OpenMp` libraries as follows.

   (a) Divide the entire dataset into S shards; S should be set to the maximum number of cores available on your personal computer. Report the available number of cores using `detectCores` function in `parallel` package.

   (b) Run S separate MCMC to generate subset posterior samples for each shard. This step should be implemented in parallel via `OpenMp` library.

(c) Combine subset posterior samples using weighted average as suggested in the consensus Monte Carlo algorithm.

Report the followings:

(a) Report the user time and elapsed time that can be checked through `proc.time` function in `R`.

(b) For each subset posterior, report the trace plots, density plots, 95% HPD intervals, posterior mean, acceptance probability, and effective sample size for all parameters.

(c) Draw the density plot of S subset posteriors and the combined posterior for all parameters. You can overlap all the densities with different colors.