STAT 992: Advanced MCMC

Final project information

10 September 2024

Overview

The final project is an opportunity for you to reflect on what we learnt this semester and to demonstrate an ability to derive and implement several of the Markov chain Monte Carlo sampling strategies discussed in class. You should work on the final project individually in groups of up to four (i.e., you plus three fellow classmates). The project requires you to derive, implement, and compare several samplers for the test problems listed below.

Deliverables

Final report

For each test problem, you should provide a full derivation of each sampler and compare their performances both qualitatively and quantatitively. The comparisons could be based on diagnostics like effective sample size (ESS), ESS per iteration or unit time, runtime, $\hat{R}$, or otherwise. Your report must be typeset using the provided template. The exposition comparing sampler performance should be written as if you were writing a research paper. In a sense, this project mimics what you would do in a paper introducing a new sampler. Specifically, you would derive the sampler in sufficient technical details and compare it to existing ones on synthetic data.

Code to reproduce experiments

You will additionally submit code to run your experiments. This code should be self-contained and must be able to be run with minimal modification.

Grading & rubric

The final project is worth a total of 900 points towards your final grade. These points will be distributed equally between three parts: (i) correctness of the sampler derivations and implementations; (ii) design and implementation of the comparisons; and (iii) quality of the written report.

Test problems

Model selection

Suppose we have p covariates $X_1, \dots, X_p$ and an outcome Y. For any subset $S \subseteq \{1, 2, \dots, p\}$, consider the linear model with known variance involving only those variables whose indices are in S:

$$Y | \boldsymbol{x}_S, \beta^{(S)}, S \sim \mathcal{N}\left(\sum_{j \in S} \beta_j^{(S)} x_j, 1\right)$$
$$\beta^{(S)} \sim \mathcal{N}_{|S|}\left(\mathbf{0}_{|S|}, \mathrm{I}_{|S| \times |S|}\right)$$

In total, there are 2^p possible regression models (one for each subset S of covariates), each with its own parameter $\beta^{(S)}$. We wish to perform inference on the pair $(S, \beta^{(S)})$. To this end, we specify a uniform prior on S, so that $p(S) = 2^{-p}$ for all subsets $S \subseteq \{1, 2, \dots, p\}$.

For this test problem, you should

1. Design and implement an algorithm that returns samples of $(S, \beta^{(S)})$. You should explicitly describe the cross-model moves (i.e., those that move between $(S, \beta^{(S)})$ and $(S', \beta^{(S')})$ where S, S' are distinct subsets of $\{1, 2, \dots, p\}$.)
2. Deploy your sampler on synthetic datasets with $p = 5$ and $p = 10$. Report the approximate posterior model probabilities (i.e., the probabilities $p(S|\boldsymbol{y})$). Compare these to the actual probabilities, which you can compute explicitly for small p.

MALA for standard normals

Derive and implement MALA for generating samples from the standard multivariate normal distribution $\mathcal{N}_d\left(\mathbf{0}_d, \mathrm{I}_{d \times d}\right)$. You should investigate how the performance of the algorithm changes as you vary the step-size and increase the dimensionality. There are many ways to assess performance including, but not limited to,

- Examining how effective sample size grows as the number of MCMC iterations increases
- Examining the bias of estimating the first and second moments of the distribution, which are known in advance

MALA for Bayesian logistic regression

Consider the p-dimensional Bayesian logistic regression problem

$$Y | \beta \sim \beta \sim \mathrm{Bernoulli}\left(\left[1 + e^{-\boldsymbol{x}^\top \beta}\right]^{-1}\right)$$
$$\beta \sim \mathcal{N}_p\left(\mathbf{0}_p, \mathrm{I}_{p \times p}\right),$$

where the covariates $\boldsymbol{x}$ have been centered and scaled to have standard deviation 1/2.

In this problem, you will compare MALA (run with different step-sizes) to HMC, as implemented in Stan. You should investigate how these algorithms' performance vary as p increases. Specifically you should:

1. Create a synthetic dataset from the model
2. Run MALA for various step-sizes and implement a sampler in Stan
3. Compare the samplers in terms of effect sample size, run time, and the bias of the posterior mean of β