MCMC.jl

MCMC algorithms in Julia

0 Contributors

0 Issues

0 Stars

MCMC Methods in Julia

Project Resources

Vansh Bansal
Pramodh Gopalan

Week 1

1. MSO201 Refresher:
    - Notes
2. Introduction to Monte Carlo:
    - Notes
    - Video

Week 2

1. Pseudo-Random Number Generators:
    - Notes
    - Video
2. Discrete Inverse Transform:
    - Notes
    - Video
3. Discrete Accept Reject:
    - Notes
    - Video
4. Composition Methods and Bernoulli Factories:
    - Notes
    - Video

Assignment 1

Due By: 11:59 pm, 19/04/2022

General Instructions:

1. Submit only one jupyter notebook.
2. For the theoretical questions, use the markdown text blocks of jupyter notebook., You may use LaTeX in them to type math equations (as and when required). Hand-written solutions will NOT be accepted.
3. For the programming questions, clearly state the details of your algorithm in the text blocks.
4. Do NOT forget to cite any reference source you use while solving the assignment.

1. Implement a Discrete Inverse Transform sampler for Poisson distribution.

2. Implement a Discrete Accept Reject Sampler to simulate draws from Binomial(n, p) using a Poisson proposal.

   Specifications and Hints:

   - n and p are user-given inputs. Considering both the conditions for an appropriate proposal, choose the right parametrization of the proposal distribution.
   - We not only expect the code for the algorithm, but we want you to verify the quality and correctness of your samples by plotting the histogram and the actual density plots also. (You can use the default functions for plotting.)
   - Also, see if the mean and variance of your samples match the density mean and variance.
   - A relevant paper that might be useful: (Section 2.4) https://www.jstor.org/stable/2286346

Week 3

1. Continuous Inverse Transform:
    - Notes
    - Video
2. Continuous Accept Reject:
    - Notes
    - Video
3. Continuous Accept Reject Examples:
    - Notes
    - Video
4. Miscellaneous Methods in Sampling:
    - Notes
    - Video

Assignment 2

Due By: 11:59 pm, 10/05/2022

General Instructions:

1. Submit only one jupyter notebook.
2. For the theoretical questions, use the markdown text blocks of jupyter notebook., You may use LaTeX in them to type math equations (as and when required). Hand-written solutions will NOT be accepted.
3. For the programming questions, clearly state the details of your algorithm in the text blocks.
4. Do NOT forget to cite any reference source you use while solving the assignment.

   PS: This assignment is slightly big. This is done deliberately as this week's concepts form the basis of the motivation of this project, i.e., Markov Chain Monte Carlo, and this assignment will help you get a deeper insight into them.

1. Prove that the given algorithm generates a Bern$\left(\frac{c_y p_y}{c_x p_x + c_y p_y}\right)$ event.

   Also, find the probability distribution of the number of iterations it takes to give an output.

   1: Draw $C_1 \sim \text{Bern}\left(\frac{c_y}{c_y + c_x}\right)$
   2: **if** $C_1 = 1$ **then**
   3:     Draw $C_2 \sim \text{Bern}(p_y)$
   4:     **if** $C_2 = 1$ **then**
   5:         output 1
   6:     **if** $C_2 = 0$ **then**
   7:         go to Step 1
   8: **if** $C_1 = 0$ **then**
   9:     Draw $C_2 \sim \text{Bern}(p_x)$
   10:     **if** $C_2 = 1$ **then**
   11:         output 0
   12:     **if** $C_2 = 0$ **then**
   13:         go to Step 1

2. Sample uniformly from a p-dimensional sphere (a circle is $p = 2$). Consider a p-vector $x = (x_1, x_2, \ldots, x_p)$ and let $||.||$ denote the Euclidean norm. The pdf of this distribution is:

$$f(\mathbf{x}) = \frac{\Gamma\left(\frac{p}{2}+1\right)}{\pi^{p/2}} I\{\|\mathbf{x}\| \le 1\}$$

   Use a uniform p-dimensional hypercube to sample uniformly from this sphere. Implement this for $p = 2, 3, 4, 5, and\ 6$.
   What happens as p increases?

   Specifications and Hints:

   - We not only expect the code for the algorithm, but we want you to verify the quality and correctness of your samples by plotting them. (at least, for p=2).
     (You can use the default functions for plotting.)

- As already discussed, this is one of the motivating examples for developing/learning better sampling techniques.

3. Using accept-reject and a standard normal proposal, obtain samples from a truncated standard normal distribution with pdf:

$$f(x) = \frac{1}{\Phi(a) - \Phi(-a)} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} I(-a < x < a)$$

where Φ(·) is the CDF of a standard normal distribution. Run for $a = 4$ and $a = 1$.
What are the differences between the two settings?
<u>Specifications and Hints</u>:
- We not only expect the code for the algorithm, but we want you to verify the quality and correctness of your samples by comparing their mean and variance with the true ones.

Week 4

1. Simple Importance Sampling
   - Notes
   - Video
2. Optimal Simple Importance Sampling'
   - Notes
   - Video
3. Weighted Importance Sampling
   - Notes
   - Video
4. Maximum Likelihood Estimation
   - Notes
   - Video
5. Introduction to Bayesian Models
   - Notes
   - Video
6. Accept-Reject for Bayesian Models
   - Notes
   - Video

Assignment 3 Due By: 11:59 pm, 20/05/2022

General Instructions:
1. Submit only one jupyter notebook.
2. For the theoretical questions, use the markdown text blocks of jupyter notebook., You may use LaTeX in them to type math equations (as and when required). Hand-written solutions will NOT be accepted.
3. For the programming questions, clearly state the details of your algorithm in the text blocks.
4. Do NOT forget to cite any reference source you use while solving the assignment.

1. Consider estimating the mean of a standard Cauchy distribution using importance sampling with a normal proposal distribution. Does the estimator have finite variance?

2. Considering a target density $f(x)$ and an importance proposal $g(x)$. Suppose

$$\sup_x \frac{f(x)}{g(x)} < \infty$$

   a. Then we know that the simple importance estimator has a finite variance. Does the weighted importance estimator also have finite variance?
   b. To estimate the mean of the target density, is there any benefit to using importance sampling over accept-reject sampling?

3. For some known $y_i \in \mathbb{R}$, $i = 1, 2, \ldots, n$ and some $\nu > 2$, suppose the target density is

$$\pi(x) \propto e^{-x^2/2} \prod_{i=1}^{n} \left(1 + \frac{(y_i - x)^2}{\nu}\right)^{-(\nu+1)/2}$$

   To generate y's use the following specifications:
   - Set the seed as 1
   - $n = 50$
   - $\nu = 5$
   - Use the 'Distributions' package in Julia to randomly sample n values from a T-distribution with ν degrees of freedom. The sampled vector is y.

   Implement an importance sampling estimator with a $N(0, 1)$ proposal to estimate the first moment of this distribution. Does the weighted importance sampling estimator have finite variance? What happens if $\nu = 1$ and $\nu = 2$?

4. Suppose $Y_1, Y_2, \ldots, Y_n \mid \lambda \sim Poisson(\lambda)$ (iid draws) and the prior $\lambda \sim Gamma(\alpha, \beta)$, where α, β are known. Find the posterior distribution of λ .

Week 5

Notes:

- Notes 1 (by Prof. Dootika Vats, IIT Kanpur)
  Concise, to-the-point
  Follow these if you want to focus only on MCMC
- Notes 2 (by Charles J. Geyer)
  Elaborate, discusses concepts in detail with examples
  Follow for reference

1. Why MCMC?
   - Lecture 1 from Notes 1
2. Measure Theory
   - Lecture 2 from Notes 1
   - Video
3. Markov Chain Basics
   - Lecture 3 from Notes 1
   - Video
4. Metropolis-Hastings Algorithm
   - Lecture 4 from Notes 1
   - Video (Part 1) Video (Part 2) Video (Part 3)
   - Notes (with examples)
   - Video
   - Notes (more examples)
   - Video

Assignment 4

Due By: 11:59 pm, 31/05/2022

General Instructions:

1. Submit only one jupyter notebook.
2. For the theoretical questions, use the markdown text blocks of jupyter notebook., You may use LaTeX in them to type math equations (as and when required). Hand-written solutions will NOT be accepted.
3. For the programming questions, clearly state the details of your algorithm in the text blocks.
4. Do NOT forget to cite any reference source you use while solving the assignment.

1. Suppose $X_1, X_2, \cdots, X_n \sim F$ (iid draws) and $Y_1, Y_2, \cdots, Y_n$ is a Markov chain with F as the stationary distribution. Consider two estimates of the mean:
$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \text{and} \qquad \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$
Which estimator is better? In other words, which estimator has smaller variance? Support your answer with formal mathematical arguments.

2. Suppose $Y_1, Y_2, ..., Y_n | \mu \sim N(\mu, 1)$. Assume the prior on $\mu \sim t_\nu$, where ν is the degrees of freedom. Assume the prior on $\nu \sim Truncated\ Gamma(a_0, b_0, (2, \infty))$, where $(2, \infty)$ is the support of $Truncated\ Gamma$.
    a. What is the joint posterior distribution of (μ, ν)?
    b. Write an MH algorithm to sample from the above posterior distribution.
    c. Generate $n = 100$ data points. Set $a_0 = 2,\ b_0 = 0.1$, and run the MH algorithm described above.

3. Consider the Bayesian linear regression model. The likelihood is
$y_1, y_2, \cdots, y_n | \beta, \sigma^2 \sim N_n(X\beta, \sigma^2 I_n)$
The parameters of interest are β, σ^2 , just like regular MLE. We assume priors:
$\beta \sim N_p(\mu, \sigma^2 I_p)$ and $\sigma^2 \sim Inverse\ Gamma(a, b)$.
N_p is the p-dimensional multivariate normal distribution and the density of $Inverse\ Gamma(a, b)$ is given by:
$$\pi(\sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{-a+1} e^{-b/\sigma^2}$$
Here a, b and μ are hyper-parameters, which need to be chosen according to the dataset.
    a. What is the posterior distribution of (β, σ^2)?
    b. Implement an MH algorithm to sample from the posterior distribution for any suitable dataset (for example, Boston-housing dataset, etc.)

Week 6

Notes:

- Notes 1 (by Prof. Dootika Vats, IIT Kanpur)
  Concise, to-the-point
  Follow these if you want to focus only on MCMC
- Notes 2 (by Charles J. Geyer)
  Elaborate, discusses concepts in detail with examples
  Follow for reference

1. Stochastic Stability and Types of MH Algorithms
   - Lecture 5 from Notes 1
   - Video (Part 1) Video (Part 2)
2. General Accept-Reject MCMC and Combining Kernels
   - Lecture 6 from Notes 1
   - Video (Part 1) Video (Part 2)
3. Components-wise Updates and Gibbs Sampler
   - Lecture 7 from Notes 1
   - Video (Part 1) Video (Part 2)
4. Gibbs Sampler Examples
   - Lecture 8 from Notes 1
   - Video (Part 1) Video (Part 2)

Assignment 5

Due By: 11:59 pm, 10/06/2022

General Instructions:

1. Submit only one jupyter notebook.
2. For the theoretical questions, use the markdown text blocks of the jupyter notebook. You may use LaTeX in them to type math equations (as and when required). Hand-written solutions will NOT be accepted.
3. For the programming questions, clearly state the details of your algorithm in the text blocks.
4. Do NOT forget to cite any reference source you use while solving the assignment.

Research Paper: Stochastic Gradient Markov Chain Monte Carlo

While MCMC algorithms have the advantage of providing asymptotically exact posterior samples, this comes at the expense of being computationally slow to apply in practice. This issue is further exacerbated by the demand to store and analyze large-scale datasets and to fit increasingly sophisticated and complex models to these high-dimensional data.
If the data can be split across multiple computer cores then the computational challenge of inference can be parallelized, with an MCMC algorithm run on each core to draw samples from a partial posterior that is conditional on only a subset of the full data. The challenge is then to merge these posterior samples from each computer to generate an approximation to the full posterior distribution, but it is hard to quantify the level of approximation accuracy such merging procedures have in general.
Alternatively, rather than using multiple computer cores, a single MCMC algorithm can be used, where only a subsample of the data is evaluated at each iteration. Perhaps the most general and popular class of scalable, subsampling-based algorithms are stochastic-gradient MCMC (SGMCMC) methods. These algorithms are derived from diffusion processes which admit the posterior as their invariant distribution.

1. Summarize sections 2 and 3 in your own words (with all mathematical details and rigor). Do NOT copy. Write in your own words.
2. Implement the SG-MCMC algorithm for the Bayesian Neural Network detailed in Section 6.2.
   If your laptop configurations do not allow training on the complete dataset, train on a smaller subset of the dataset and for fewer iterations. But bonus points, if you can train properly.

Assignment 5

Due By: 11:59 pm, 25/06/2022

General Instructions:

1. Submit only one jupyter notebook.
2. For the theoretical questions, use the markdown text blocks of the jupyter notebook. You may use LaTeX in them to type math equations (as and when required). Hand-written solutions will NOT be accepted.
3. For the programming questions, clearly state the details of your algorithm in the text blocks.
4. Do NOT forget to cite any reference source you use while solving the assignment.

Research paper: Efficient Bernoulli factory MCMC for intractable posteriors

Accept-reject based Markov chain Monte Carlo (MCMC) algorithms have traditionally utilised acceptance probabilities that can be explicitly written as a function of the ratio of the target density at the two contested points. This feature is rendered almost useless in Bayesian posteriors with unknown functional forms. We introduce a new family of MCMC acceptance probabilities that has the distinguishing feature of not being a function of the ratio of the target density at the two points. We present two stable Bernoulli factories that generate events within this class of acceptance probabilities. The efficiency of our methods rely on obtaining reasonable local upper or lower bounds on the target density and we present two classes of problems where such bounds are viable: Bayesian inference for diffusions and MCMC on constrained spaces. The resulting portkey Barker's algorithms are exact and computationally more efficient that the current state-of-the-art.

1. Summarize sections 2, 3 and 4 in your own words (with all mathematical details and rigor). Do NOT copy. Write in your own words.
2. Implement the example 5.1, play with the parameter β and create different plots to judge the quality of your samples