Gradient-Free Optimal Postprocessing of MCMC Output

Artem Glebov

King's College London

2024

Overview

Problem

Develop a computationally efficient algorithm for summarising the output of a Markov Chain Monte Carlo simulation.

Motivation

Uncertainty quantification in a multi-stage simulation of the functioning of the human heart.

Existing solution

The optimisation algorithm of Riabiz et al. (2022) to select a subsample of MCMC output that minimises a measure of proximity to the target distribution (kernel Stein discrepancy), which requires the gradients of the log-posterior and is thus expensive.

Proposal

Modify the algorithm of Riabiz et al. (2022) to use the gradient-free kernel Stein discrepancy of Fisher and Oates (2024).

Table of Contents

Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) are a popular class of algorithms for sampling from complex probability distributions.
Given a target distribution P defined on a state space $\mathcal{X}$, an MCMC algorithm proceeds by constructing a chain of random variables $(X_i)_{i=0}^{\infty}$ which satisfy the Markov property:

$$\mathbb{P}(X_{i+1} \in A | X_0, \ldots, X_i) = \mathbb{P}(X_{i+1} \in A | X_i) \quad \text{for any measurable } A \in \mathcal{X}.$$

Viewed as a function, the right-hand side above is called the Markov transition kernel and is denoted

$$R(A|x) := \mathbb{P}(X_{i+1} \in A | X_i = x). \tag{1}$$

The transition kernel R is selected so that it is easy to sample from and to ensure asymptotic convergence to the target distribution P:

$$P_i \xrightarrow{d} P \quad \text{as} \quad i \to \infty.$$

A sample of size n is a realisation $(x_i)_{i=0}^{n}$ of the first n variables in the chain, which is constructed sequentially.

Challenges of running MCMC

1. The choice of a starting point for a chain.
2. Exploring the modes of a multimodal distribution.
3. Calibrating the scale of the proposal distribution.
4. Convergence detection.
5. Detecting and eliminating the burn-in.
6. Autocorrelation between samples in a chain.
7. Compressing sample for further expensive processing.

Table of Contents

Algorithm 1: Gradient-free Stein thinning.

Data:

sample $(x_i)_{i=1}^n$ from MCMC,

target log-densities $(\log p(x_i))_{i=1}^n$

auxiliary log-densities $(\log q(x_i))_{i=1}^n$

auxiliary gradients $(\nabla \log q(x_i))_{i=1}^n$

desired cardinality $m \in \mathbb{N}$

Result: Indices π of a sequence $(x_{\pi(j)})_{j=1}^m$ where $\pi(j) \in \{1, \ldots, n\}$.

for $j = 1, \ldots, m$ **do**

$$\pi(j) \in \underset{i=1,\ldots,n}{\arg\min} \frac{k_{P,Q}(x_i, x_i)}{2} + \sum_{j'=1}^{j-1} k_{P,Q}(x_{\pi(j')}, x_i)$$

end

Algorithm 2: Optimised gradient-free Stein thinning.

Data:

sample $(x_i)_{i=1}^n$ from MCMC,

target log-densities $(\log p(x_i))_{i=1}^n$

auxiliary log-densities $(\log q(x_i))_{i=1}^n$

auxiliary gradients $(\nabla \log q(x_i))_{i=1}^n$

desired cardinality $m \in \mathbb{N}$.

Result: Indices π of a sequence $(x_{\pi(j)})_{j=1}^m$ where $\pi(j) \in \{1, \ldots, n\}$.

Initialise an array $A[i]$ of size n

Set $A[i] = k_{P,Q}(x_i, x_i)$ for $i = 1, \ldots, n$

Set $\pi(1) = \arg\min_i A[i]$

for $j = 2, \ldots, m$ **do**

Update $A[i] = A[i] + 2k_{P,Q}(x_{\pi(j-1)}, x_i)$ for $i = 1, \ldots, n$

Set $\pi(j) = \arg\min_i A[i]$

end

Table of Contents

Table of Contents

Contribution

The project makes three contributions:

- implementation of the gradient-free Stein thinning algorithm in the Python library `stein-thinning`,
- evaluation of the performance of the proposed algorithm,
- improvement of the computational efficiency of the existing Stein thinning algorithm from $O(nm^2)$ to $O(nm)$, where n is the input sample size and m is the desired thinned sample size.

Conclusions

- The gradient-free approach is feasible and performs similarly to the Stein thinning algorithm of Riabiz et al. (2022) for small thinned sample sizes,
- The performance of the algorithm depends crucially on the choice of the auxiliary distribution. For example, even in the highly favourable setting of i.i.d. samples from a Gaussian mixture, choosing the auxiliary distribution based on the Laplace approximation fails to produce a thinned sample.
- The simple multivariate Gaussian distribution using the sample mean and covariance offered a good starting point in our experiments, however bespoke treatment might be required for more complex problems.
- In deciding whether to use the new algorithm as opposed to the gradient-based approach, the effort involved in selecting a good auxiliary distribution must be weighed against the computational cost of obtaining gradients.

Table of Contents

Further Research

- Evaluate the choices of KDE kernels other than Gaussian for constructing the auxiliary distribution.
- Parallelise the computation of KDE.
- Perform thinning in a lower-dimensional space.
- Investigate the behaviour of Stein thinning for large thinned sample sizes.
- Compare the performance of the approaches in terms of estimating the true parameters of the Lotka-Volterra model.
- Run an experiment with randomised starting points.

Further Research (continued)

- Repeat the experiments with more advanced MCMC algorithms.
- Check how running a gradient-free MCMC sampling algorithm (such the random-walk Metropolis-Hastings) followed by Stein thinning of the sample compares to running a gradient-based sampling algorithm (e.g. HMC).
- Provide theoretical justification for gradient-free Stein thinning.
- Explore other gradient-free alternatives.

Table of Contents

Bibliography

M. A. Fisher and C. J. Oates. Gradient-Free Kernel Stein Discrepancy. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 23855–23885, May 2024.

M. Riabiz, W. Y. Chen, J. Cockayne, P. Swietach, S. A. Niederer, L. Mackey, and C. J. Oates. Optimal Thinning of MCMC Output. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1059–1081, September 2022.