Gradient-Free Optimal Postprocessing of MCMC Output

by

Artem Glebov

Department of Mathematics
King's College London
The Strand, London WC2R 2LS
United Kingdom

Abstract

Contents

Introduction

The Python code accompanying this report can be found at:

`https://github.com/aglebov/gradient-free-mcmc-postprocessing`.

The implementation of the gradient-free kernel Stein thinning was contributed by the author directly to the `stein-thinning` Python library:

`https://github.com/wilson-ye-chen/stein_thinning`.

Chapter 1

Background

1.1 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) are a popular class of algorithms for sampling from complex probability distributions.

The need to sample from a probability distribution arises in exploratory analysis as well as when analytical expressions are unavailable for quantities of interest, such as the modes or quantiles of the distribution, or for expectations with respect to the distribution, so a numerical simulation is used to obtain approximations instead. Such cases are frequent in Bayesian analysis, where the posterior density often has a complex structure with an analytically intractable normalising constant.

Describe alternatives: the inverse method, accept-reject and importance sampling

Include a simple motivating example

An MCMC algorithm proceeds by sequentially constructing a chain of samples[1] x_1, x_2, ..., where each sample is drawn from a transition distribution Q conditional on the preceding value:

$$x_{n+1} \sim Q(x_{n+1}|x_n).$$

The distribution Q is known as the transition kernel and is selected so that

[1]These are also sometimes called "draws". In this report, we follow the literature in using the term "sample" both for a single element in an MCMC chain and for all such elements taken together as a sample from the target distribution.

it is easy to sample from and to ensure asymptotic convergence to the target distribution Π:

$$x_n \xrightarrow{d} \Pi \quad \text{as} \quad n \to \infty.$$

Two classical variations of this technique are the Metropolis-Hastings and Gibbs algorithms.

Metropolis-Hastings algorithm. The algorithm due to Metropolis et al. (1953) and Hastings (1970) uses an auxiliary distribution q to sample a proposed value

$$x' \sim q(x'|x_n),$$

which is then accepted with probability

$$\alpha(x_n, x') = 1 \wedge \frac{\pi(x')}{\pi(x_n)} \frac{q(x_n|x')}{q(x'|x_n)}.$$

If x' is accepted, the algorithm sets $x_{n+1} = x'$. If x' is rejected, the value remains unchanged: $x_{n+1} = x_n$.

Consider using a different notation to avoid the confusion between the density of the proposal q and the transition kernel Q.

The common choice for the proposal distribution q is a symmetric proposal satisfying $q(x'|x_n) = q(x_n|x')$, so that the ratio of these two quantities disappears from the expression for the acceptance probability:

$$\alpha(x_n, x') = 1 \wedge \frac{\pi(x')}{\pi(x_n)}.$$

In the special case where $q(x'|x_n) = q(x' - x_n)$ we obtain a random walk proposal:

$$x' = x_n + Z,$$

where Z is the distribution of the step taken by the algorithm, e.g. a multivariate normal distribution. The operation of the algorithm then resembles a random walk across the domain of the target distribution where steps towards areas of lower probability are more likely to be rejected. The scale of the step distribution Z determines the average size of the jump that the algorithm can make in one iteration and thus the speed of traversal of the target domain.

An alternative to symmetric proposals is an independence proposal satisfying $q(x'|x_n) = q(x')$.

Cite the ST03 lecture notes or Robert & Casella

Gibbs algorithm. Suppose x is a d-dimensional vector and the components $x^{(1)}$, $x^{(2)}$, …, $x^{(d)}$ can be partitioned in such a way that we can sample the components belonging to each partition while keeping the components in other partitions fixed. That is, let $I_i \subset \{1, \dots, d\}$ with $\cup_{i=1}^{k} I_i = \{1, \dots, d\}$ for some k and $I_i \cap I_j = \emptyset$ for $i \neq j$, and assume we can sample

$$x^{(I_i)} \sim f_i\left(x^{(I_i)} | x^{(I_1, \dots, I_{i-1}, I_{i+1}, \dots, I_k)}\right).$$

The sample x_{n+1} can then be constructed by sequentially sampling for each partition:

$$x_{n+1}^{(I_i)} \sim f_i\left(x^{(I_i)} | x_{n+1}^{(I_1, \dots, I_{i-1})}, x_n^{(I_{i+1}, \dots, I_k)}\right).$$

Note that the newly sampled values $x_{n+1}^{(I_1, \dots, I_{i-1})}$ enter the computation for subsequent partitions.

Read and cite the original paper for Gibbs sampler

Consider simplifying this description

Mention HMC and other recent variations

1.2 Challenges of running MCMC

While the asymptotic convergence of MCMC samples to the target distribution is guaranteed, no general guarantee is available for finite samples, resulting in several interrelated challenges that a practitioner faces when applying this class of algorithms:

1. The choice of a starting point for a chain affects the speed of convergence to the target distribution.
2. For a multimodal distribution, the algorithm might struggle to move between the modes within a feasible time. This problem becomes especially acute in high dimensions.
3. The scale of the proposal distribution must be calibrated to ensure that the algorithm is able to explore the domain of the target distribution efficiently.
4. Assessing how close an MCMC chain is to convergence is difficult, since the knowledge about the target distribution often comes from the chain itself.

5. In order to eliminate the impact of the starting point, it can be useful to discard the initial iterations of an MCMC chain, which are considered as “burn-in”. Selecting the optimal length of the burn-in period is contingent on being able to detect convergence.

6. The sequential procedure of constructing a chain induces autocorrelation between the samples, which leads to increased variance of derived estimators.

7. The large number of samples resulting from an MCMC algorithm needs to be summarised for subsequent analysis, particularly when the cost of using all available samples is too high. Such situations arise when samples obtained from MCMC are used as starting points for further expensive simulations.

The first three challenges require decisions to be made upfront before running the algorithm or adaptively during its run. In order to address the impact of the starting point, running multiple chains with starting points sampled from an overdispersed distribution is recommended (Gelman and Rubin (1992)). This approach has the added benefit of increasing the chance of discovering the modes of the target distribution, although it does not provide a guarantee in this respect.

Mention perfect sampling.

Comparing the summary statistics of several chains (Gelman and Rubin (1992); Brooks and Gelman (1998); Vehtari et al. (2021)) offers a way to detect a lack of convergence at the cost of additional computation. Alternatively, the comparison can be applied to batches of samples from a single chain, as proposed by Vats and Knudson (2021). Convergence detection can be used to terminate the algorithm once a chosen criterion is satisfied, or to assess the quality of a sample retrospectively. It should be noted that convergence criteria establish a necessary but not sufficient condition for convergence, so the outcomes need to be interpreted accordingly.

The scaling of the step distribution in a random-walk Metropolis-Hastings algorithm is commonly tuned to target the acceptance rate of roughly 0.234 for proposed samples (Gelman et al. (1996, 1997); Roberts and Rosenthal (2001)), which balances the speed of traversal and the computational effort generating samples that end up rejected.

The last three challenges are typically addressed by post-processing a sample from a completed MCMC run. A recent proposal by Riabiz et al.

(2022) addresses these challenges by selecting a fixed-size subset of samples from an MCMC run such that the empirical distribution given by the subset best approximates the distribution resulting from the full sample. In the following section, we consider their approach in greater detail.

Read and cite Cowles and Carlin (1996) regarding the choice of burn-in length.

1.3 Optimal thinning

Given a Markov chain $(X_i)_{i\in\mathbb{N}}$ and its realisation of length n, the empirical approximation of the target posterior distribution is given by

$$\frac{1}{n}\sum_{i=1}^{n}\delta(X_i), \tag{1.1}$$

where $\delta(x)$ is the Dirac delta function. Riabiz et al. (2022) set out to identify $m \ll n$ indices $\pi(j) \in \{1,\ldots,n\}$ with $j \in \{1,\ldots,m\}$, such that the approximation provided by the subset of samples

$$\frac{1}{m}\sum_{j=1}^{m}\delta(X_{\pi(j)}) \tag{1.2}$$

is closest to the approximation given by the full set, and thus to the target distribution. The criterion of proximity is based on the kernel Stein discrepancy, itself a special case of the integral probability metric.

1.3.1 Kernel Stein discrepancy

The integral probability metric between two distributions P and P' is defined as

$$\mathcal{D}_{\mathcal{F}}(P,P') \coloneqq \sup_{f\in\mathcal{F}}\left|\int_{\mathcal{X}} f\,\mathrm{d}P - \int_{\mathcal{X}} f\,\mathrm{d}P'\right|, \tag{1.3}$$

where $\mathcal{X}$ is a measurable space on which both P and P' are defined and $\mathcal{F}$ is a set of test functions. Depending on the choice of $\mathcal{F}$, the definition (1.3) gives rise to different classes of probability metrics, including the well-known Kolmogorov distance, Wasserstein distance and total variation distance.

If P is taken to be the target distribution of an MCMC algorithm, evaluating (1.3) poses two practical challenges:

- the integral $\int_{\mathcal{X}} f \, \mathrm{d}P$ is often analytically intractable,
- the supremum requires a non-trivial optimisation procedure to find.

The need to integrate with respect to P can be eliminated if we find a set of function $\mathcal{F}$ for which $\int_{\mathcal{X}} f \, \mathrm{d}P = 0$ for all $f \in \mathcal{F}$. The expression (1.3) then simplifies to

$$\mathcal{D}_{\mathcal{F}}(P, P') = \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{X}} f \, \mathrm{d}P' \right|. \tag{1.4}$$

Gorham and Mackey (2015) propose choosing such a set $\mathcal{F}$ based on the observation that the infinitesimal generator

$$(\mathcal{L}u)(x) = \lim_{t \to 0} \frac{\mathbb{E}[u(Z_t) | Z_0 = x] - u(x)}{t} \quad \text{for } u : \mathbb{R}^d \to \mathbb{R}$$

of a Markov process $(Z_t)_{t \geq 0}$ with stationary distribution P satisfies

$$\mathbb{E}[(\mathcal{L}u)(Z)] = 0$$

under mild conditions on $\mathcal{L}$ and u

Check the conditions.

. In the specific case of an overdamped Langevin diffusion

$$\mathrm{d}Z_t = \frac{1}{2} \nabla \log p(Z_t) \, \mathrm{d}t + \mathrm{d}W_t,$$

where W_t is the standard Brownian motion, the infinitesimal generator becomes

$$(\mathcal{L}_P u)(x) = \frac{1}{2} \langle \nabla u(x), \nabla \log p(x) \rangle + \frac{1}{2} \langle \nabla, \nabla u(x) \rangle.$$

Denoting $g = \frac{1}{2} \nabla u$, they obtain the Stein operator

$$\mathcal{A}_P g = \langle g, \nabla \log p \rangle + \langle \nabla, g \rangle = \langle p^{-1} \nabla, pg \rangle, \tag{1.5}$$

and rewrite (1.4) as

$$\mathcal{D}_{P,\mathcal{G}}(P') = \sup_{g \in \mathcal{G}} \left| \int_{\mathcal{X}} \mathcal{A}_P g \, \mathrm{d}P' \right| \tag{1.6}$$

for a suitably chosen set $\mathcal{G}$. Note that p enters (1.5) via $\nabla \log p$, so the knowledge of its normalising constant is not required to evaluate the operator.

To remove the optimisation step in the calculation of (1.6), Gorham and Mackey (2017) employ a reproducing kernel Hilbert space (RKHS) $\mathcal{H}(k)$ with kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfying the reproducing property:

$$f(x) = \langle f, k(x, \cdot)\rangle \quad \text{for } f \in \mathcal{H}(k).$$

Taking the set

$$\mathcal{G} := \left\{ \mathrm{g} : \mathbb{R}^d \to \mathbb{R}^d \,\middle|\, \sum_{i=1}^{d} \|g_i\|^2_{\mathcal{H}(k)} \leq 1 \right\} \tag{1.7}$$

which defines a unit-ball in a Cartesian product of d copies $\mathcal{H}(k)$, Proposition 2 in Gorham and Mackey (2017) establishes that

$$\mathcal{D}^2_P(P') = \mathbb{E}_{P' \times P'}[k_P(X, \tilde{X})] = \iint_{\mathcal{X}} k_P(x, y)\, \mathrm{d}p'(x)\, \mathrm{d}p'(y), \tag{1.8}$$

where $X, \tilde{X} \sim P'$, p' is the density of P', and $k_P(x, y)$ is given by

$$\begin{aligned} k_P(x, y) :=& (\nabla_x \cdot \nabla_y) k(x, y) \\ &+ \langle \nabla_x k(x, y), \nabla_y \log p(y)\rangle + \langle \nabla_y k(x, y), \nabla_x \log p(x)\rangle \\ &+ k(x, y)\langle \nabla_x \log p(x), \nabla_y \log p(y)\rangle. \end{aligned} \tag{1.9}$$

Here ∇_x and ∇_y are gradients w.r.t. x and y, respectively, and the operator $\nabla_x \cdot \nabla_y$ is defined as:

$$(\nabla_x \cdot \nabla_y) k(x, y) := \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i\, \partial y_i} k(x, y).$$

Note that we have dropped $\mathcal{G}$ in the subscript of (1.8) as the choice of $\mathcal{G}$ does not depend on P or P'. If P' is a discrete distribution, as in (1.1), the squared discrepancy (1.8) becomes

$$\mathcal{D}^2_P\left(\frac{1}{n}\sum_{i=1}^{n} \delta(x_i)\right) = \frac{1}{n^2}\sum_{i,j=1}^{n} k_P(x_i, x_j). \tag{1.10}$$

When $k(x, y)$ is chosen to be the inverse multiquadratic kernel (IMQ)

$$k(x, y) = \left(c^2 + \|\Gamma^{-1/2}(x - y)\|\right)^{\beta} \tag{1.11}$$

with $\beta \in (-1, 0)$, Gorham and Mackey (2017) demonstrate for $\Gamma = I$ that $\mathcal{D}_P(P')$ provides convergence control:

- if $\mathcal{D}_P(P'_m) \to 0$, then P'_m converges in distribution to P (Theorem 8),
- if P'_m converges to P in Wasserstein distance, then $\mathcal{D}_P(P'_m) \to 0$ (Proposition 9)

for a sequence of distributions P'_m under suitable conditions. Theorem 4 by Chen et al. (2019) justifies the introduction of Γ in IMQ.

The constant c in (1.11) can be set to 1 without loss of generality, and the positive-definite preconditioner matrix Γ can be chosen to exploit the geometry of the parameter space. Riabiz et al. (2022) suggest several choices for Γ, in particular the identity matrix scaled by the median Euclidean distance between points in the sample, possibly further rescaled by $(\log m)^{-1/2}$ where m is the desired cardinality of the thinned sample, or the sample covariance matrix. Based on empirical evaluation, Gorham and Mackey (2017) and Riabiz et al. (2022) settle on the value $\beta = -\frac{1}{2}$.

The expression for $k_P(x, y)$ when $k(x, y)$ is taken to be the inverse multiquadratic kernel is derived in section A.1 and is coded directly in the Python library `stein-thinning`[2].

Other choices of kernels are possible and offer convergence control, as demonstrated by Chen et al. (2018), however IMQ performed on par or better than the alternatives considered by the authors.

1.3.2 Stein thinning

Equipped with the kernel Stein discrepancy (KSD) as defined above, Riabiz et al. (2022) develop a greedy optimisation algorithm to select a subset of points from a sample that minimises the total kernel Stein discrepancy. Rather than attempting to evaluate KSD for all $\binom{n}{m}$ combinations of points, they construct a subsample iteratively, each time picking a point that minimises the KSD with previously selected points. We reproduce their procedure verbatim in Algorithm 1 for the reader's convenience.

The algorithm was implemented by the authors and made available in the open-source library `stein-thinning`. The computational complexity of the provided implementation is $O(nm)$.

The strength of the method lies in its ability to correct for bias in the input sample, as established by Theorem 3 in Riabiz et al. (2022), meaning

[2] Available from `https://github.com/wilson-ye-chen/stein_thinning`.

```
Data: Sample $(x_i)_{i=1}^n$ from MCMC, Stein kernel $k_P$, desired
      cardinality $m \in \mathbb{N}$.
Result: Indices $\pi$ of a sequence $(x_{\pi(j)})_{j=1}^m \subset \{x_i\}_{i=1}^n$ where
        $\pi(j) \in \{1, \dots, n\}$.
for $j = 1, \dots, m$ do
    $\pi(j) \in \arg\min_{i=1,\dots,n} \frac{k_P(x_i, x_i)}{2} + \sum_{j'=1}^{j-1} k_P(x_{\pi(j')}, x_i)$
end
```

Algorithm 1: Stein thinning.

that the algorithm can be applied to samples from MCMC chains that have not converged to the target distribution, provided that its domain is sufficiently explored by the chains. The limitation of the method comes from its reliance on the gradients of the log-target, which may be expensive to compute.

1.3.3 Gradient-free kernel Stein discrepancy

Cite cases where gradient cannot be computed easily.

To address the limitation of the kernel Stein discrepancy, Fisher and Oates (2024) propose the gradient-free Stein operator $\mathcal{S}_{P,Q}$ defined for any differentiable function g as

$$\mathcal{S}_{P,Q}g \coloneqq \frac{q}{p}(\langle g, \nabla \log q \rangle + \langle \nabla, g \rangle) = \frac{q}{p}\langle q^{-1}\nabla, qg \rangle. \tag{1.12}$$

This definition generalises expression (1.5) by introducing a proxy distribution Q with density q chosen such that its gradient is easily computable. Note that, again, q is only required to be known up to a normalising constant. When $q = p$, we recover the original Langevin Stein operator (1.5).

Fisher and Oates (2024) proceed to show that, under certain regularity conditions,

$$\int_{\mathcal{X}} \mathcal{S}_{P,Q} \,\mathrm{d}P = 0,$$

and thus define the gradient-free kernel Stein discrepancy

$$\mathcal{D}_{P,Q}(P') = \sup_{g \in \mathcal{G}} \left| \int_{\mathcal{X}} \mathcal{S}_{P,Q} g \,\mathrm{d}P' \right|$$

in analogy with (1.6). The set $\mathcal{G}$ is the unit-ball given by (1.7). Futhermore, Proposition 7 in Fisher and Oates (2024) establishes that

$$\mathcal{D}^2_{P,Q}(P') = \iint_{\mathcal{X}} \frac{q(x)}{p(x)} \frac{q(y)}{p(y)} k_Q(x,y)\,\mathrm{d}p'(x)\,\mathrm{d}p'(y),$$

where p' is the density of distribution P' and $k_Q(x,y)$ is given by (1.9) with q replacing p. For a discrete P', we obtain

$$\mathcal{D}^2_{P,Q}\left(\frac{1}{n}\sum_{i=1}^{n}\delta(x_i)\right) = \frac{1}{n^2}\sum_{i,j=1}^{n} \frac{q(x_i)}{p(x_i)} \frac{q(x_j)}{p(x_j)} k_Q(x_i,x_j). \tag{1.13}$$

Both (1.10) and (1.13) can be viewed as averaging the elements of matrices, and this fact can be used to provide an efficient implementation of the greedy search described in Section 1.3.2.

Chapter 2

Methodology

2.1 Data

2.2 Evaluating the approximation

In order to assess how well the selected sample approximates the posterior distribution, we use the energy distance. Following Rizzo and Székely (2016), the squared energy distance is defined for two distributions F and G as

$$D^2(F, G) \coloneqq 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|,$$

where $X, X' \sim F$, $Y, Y' \sim G$, and $\|\cdot\|$ denotes the Euclidean norm. For samples $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ from X and Y, respectively, the corresponding statistic is given by

$$\mathcal{E}_{n,m}(X, Y) \coloneqq \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\|x_i - y_j\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|x_i - x_j\| - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\|y_i - y_j\|.$$

Chapter 3

Results

Chapter 4

Conclusions

Appendix A

Derivations

A.1 Stein kernel based on inverse multiquadratic kernel

Given a kernel $k(x, y)$, the corresponding Stein kernel is given by (1.9). For the inverse multiquadratic kernel (1.11) we obtain

$$\begin{aligned}\frac{\partial}{\partial x_r} k(x,y) &= \beta \left(c^2 + \sum_{i=1}^{d} \sum_{j=1}^{d} (x_i - y_i) \Gamma^{-1}_{ij} (x_j - y_j) \right)^{\beta - 1} \\ &\quad \times \sum_{j=1}^{d} (\Gamma^{-1} + \Gamma^{-T})_{rj} (x_j - y_j) \\ &= 2\beta \left(c^2 + \sum_{i=1}^{d} \sum_{j=1}^{d} (x_i - y_i) \Gamma^{-1}_{ij} (x_j - y_j) \right)^{\beta - 1} \sum_{j=1}^{d} \Gamma^{-1}_{rj} (x_j - y_j), \end{aligned} \tag{A.1}$$

where we used that Γ is a symmetric matrix. The gradient is then

$$\nabla_x k(x,y) = 2\beta \left(c^2 + \|\Gamma^{-1/2}(x-y)\|^2 \right)^{\beta-1} \Gamma^{-1}(x-y). \tag{A.2}$$

and similarly

$$\nabla_y k(x,y) = -2\beta \left(c^2 + \|\Gamma^{-1/2}(x-y)\|^2 \right)^{\beta-1} \Gamma^{-1}(x-y). \tag{A.3}$$

Now

$$\begin{aligned}\frac{\partial^2}{\partial x_r\,\partial y_r}k(x,y) = &-4\beta(\beta-1)\left(c^2+\sum_{i=1}^{d}\sum_{j=1}^{d}(x_i-y_i)\Gamma^{-1}_{ij}(x_j-y_j)\right)^{\beta-2} \\ &\times\left(\sum_{j=1}^{d}\Gamma^{-1}_{rj}(x_j-y_j)\right)^2 \\ &-2\beta\left(c^2+\sum_{i=1}^{d}\sum_{j=1}^{d}(x_i-y_i)\Gamma^{-1}_{ij}(x_j-y_j)\right)^{\beta-1}\Gamma^{-1}_{rr}\end{aligned} \tag{A.4}$$

which gives us

$$\begin{aligned}(\nabla_x\cdot\nabla_y)k(x,y) = &-4\beta(\beta-1)\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta-2}\|\Gamma^{-1}(x-y)\|^2 \\ &-2\beta\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta-1}\operatorname{trace}(\Gamma^{-1})\end{aligned} \tag{A.5}$$

Substituting (A.2), (A.3) and (A.5) into (1.9), we obtain

$$\begin{aligned}k_P(x,y) = &-4\beta(\beta-1)\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta-2}\|\Gamma^{-1}(x-y)\|^2 \\ &-2\beta\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta-1}\operatorname{trace}(\Gamma^{-1}) \\ &+2\beta\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta-1}\langle\Gamma^{-1}(x-y),\nabla_y\log p(y)\rangle \\ &-2\beta\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta-1}\langle\Gamma^{-1}(x-y),\nabla_x\log p(x)\rangle \\ &+\left(c^2+\|\Gamma^{-1/2}(x-y)\|^2\right)^{\beta}\langle\nabla_x\log p(x),\nabla_y\log p(y)\rangle \\ = &-4\beta(\beta-1)D^{\beta-2}\|\Gamma^{-1}(x-y)\|^2 \\ &-2\beta D^{\beta-1}(\operatorname{trace}(\Gamma^{-1})+\langle\Gamma^{-1}(x-y),\nabla_x\log p(x)-\nabla_y\log p(y)\rangle) \\ &+D^{\beta}\langle\nabla_x\log p(x),\nabla_y\log p(y)\rangle,\end{aligned} \tag{A.6}$$

where we have denoted $D=c^2+\|\Gamma^{-1/2}(x-y)\|^2$.

Bibliography

S. P. Brooks and A. Gelman. General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, December 1998.

W. Y. Chen, L. Mackey, J. Gorham, F.-X. Briol, and C. J. Oates. Stein Points. In *Proceedings of the 35th International Conference on Machine Learning*, pages 843–852. PLMR, 2018.

W. Y. Chen, A. Barp, F.-X. Briol, J. Gorham, M. Girolami, L. Mackey, and Chris. J. Oates. Stein Point Markov Chain Monte Carlo. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, California, 2019. PLMR.

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.

A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4), November 1992.

A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis Jumping Rules. In *Bayesian Statistics*, volume 5, pages 599–608. Oxford University Press, Oxford, May 1996.

A. Gelman, W. R. Gilks, and G. O. Roberts. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1), February 1997.

J. Gorham and L. Mackey. Measuring Sample Quality with Stein's Method. In *Advances in Neural Information Processing Systems*, volume 28. MIT Press, 2015.

J. Gorham and L. Mackey. Measuring Sample Quality with Kernels. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017. PLMR.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

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

M. L. Rizzo and G. J. Székely. Energy distance. *WIREs Computational Statistics*, 8(1):27–38, January 2016.

G. O. Roberts and J. S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4), November 2001.

D. Vats and C. Knudson. Revisiting the Gelman–Rubin Diagnostic. *Statistical Science*, 36(4), November 2021.

A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner. Rank-Normalization, Folding, and Localization: An Improved R for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, 16(2), June 2021.