Stochastic Simulation

Autumn Semester 2024

Prof. Fabio Nobile *Assistant: Matteo Raviola*

Project - 6

Submission deadline: 16 January 2025

MCMC on manifolds

This project concerns the construction of MCMC methods to sample from probability distributions that are concentrated on manifolds of a Euclidean space defined implicitly as subsets of the zero-level set of a function.

Introduction

Many models arising in science and engineering can be understood as constrained systems. In such problems, the space of accessible configurations is lower-dimensional than the space of variables which describe the system, often forming a manifold embedded in the full configuration space.

One may then be interested in sampling a probability distribution defined on the manifold, or calculating an integral over the manifold. The goal of this mini-project is to construct an MCMC algorithm that is able to sample a target distribution on a d-dimensional manifold $\mathcal{M}$ described by equality and inequality constrains, which is embedded on a larger d_a dimensional Euclidean space. More formally, let

$$\mathcal{M} = \Big\{ x \in \mathbb{R}^{d_a} \text{ such that } q_i(x) = 0,\ i = 1, 2, \ldots, m, \text{ and } h_j(x) > 0,\ j = 1, 2, \ldots, l \Big\} \tag{1}$$

be a d-dimensional manifold embedded on an *ambient* space $\mathbb{R}^{d_a}$ (with $d_a > d$), subject to m equality constrains, described by m continuously differentiable functions $q_i(x),\ i = 1, 2 \ldots m$, and l inequality constrains, described by l functions $h_j(x), j = 1, 2, \ldots, l$. Furthermore, denote by G_x the matrix whose columns are the gradients $\{\nabla q_i(x)\}_{i=1}^m$, which is assumed to have full-rank m at any $x \in \mathcal{M}$ hence the manifold has a dimension $d = d_a - m$. Lastly, let $\mathcal{T}_x$ be the tangent space to $\mathcal{M}$ at $x \in \mathcal{M}$. Let us denote the target measure on the manifold by

$$\rho(\mathrm{d}x) = \frac{1}{Z} f(x) \sigma(\mathrm{d}x),$$

where, $\sigma(\mathrm{d}x)$ is the d-dimensional surface area measure, and f is a given (unnormalized) probability density function defined on the manifold. Our goal is then to sample from $\rho(\mathrm{d}x)$ and compute integrals of the form

$$I = \int_{\mathcal{M}} g(x) \sigma(\mathrm{d}x) = Z \int_{\mathcal{M}} \frac{g(x)}{f(x)} \rho(\mathrm{d}x)$$

for some σ-integrable function $g : \mathcal{M} \mapsto \mathbb{R}$. We assume here that $f(x) \neq 0$ whenever $g(x) \neq 0$. To that end, we need to generate samples $x_n \in \mathcal{M}, n = 0, 1, \ldots, N$, distributed as ρ. This can be done by the MCMC algorithm proposed in [1], which we describe in what follows:

Given a current state x_n of the chain, the MCMC algorithm first proposes a tangential move to a state $x_n + v$, with $v \in \mathcal{T}_{x_n}$, which is then followed by a projection back to $y \in \mathcal{M}$, that can be written as $y = x_n + v + w$, with $w \perp \mathcal{T}_{x_n}$.

The algorithm

The MCMC algorithm iteratively repeats the following procedure:

1. **Proposal:** Given some state x_n, the proposal process begins with a tangential move $x_n \to x_n + v$, with $v \in \mathcal{T}_{x_n}$. We generate $v \sim p(\cdot|x_n)$ using an orthonormal basis for $\mathcal{T}_{x_n}$, which is the orthogonal complement of the columns of G_{x_n}. This orthogonal complement basis can be found, for instance, using the last d columns of the $d_a \times d_a$ matrix Q in the full QR factorization of G_{x_n}.

   1.1 **Projection:** Given x_n and v, the projection step looks for some $w \perp \mathcal{T}_{x_n}$, such that $y = x_n + v + w$ satisfies all the equality constraints. It does so by finding an m-dimensional column vector a, and setting $w = \sum_{j=1}^{m} a_j \nabla q_j(x_n) = G_{x_n} a$ such that a solves

   $$q_i(x_n + v + G_{x_n} a) = 0, \quad i = 1, 2, \ldots, m.$$

   This can be done using any non-linear equation solver. If such a solution w can be found, we set as a proposal $y = x_n + v + w$ and advance to 1.2. Othewise, we set $x_{n+1} = x_n$ as the new state of the chain. This procedure is depicted in Figure 1.

   1.2 **Check inequality constrain:** Check if any constrain is violated, that is, check if $h_i(y) \leq 0$ for some i. If so, reject y and set $x_{n+1} = x_n$. Otherwise, advance to 1.3.

   1.3 **Check for lack of reversibility:** In order to satisfy the detailed-balance condition (i.e., reversibility of the chain), we need to verify that we can propose x_n starting from y. To that end, we need to find $v' \in \mathcal{T}_y$ and $w' \in \mathcal{T}_y^{\perp}$ such that $x_n = y + v' + w'$. Such w', v' always exist uniquely and are given by the projection of $x_n - y$ onto $\mathcal{T}_y$ and $\mathcal{T}_y^{\perp}$, respectively; they can be computed using the QR decomposition of G_y. However, one needs to verify that the non-linear solver would find x_n starting from $y + v'$. If it doesn't (in a given number of steps `nmax`), y is rejected and we set $x_{n+1} = x_n$. Otherwise, we continue to step 2.

2. **Acceptance-rejection step** We set $x_{n+1} = y$ with probability $\alpha(x_n, y)$, with

   $$\alpha(x_n, y) = \min\left\{1, \frac{f(y)p(v'|y)}{f(x_n)p(v|x_n)}\right\}, \tag{2}$$

   otherwise, we set $x_{n+1} = x_n$.

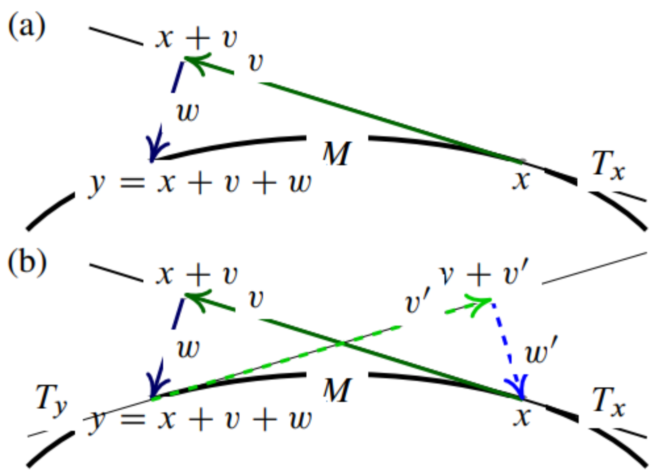

Figure 1: Illustration of the proposal mechanism. In (a), the vector $v \in \mathcal{T}_x$ is projected orthogonally to $\mathcal{T}_x$ to a point $y = x + v + w \in \mathcal{M}$. The reverse step is presented by (b).

Goals of the project

(a) Show that the described algorithm generates a reversible Markov chain. **Hint:** See [1, Section 2.2].

(b) Consider a torus $\mathbb{T}^2$ embedded in $\mathbb{R}^3$, implicitly defined by

$$\mathbb{T}^2 := \left\{ (x, y, z) \in \mathbb{R}^3 \text{ such that } \left(R - \sqrt{x^2 + y^2}\right)^2 + z^2 - r^2 = 0 \right\}, \tag{3}$$

where $R, r > 0$ and $r < R$, and the uniform measure $\rho(\mathrm{d}x) = \frac{1}{Z}\sigma(\mathrm{d}x)$ (i.e., $f = 1$). Implement the algorithm described in the previous section with $R = 1$ and $r = 0.5$, and a Gaussian proposal $p(\cdot|x_n) = \mathcal{N}(0, \Sigma)$, (with suitable Σ) to obtain $N = 10^6$ samples $x_n \in \mathbb{T}^2, n = 1, 2, \dots, N$. Describe, in particular, the construction of the covariance matrix Σ. Verify the accuracy of your implementation by plotting the points $\{x_n\}_{n=1}^N$ obtained by the sampler.

(c) Based on the generated chain $\{x_n\}_{n=1}^N$, estimate the *moment of inertia, in the x direction* given by

$$I = \int_{\mathbb{T}^2} x^2 \sigma(\mathrm{d}x) = Z \int_{\mathbb{T}^2} x^2 \rho(\mathrm{d}x), \tag{4}$$

with $Z = 4\pi^2 rR$. Estimate the sample size N needed to achieve a root mean squared error smaller than a prescribed tolerance `tol`. Describe the method you use for the error estimation. Monitor the convergence of your estimator.

(d) An explicit parametrization of $\mathbb{T}^2$ is given by

$$\mathbb{T}^2 := \{[(R + r\cos(\phi))\cos(\theta), (R + r\cos(\phi))\sin(\theta), r\sin(\phi)] : \ \theta, \phi \in [0, 2\pi]\}. \tag{5}$$

Exploit this to construct a Monte Carlo estimator of I and compare its efficiency with that of the estimator in the previous point.

(e) We now consider the more interesting example of sampling from the *special orthogonal group* $SO(d)$ with uniform distribution. We view $SO(d)$ as the set of $d \times d$ matrices, $x \in R^{d \times d}$ that satisfy the following $\frac{1}{2}d(d+1)$ row orthonormality constraints for $k = 1, \ldots, d$ and $l > k$

$$q_{kk}(x) = \sum_{m=1}^{d} x_{km}^2 = 1, \quad q_{kl}(x) = \sum_{m=1}^{d} x_{km} x_{lm} = 0. \tag{6}$$

Choosing $f(x) = 1$, implement the manifold MCMC algorithm to obtain 10^6 samples from $SO(d)$ with $d = 11$. Notice that any x satisfying (6) has $\det(x) = \pm 1$. The set with $\det(x) = 1$ is connected. It is possible that the sampler would propose an x with $\det(x) = -1$. In this case, this proposal should be rejected. It is known that the distribution of $T = \mathrm{Tr}(x)$ converges to a standard normal as $d \to \infty$. As suggested in [1], one can use this fact to assess the correctness of the manifold MCMC algorithm.

(f) Could you imagine an altenrative way to generate an i.i.d. sample from the uniform distribution on $SO(d)$? How would it compare in terms of efficiency with the algorithm in point (e)?

References

[1] Emilio Zappa, Miranda Holmes-Cerfon, and Jonathan Goodman. Monte Carlo on manifolds: sampling densities and integrating functions. *Communications on Pure and Applied Mathematics*, 71(12):2609–2647, 2018.