Proximal MCMC for linearly constrained multivariate normal distributions

FabianKP

Notation

- $\mathbb{1}_C$ denotes the indicator function,

$$\mathbb{1}_C(x) := \begin{cases} 1, & \text{if } x \in C, \\ 0, & \text{otherwise.} \end{cases}$$

- χ_C denotes the characteristic function,

$$\chi_C(x) := \begin{cases} 0, & \text{if } x \in C, \\ \infty, & \text{otherwise.} \end{cases}$$

  (Not to be confused with $\chi^2_{q,n}$, which denotes the q-quantile of the chi-squared distribution with n degrees of freedom.)

1 Introduction

We consider linearly constrained multivariate normal distributions on $\mathbb{R}^d$. Such distributions have the general form

$$p(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m})\right) \mathbb{1}_{\mathcal{C}}(\boldsymbol{x}),$$
$$\mathcal{C} = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{C}\boldsymbol{x} \geq \boldsymbol{d}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \right\}.$$

Sampling from this distribution is important in a range of applications, in particular in Bayesian inverse problems with constraints. In such applications, the dimension d will often be very large, and the covariance matrix $\boldsymbol{\Sigma}$ will sometimes be ill-conditioned. In that case, sampling from p using non-specialized methods does not work.

One method that was proposed for sampling from ill-conditioned log-concave distributions such as these is the so-called **proximal Markov chain Monte Carlo method** (proximal MCMC) [2, 1].

2 Description of proximal MCMC

2.1 Basic terminology and definitions

In the following, let $p : \mathbb{R}^d \to [0, \infty)$ be a probability density function of the form

$$p(\boldsymbol{x}) = \frac{1}{Z} e^{-f(\boldsymbol{x})},$$

where $Z > 0$ is a constant and $f : \mathbb{R}^d \to (-\infty, \infty]$ is a convex function.

Given $\lambda > 0$, we define the *proximal operator* $\text{prox}_f^\lambda : \mathbb{R}^d \to \mathbb{R}$ associated to f and λ as

$$\text{prox}_f^\lambda(\boldsymbol{x}) = \underset{\boldsymbol{z}}{\text{argmin}}\{f(z) + \frac{1}{2\lambda}\|z - x\|_2^2\}.$$

Furthermore, we define the *Moreau-Yosida envelope* f_λ associated to f and λ as

$$f_\lambda(\boldsymbol{x}) = \inf_{\boldsymbol{z}}\{f(\boldsymbol{z}) + \frac{1}{2\lambda}\|z - x\|_2^2\}.$$

2.2 High-level description

2.2.1 Px-MALA

The proximal Metropolis-adjusted Langevin algorithm (Px-MALA) is a special case of the general Metropolis-Hastings scheme. This means it creates a Markov chain $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$ by repeating the following two steps for $n = 0, \dots, N$ (the first sample x_0 is given):

1. Given the current iterate $\boldsymbol{x}_n \in \mathbb{R}^d$, create a proposal $\boldsymbol{y}_{n+1} \sim q(\cdot|\boldsymbol{x}_n)$ from the proposal kernel $q(\cdot|\cdot)$.
2. Compute the Hastings-ratio

$$r_{n+1} = \min\left(1, \frac{p(\boldsymbol{y}_n)q(\boldsymbol{x}_n|\boldsymbol{y}_n)}{p(\boldsymbol{x}_n)q(\boldsymbol{y}_n|\boldsymbol{x}_n)}\right).$$

Then, with probabilty r, set $\boldsymbol{x}_{n+1} = \boldsymbol{y}_n$. Otherwise, set $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n$.

Proximal MCMC uses the following proposal kernel:

$$q(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(\operatorname{prox}_f^\delta(\boldsymbol{x}), 2\delta\mathbb{I}_d\right).$$

That is, given $\boldsymbol{x} \in \mathbb{R}^d$, we can generate a new proposal $\boldsymbol{y}$ by sampling $\boldsymbol{z} \sim \mathcal{N}(0, \mathbb{I})$ and then setting

$$\boldsymbol{y} = \operatorname{prox}_f^\delta(\boldsymbol{x}) + \sqrt{2\delta}\boldsymbol{z}.$$

2.2.2 MYULA

Alternatively, let

$$f(\boldsymbol{x}) = h(\boldsymbol{x}) + g(\boldsymbol{x}),$$

where we assume that h is twice continuously differentiable. Let $\lambda > 0$ and $(\delta_n)_{n=1}^\infty$ be a nonincreasing sequence of positive stepsizes. We define the so-called *Moreau-Yosida Unadjusted Langevin Algorithm* (MYULA) by the iteration

$$\begin{aligned} z_{n+1} &\sim \mathcal{N}(0, \mathbb{I}), \\ x_{n+1} &= x_n - \delta_{n+1}\left(\nabla h(x_n) + \lambda^{-1}(x_n - \operatorname{prox}_g^\lambda(x_n))\right) + \sqrt{2\delta_{n+1}}z_{n+1}, \end{aligned}$$

An important caveat for the samples obtained from MYULA is that they represent a smoothed distribution

$$p_\lambda(\boldsymbol{x}) = \frac{1}{Z}e^{-h(\boldsymbol{x})+g_\lambda(\boldsymbol{x})}.$$

Hence, when estimating a quantity of interest

$$\bar{\phi} = \int \phi(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}x,$$

instead of estimating it by an equally weighted sum over the samples,

$$\bar{\phi}^{\text{bad}} = \frac{1}{N}\sum_{n=1}^N \phi(x_n),$$

one should use importance sampling,

$$\bar{\phi}^{\text{good}} = \sum_{n=1}^N w_n\phi(x_k),$$

where

$$\begin{aligned} w_n &:= \frac{\delta_n e^{\bar{g}_\lambda(\boldsymbol{x}_n)}}{\sum_{m=1}^N \delta_m e^{\bar{g}_\lambda(\boldsymbol{x})}}, \\ \bar{g}_\lambda(\boldsymbol{x}) &= g_\lambda(\boldsymbol{x}) - g(\boldsymbol{x}). \end{aligned}$$

2.3 Specialization to constrained Gaussians

2.3.1 Px-MALA

In our particular case, we have

$$f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m}) + \chi_{\mathcal{C}}(\boldsymbol{x}).$$

In order to evaluate the proximal operator $\operatorname{prox}_f^\lambda$, one has to solve a constrained least-squares problem

$$\operatorname{prox}_f^\lambda(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{z}} \left\{ \left\| \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{z}-\boldsymbol{m}) \right\|_2^2 + \frac{1}{\lambda}\left\|\boldsymbol{z}-\boldsymbol{x}\right\|_2^2 : \boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}, \boldsymbol{C}\boldsymbol{z} \geq \boldsymbol{d}, \boldsymbol{l} \leq \boldsymbol{z} \leq \boldsymbol{u} \right\}.$$

This problem can be solved fast using any method for quadratic optimization.

2.3.2 MYULA

For MYULA, we use the following splitting:

$$\begin{aligned}
&f(\boldsymbol{x}) = h(\boldsymbol{x}) + g(\boldsymbol{x}),\\
&h(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m}),\\
&g(\boldsymbol{x}) = \chi_{\mathcal{C}}(\boldsymbol{x}),\\
&\mathcal{C} = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{C}\boldsymbol{x} \geq \boldsymbol{d}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \right\}.
\end{aligned}$$

Performing a MYULA-step thus requires evaluation of

$$\nabla h(\boldsymbol{x}) = \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{m})$$

and

$$\operatorname{prox}_g^\lambda(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{z}} \left\{ \left\|\boldsymbol{z}-\boldsymbol{x}\right\|_2^2 : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{C}\boldsymbol{x} \geq \boldsymbol{d}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \right\}.$$

(since minimizing $\frac{1}{\lambda}\left\|\boldsymbol{z}-\boldsymbol{x}\right\|$ is equivalent to minimizing $\left\|\boldsymbol{z}-\boldsymbol{x}\right\|$). The latter can again be implemented using any general-purpose quadratic optimization solver.

2.4 Choosing the hyper-parameters

Finally, let us provide some guidelines on how to choose the hyperparameters for Px-MALA and MYULA:

Px-MALA

The stepsize parameter δ should usually be small. A common rule-of-thumb is that one should choose δ such that the acceptance frequency is around 0.5. Our implementation provides an option to automatically adapt δ during burnin to achieve a prescribed acceptance frequency.

MYULA

The sequence of stepsizes $(\gamma_n)_{n=1}^\infty$ should be nonincreasing and satisfy $\gamma_n < \frac{2}{\operatorname{Lip}(h)+1/\lambda}$ for all $n \in \mathbb{N}$, where $\operatorname{Lip}(h)$ is the Lipschitz constant of ∇h. If $\operatorname{Lip}(h)$ is not known, γ has to be tuned by hand. Since in any case $\operatorname{Lip}(h) \geq 0$, we now that γ and λ in any case have to satisfy the relation $\gamma \leq 2\lambda$. Hence, in absence of further information, $\gamma = \lambda$ can be used as a default choice.

2.5 Pseudocode

Px-MALA

Note that the proposal kernel for Px-MALA satisfies

$$\log q(\boldsymbol{y}|\boldsymbol{x}) = -\frac{1}{4\delta}\left\|\boldsymbol{y} - \operatorname{prox}_f^\delta(\boldsymbol{x})\right\|_2^2. \tag{2.1}$$

Note that for the evaluation of the Hastings ratio, it is easier to work with log-probabilities. That is, instead of computing

$$r = \min\left(1, \frac{p(\boldsymbol{y})q(\boldsymbol{x}|\boldsymbol{y})}{p(\boldsymbol{x})q(\boldsymbol{y}|\boldsymbol{x})}\right),$$

it is better to compute (using (2.1))

$$s = h(\boldsymbol{x}) - h(\boldsymbol{y}) + \frac{1}{4\delta}\left\|\boldsymbol{x} - \mathrm{prox}_f^\delta(\boldsymbol{y})\right\|_2^2 - \frac{1}{4\delta}\left\|\boldsymbol{y} - \mathrm{prox}_f^\delta(\boldsymbol{x})\right\|_2^2$$

and then to set $r = \min(1, e^s)$.

Algorithm 1 `Px-MALA`

Given a feasible point $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\delta > 0$, and sample size $N \in \mathbb{N}$;

for $n = 0, 1, 2, \ldots, N$ **do**

$\boldsymbol{x} \leftarrow \boldsymbol{x}_n$;

Solve

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & \{\left\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\xi} - \boldsymbol{m})\right\|_2^2 + \frac{1}{\delta}\left\|\boldsymbol{\xi} - \boldsymbol{x}\right\|_2^2\} \\ \text{s. t.} \quad & \boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b}, \quad \boldsymbol{C}\boldsymbol{\xi} \geq \boldsymbol{d}, \quad \boldsymbol{l} \leq \boldsymbol{\xi} \leq \boldsymbol{u}, \end{aligned}$$

and store the minimizer as $\boldsymbol{\xi}$;

Sample $\boldsymbol{z} \sim \mathcal{N}(0, \mathbb{I}_d)$;

Set $\boldsymbol{y} \leftarrow \boldsymbol{\xi} + \sqrt{2\delta}\boldsymbol{z}$;

Solve

$$\begin{aligned} \min_{\boldsymbol{\zeta}} \quad & \{\left\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\zeta} - \boldsymbol{m})\right\|_2^2 + \frac{1}{\delta}\left\|\boldsymbol{\zeta} - \boldsymbol{y}_{n+1}\right\|_2^2\} \\ \text{s. t.} \quad & \boldsymbol{A}\boldsymbol{\zeta} = \boldsymbol{b}, \quad \boldsymbol{C}\boldsymbol{\zeta} \geq \boldsymbol{d}, \quad \boldsymbol{l} \leq \boldsymbol{\zeta} \leq \boldsymbol{u}, \end{aligned}$$

and store the minimizer as $\boldsymbol{\zeta}$;

$h \leftarrow \frac{1}{2}\left\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x} - \boldsymbol{m})\right\|_2^2$;

$\tilde{h} = \frac{1}{2}\left\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{y} - \boldsymbol{m})\right\|_2^2$;

$q \leftarrow \frac{1}{4\delta}\left\|\boldsymbol{x} - \boldsymbol{\xi}\right\|_2^2$;

$\tilde{q} \leftarrow \frac{1}{4\delta}\left\|\boldsymbol{y} - \boldsymbol{\zeta}\right\|_2^2$;

$s \leftarrow h - \tilde{h} + q - \tilde{q}$;

$r \leftarrow \min(1, e^s)$;

Sample $\eta \sim \mathrm{U}([0, 1])$;

if $r \geq \eta$ **then**

$\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{y}$;

else

$\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}$;

end if

end for

return $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$;

Algorithm 2 `MYULA`

Given a feasible point $\boldsymbol{x}_0 \in \mathbb{R}^d$, sample size $N \in \mathbb{N}$, $\lambda > 0$, and a nonincreasing sequence $(\delta_n)_{n=1}^N > 0$;
for $n = 0, 1, 2, \ldots, N$ **do**
  $\boldsymbol{x} \leftarrow \boldsymbol{x}_n$;
  Solve

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \\ \text{s. t.} \quad & \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \quad \boldsymbol{C}\boldsymbol{y} \geq \boldsymbol{d}, \quad \boldsymbol{\ell} \leq \boldsymbol{y} \leq \boldsymbol{u}. \end{aligned}$$

  $\boldsymbol{u} \leftarrow \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{m})$;
  $\boldsymbol{z} \leftarrow \mathcal{N}(\boldsymbol{0}, \mathbb{I}_d)$;
  $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x} - \delta_{n+1}(\boldsymbol{y} - \lambda^{-1}(\boldsymbol{x} - \boldsymbol{u}) + \sqrt{2\delta_{n+1}}\boldsymbol{z}$;
end for
return $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$;

3 Indicators of convergence

Assessing the quality of an MCMC run is notoriously difficult. We present a couple of numerical quantities that allow the user to judge whether the obtained samples represent the probability distribution of interest.

In the following, we assume that we are given M independently run MCMC chains $(\mathbf{x}_n^{(1)})_{n=1}^N, \ldots, (\mathbf{x}_n^{(M)})_{n=1}^N$, each of length N. The subsequent computations also make sense for a single chain ($M = 1$), but it is in general recommended to perform multiple MCMC runs if at all possible.

Let us fix some notation that we will make heavy use of in the rest of this section: For $m = 1, \ldots, M$, let

$$\bar{\mathbf{x}}_N^m = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n^{(m)}$$

be the sample average and

$$\boldsymbol{\Sigma}_m = \frac{1}{N-1} \sum_{n=1}^N (x_n^{(m)} - \bar{x}^{(m)})(x_n^{(m)} - \bar{x}^{(m)})^\top$$

be the sample covariance of the m-th chain. Furthermore, let

$$\bar{\mathbf{x}}_N = \frac{1}{M} \sum_{m=1}^M \bar{\mathbf{x}}_N^m$$

be the sample average and

$$\boldsymbol{\Sigma} = \frac{1}{M} \sum_{m=1}^M \boldsymbol{\Sigma}_m$$

be the sample covariance of all chains.

3.1 Effective sample size

A very intuitive tool for a-posteriori assessments of the quality of a given MCMC chain is the so-called *effective sample size*. Its multivariate version is described in [3]. Intuitively, the effective sample size gives the number of independent samples with the same estimation power as the MCMC samples. That is, estimating a quantity of interest using an MCMC sample with effective sample size N_{eff} is comparable to an estimate that uses N_{eff} independent samples from the posterior distribution.

Definition of the effective sample size estimator

The effective sample size can be estimated by

$$\widehat{\mathrm{ESS}} = MN \left(\frac{\det(\mathbf{\Sigma})}{\det(\mathbf{T}_L)} \right)^{1/d},$$

where

- $\mathbf{\Sigma}$ is the sample covariance matrix of all chains as described above;
- $\hat{\mathbf{T}}_L$ is the so-called *multivariate replicated lugsail batch means estimator* for the Monte Carlo standard error. Given a batch size $b < N$, let $a = \lfloor b \cdot N \rfloor$ denote the number of batches. For $i = 1, \ldots, a$, let

  $$\hat{\boldsymbol{x}}_i^{(m)} = \frac{1}{b} \sum_{j=1}^{b} \boldsymbol{x}_{(i-1)b+j}^{(m)}$$

  denote the i-th batch mean for the m-th chain. Let

  $$\hat{\mathbf{T}}_b = \frac{b}{aM-1} \sum_{m=1}^{M} \sum_{i=1}^{a} (\bar{\mathbf{x}}_i^{(m)} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i^{(m)} - \bar{\mathbf{x}})^\top.$$

  Then, $\hat{\mathbf{T}}_L$ is defined as

  $$\hat{T}_L = 2\hat{T}_b - \hat{T}_{b/3}.$$

Usage

The authors of [3] propose to stop the sampling once

$$\widehat{\mathrm{ESS}} \geq W(d, \alpha, \epsilon),$$

with the number $W(d, \alpha, \epsilon)$ defined as

$$W(d, \alpha, \epsilon) = \frac{2^{2/d} \pi \chi^2_{1-\alpha,d}}{(d\Gamma(d/2))^{2/d} \epsilon^2}, \tag{3.1}$$

where

- d is the dimension of the parameter space.
- α is the desired confidence level (e.g. $\alpha = 0.05$ for 95%-confidence.
- ϵ is the desired relative precision. That is, a value of $\epsilon = 0.1$ means that approximately 10% of the variability in the samples come from the Monte Carlo error.
- $\chi^2_{1-\alpha,d}$ is the $(1-\alpha)$-percentile of the chi-squared distribution with d degrees of freedom.
- Γ is the gamma function, $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} \, \mathrm{d}x$.

3.2 R hat

The $\hat{R}$ statistic (also known as Gelman-Rubin statistic or potential scale reduction factor (PSRF)) is the most widely used convergence diagnostic for MCMC. The current state-of-the-art for computing $\hat{R}$ in the univariate case is described in [5].

We implement the stabilized, multivariate version of $\hat{R}$ described in [4]. The precise definition is provided next.

Definition of R hat

In [4], the multivariate stabilized $\hat{R}$ statistic is defined as

$$\hat{R}_L^d = \sqrt{\frac{N-1}{N} + \frac{\det(\mathbf{\Sigma}^{-1} \hat{\mathbf{T}}_L)^{1/d}}{N}},$$

where $\mathbf{\Sigma}$ and $\hat{\mathbf{T}}_L$ are as above.

Usage

A simple rule-of-thumb [5] is that samples should only be used if $\hat{R}^d < 1.01$.

The authors of [4] provide an alternative criterion that is motivated by the relation of $\hat{R}^d$ to the effective sample size. They recommend that samples should only be used if

$$\hat{R}_L^d \leq \sqrt{1 + \frac{M}{W(\alpha, \epsilon, d)}},$$

where $W(\alpha, \epsilon, d)$ is defined in (3.1).

References

[1] A. Durmus, É. Moulines, and M. Pereyra. "Efficient Bayesian Computation by Proximal Markov Chain Monte Carlo: When Langevin Meets Moreau". In: *SIAM Journal on Imaging Sciences* 11.1 (Jan. 2018), pp. 473–506. ISSN: 1936-4954. DOI: 10.1137/16M1108340 (cited on page 1).

[2] M. Pereyra. "Proximal Markov Chain Monte Carlo Algorithms". In: *Statistics and Computing* 26.4 (July 1, 2016), pp. 745–760. ISSN: 1573-1375. DOI: 10.1007/s11222-015-9567-4 (cited on page 1).

[3] D. Vats, J. M. Flegal, and G. L. Jones. "Multivariate Output Analysis for Markov Chain Monte Carlo". In: *Biometrika* 106.2 (June 1, 2019), pp. 321–337. ISSN: 0006-3444, 1464-3510. DOI: 10.1093/biomet/asz002 (cited on pages 5, 6).

[4] D. Vats and C. Knudson. "Revisiting the Gelman–Rubin Diagnostic". In: *Statistical Science* 36.4 (Nov. 2021), pp. 518–529. ISSN: 0883-4237, 2168-8745. DOI: 10.1214/20-STS812 (cited on pages 6, 7).

[5] 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)". In: *Bayesian Analysis* 16.2 (June 1, 2021). ISSN: 1936-0975. DOI: 10.1214/20-BA1221 (cited on pages 6, 7).