Markov Chains

Deadline: 05.03.25, 23:59

Submission: Homework should be submitted as one Jupyter Notebook. After the submission, it may be relaunched in Google Colab and it should not crash, as well as outputs should not change. Make sure you fix random seeds in all random experiments.
Policy on open-source: You may use any preimplemented code you may find on the web, e.g. code from seminar, open-source libraries, implementations of papers, etc. However, you may not consult with anyone else about your solutions. Any suspiciously similar code will be considered as cheating.

Exercise 1 (30 points). Implement adaptive MCMC algorithm with NF proposal to sample from mixture of Gaussians:

$$\pi = \frac{1}{2}\mathcal{N}(a, \sigma^2 \cdot I_{d\times d}) + \frac{1}{2}\mathcal{N}(-a, \sigma^2 \cdot I_{d\times d})$$

Specifically, perform the following steps:

1. Fix a set for distribution parameters, e.g. $d \in \{2, 10, 50, 100\}$, $\sigma \in \{0.1, 1, 3\}$, $a = (1, 1, \ldots, 1)^T$. Perform the following steps for each triple of them.
2. Implement non-adaptive I-SIR algorithm with a Cauchy distribution as proposal.
3. Implement [1, Algorithm 1] with MALA and NF proposal of your choice. This algorithm iterates IMH+NF and MALA algorithms.
4. In the similar manner, implement Adaptive version of I-SIR algorithm. It should iterate I-SIR+NF and MALA algorithms.
5. Visualize the samples for $d = 2$ with the contour plot of the target density. Report the metrics (ESTV, ESS, EMD) averaged over 50 independent runs of each sampler, measured w.r.t. "true" sample which is sampled from the given distribution. Compare metrics for different samplers, make conclusion.

Note 1: We have discussed, that there are several ways to learn NF, e.g. using forward or reverse KL or their mixture. [1, Algorithm 1] uses forward KL, which requires samples from π. Thus suggested procedure is iterative. It iterates steps of IMH + NF proposal and MALA.
Note 2: Several references:

- See [2, Section 2] on difference of optimising w.r.t. forward and reverse KL, as well as on description of adaptive algorithms. Also see this link to the official code implementation to see how it is done (e.g. in this file see training procedure using different combinations of KL and this section for samplers implementation).
- See [3, Section 3] for another description of adaptive procedures.
- See this link for an example of preimplemented NF library.

Exercise 2 (20 points). For $x \in \mathbb{R}^{2d}$, define the density of the asymmetric banana-shaped distribution by the formula

$$\pi(x) = \frac{1}{Z}\exp\left(\sum_{i=1}^{d} -(x_{2i-1} - x_{2i}^2)^2/\nu - (x_{2i} - 1)^2\right), \tag{1}$$

with Z being a normalizing constant and $\nu > 0$. Implement sampling from $\pi(x)$ using the following methods:

- MALA (adjust the step size properly!);
- HMC or NUTS;
- adaptive i-SIR (with NF proposal).

Visualize the projections of the samples with the contour plot of the target density. Report the metrics (ESTV, ESS, EMD) averaged over 50 independent runs of each sampler, measured w.r.t. "true" sample which is sampled from the given distribution.

Exercise 3 (50 points). Consider the sampling example on the Bayesian logistic regression. The training set $\mathcal{D}$ consists of pairs (x, y) where $x \in \mathbb{R}^d$ and labels $y \in \{-1, 1\}$. In practice, the first coordinate of x represents the bias term, that is, we fix $x_0 = 1$ for all data points. The likelihood for a pair is given by

$$p(y \mid x, \theta) = \text{logit}(y\langle x, \theta\rangle)\,.$$

Given a prior distribution $p_0(\theta)$, we sample from the posterior distribution $p(\theta \mid \mathcal{D})$ and compute the predictive posterior $p(y \mid x, \mathcal{D}) = \int p(y \mid x, \theta)p(\theta \mid \mathcal{D})d\theta$ for $(x, y) \in \mathcal{D}^{test}$. You should approximate $p(y \mid x, \mathcal{D})$ using the Monte Carlo estimate

$$\widehat{p}_n(y \mid x, \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} p(y \mid x, \mathcal{D}, \theta_i)\,,$$

where θ_i is a sample from posterior $p(\theta \mid \mathcal{D})$ obtained using different MCMC samplers. Use a normal prior distribution $p_0(\theta) = \mathcal{N}(0, \sigma^2 I_d)$ and try different values of σ^2 and the following datasets:

- *EEG* dataset consists of 15k instances of dimension 15, `https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State`
- *Digits* dataset consists of 1.8k instances of dimension 64. Consider arbitrary 2 classes from the original 10 ones for binary classification task, `https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits`

You should randomly take 0.8 of each dataset as a train set and the rest for the test set. You are also allowed to remove outliers from training set with any preprocessing method. You should implement the following methods:

- ULA (adjust the step size properly!);
- MALA (adjust the step size properly!);
- i-SIR (try different proposals);
- HMC or NUTS;
- adaptive i-SIR (with NF proposal).

You need to visualize the boxplots of the posterior predictive distribution averaged over the dataset based on 50 independent runs of each sampler.

References

[1] Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10), March 2022.

[2] Louis Grenioux, Alain Durmus, Éric Moulines, and Marylou Gabrié. On sampling with approximate transport maps, 2024.

[3] Sergey Samsonov, Evgeny Lagutin, Marylou Gabrié, Alain Durmus, Alexey Naumov, and Eric Moulines. Local-global mcmc kernels: the best of both worlds, 2022.