Efficient Sampling Techniques

Sergey Samsonov

HDI Lab,

HSE University

June 20-25, 2022

Recap: Importance Sampling procedure

- Aim: sample from π and estimate $\pi(f) = \int_{\mathbb{R}^D} f(x)\pi(\mathrm{d}x)$;
- π is known up to a normalizing factor Z_Π, $\pi(\mathrm{d}x) = \tilde{\pi}(\mathrm{d}x)/Z_\Pi$;
- Importance Sampling (IS) consists of re-weighting samples from a proposal distribution Λ.
- Define *importance weights* as $\tilde{w}(x) = \tilde{\pi}(x)/\lambda(x)$;
- The *self-normalized importance sampling* (SNIS) estimator of $\pi(f)$ is then given by

$$\widehat{\pi}_N(f) = \sum_{i=1}^{N} \omega_N^i f(X^i)\,,$$

where

$$X^{1:N} \sim \Lambda\,, \omega_N^i = \frac{\tilde{w}(X^i)}{\sum_{j=1}^{N} \tilde{w}(X^j)}\,, i \in \{1, \ldots, N\}\,.$$

From IS to SIR

- Sampling counterpart of the IS procedure is known as Sampling Importance Resampling (SIR; Rubin [1987]);
- Sample $X^1, \ldots, X^N$ - i.i.d. from Λ and compute the importance weights $\omega_N^1, \ldots, \omega_N^N$;
- Sample $Y^1, \ldots, Y^M$ from $X^1, \ldots, X^N$ with replacement, and with probabilities proportional to the weights $\omega_N^1, \ldots, \omega_N^N$. That is, we sample from the empirical distribution

$$\hat{\pi}(\mathrm{d}x) = \sum_{i=1}^{N} \omega_N^i \delta_{X^i}(\mathrm{d}x)\,,$$

  where $\delta_y(\mathrm{d}x)$ denotes the Dirac mass at y.
- As $N \to \infty$, $Y^1, \ldots, Y^M \sim \hat{\Pi}$ will be distributed according to π.
- Main drawback: the described procedure is only asymptotically valid.

Iterated SIR (i-SIR) algorithm

Iterating samples from Λ, we arrive at iterated SIR algorithm (i-SIR, Andrieu et al. [2010], and Andrieu et al. [2018]).

Algorithm 1: Single stage of i-SIR algorithm

Input : Sample Y_j from previous iteration
Output: New sample Y_{j+1}

Set $X_{j+1}^1 = Y_j$ and draw $X_{j+1}^{2:N} \sim \Lambda$.
for $i \in [N]$ **do**
    compute the normalized weights
    $\omega_{i,j+1} = \tilde{w}(X_{j+1}^i) / \sum_{k=1}^N \tilde{w}(X_{j+1}^k)$.
Set $I_{j+1} = \mathrm{Cat}(\omega_{1,j+1}, \ldots, \omega_{N,j+1})$.
Draw $Y_{j+1} = X_{j+1}^{I_{j+1}}$.

Excercises

In the sequel, we denote by $w(x)$ the normalized weight function, that is,

$$\pi(\mathrm{d}x) = w(x)\Lambda(\mathrm{d}x)\,.$$

i-SIR kernel

Write down the Markov kernel of i-SIR algorithm:

$$\mathsf{P}_N(x, \mathsf{A}) = \mathsf{P}(X_{k+1} \in \mathsf{A} | X_k = x)$$

Excercises

In the sequel, we denote by $w(x)$ the normalized weight function, that is,

$$\pi(\mathrm{d}x) = w(x)\Lambda(\mathrm{d}x)\,.$$

i-SIR kernel

Write down the Markov kernel of i-SIR algorithm:

$$\mathrm{P}_N(x, \mathrm{A}) = \mathrm{P}(X_{k+1} \in \mathrm{A} | X_k = x)$$

Solution

$$\mathrm{P}_N(x, \mathrm{A}) = \int \delta_x(\mathrm{d}x_1) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} \mathbb{1}_{\mathrm{A}}(x^i) \prod_{j=2}^{N} \Lambda(\mathrm{d}x^j) \qquad (1)$$

Excercises

Invariant distribution

Check that the distribution π is invariant for the kernel $\mathrm{P}_N(x, \mathrm{A})$.

Hint: symmetrization

$$\mathrm{P}_N(x, \mathrm{A}) = \int \delta_x(\mathrm{d}x_1) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} \mathbb{1}_{\mathrm{A}}(x^i) \prod_{j=2}^{N} \Lambda(\mathrm{d}x^j)$$

$$= \frac{1}{N} \int \sum_{\ell=1}^{N} \delta_x(\mathrm{d}x_\ell) \prod_{j \neq \ell} \Lambda(\mathrm{d}x^j) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{\ell=1}^{N} w(x^\ell)} \mathbb{1}_{\mathrm{A}}(x^i) \,.$$

Invariant distribution: continue

Solution

$$\int \pi(\mathrm{d}x)\mathsf{P}_N(x,\mathsf{A}) = \left\{\pi(\mathrm{d}x)\delta_x(\mathrm{d}x_\ell) = \Lambda(\mathrm{d}x^\ell)w(x^\ell)\right\}$$

$$= N^{-1}\int \pi(\mathrm{d}x)\sum_{\ell=1}^{N}\delta_x(\mathrm{d}x_\ell)\prod_{j\neq\ell}\Lambda(\mathrm{d}x^j)\sum_{i=1}^{N}\frac{w(x^i)}{\sum_{\ell=1}^{N}w(x^\ell)}\mathbb{1}_{\mathsf{A}}(x^i)$$

$$= N^{-1}\int (\sum_{\ell=1}^{N}w(x_\ell))\prod_{j=1}^{N}\Lambda(\mathrm{d}x^j)\sum_{i=1}^{N}\frac{w(x^i)}{\sum_{\ell=1}^{N}w(x^\ell)}\mathbb{1}_{\mathsf{A}}(x^i)$$

Invariant distribution: continue

Solution

$$\int \pi(\mathrm{d}x)\mathrm{P}_N(x,\mathrm{A}) = \left\{\pi(\mathrm{d}x)\delta_x(\mathrm{d}x_\ell) = \Lambda(\mathrm{d}x^\ell)w(x^\ell)\right\}$$

$$= N^{-1}\int \pi(\mathrm{d}x)\sum_{\ell=1}^{N}\delta_x(\mathrm{d}x_\ell)\prod_{j\neq\ell}\Lambda(\mathrm{d}x^j)\sum_{i=1}^{N}\frac{w(x^i)}{\sum_{\ell=1}^{N}w(x^\ell)}\mathbb{1}_\mathrm{A}(x^i)$$

$$= N^{-1}\int (\sum_{\ell=1}^{N}w(x_\ell))\prod_{j=1}^{N}\Lambda(\mathrm{d}x^j)\sum_{i=1}^{N}\frac{w(x^i)}{\sum_{\ell=1}^{N}w(x^\ell)}\mathbb{1}_\mathrm{A}(x^i)$$

Solution

$$\int \pi(\mathrm{d}x)\mathrm{P}_N(x,\mathrm{A}) = N^{-1}\int\prod_{j=1}^{N}\Lambda(\mathrm{d}x^j)\sum_{i=1}^{N}w(x^i)\mathbb{1}_\mathrm{A}(x^i)$$

$$= \left\{\Lambda(\mathrm{d}x^i)w(x^i) = \pi(\mathrm{d}x^i)\right\} = \pi(\mathrm{A})\,.$$

Let us check, how i-SIR works!

- Ergodicity properties of i-SIR still depends upon $L = \sup_x \frac{\pi(x)}{\lambda(x)}$;
- L typically increases exponentially when d is large (the curse of dimension)
- Let $\pi(x) \sim \mathcal{N}(0, \mathrm{I}_d)$, and $\lambda(x) \sim \mathcal{N}(0, 2\,\mathrm{I}_d)$. How L scales with d?
- Good news: one can try to adopt $\lambda(x)$, for example, with VI or normalising flow (see next lecture and seminar) .

Example: Langevin Dynamics

Itô SDE:

$$\mathrm{d}\theta_t = -\nabla U(\theta_t)\,\mathrm{d}t + \sqrt{2}\mathrm{d}W_t,$$

Invariant measure: $\pi(\theta) = e^{-U(\theta)}$

1. First-order discretization (Unadjusted Langevin Algorithm, ULA):

$$Y_{k+1} = Y_k - \gamma\nabla U(Y_k) + \sqrt{2\gamma}Z_{k+1}, \quad i.i.d.\ Z_k \sim \mathrm{N}(0, \mathrm{I}_d)$$

   Equivalently, $Y_{k+1} \sim \mathcal{N}(Y_k - \gamma\nabla U(\theta_k), 2\gamma)$
2. Demo: https://chi-feng.github.io/mcmc-demo
3. If we can't calculate ∇U replace it by its estimate over batch (SGLD, SGLD-FP, SAGA etc)

Analysis of ULA

A1 (can be relaxed)

U is L–smooth and m–strongly convex, that is, $U \in C^2(\mathbb{R}^d)$ and there exists $m, L > 0$ such that

$$\|\nabla U(x) - \nabla U(y)\| \leq L\|x - y\|$$
$$\langle \nabla U(x) - \nabla U(y), x - y \rangle \geq m\|x - y\|^2 .$$

Theorem

Durmus and Moulines [2017] For any $\gamma \in (0, m/L^2)$ there exists π_γ:

$$W_2^2(\delta_x \mathrm{P}_\gamma^k, \pi_\gamma) \leq (1 - m\gamma)^k \int \|x - y\|^2 \pi_\gamma(\mathrm{d}y)$$

Example

Sample normal distribution with ULA

Consider

$$\pi(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} .$$

Write down the sampling scheme with ULA starting from $X_0 = 0$, find the distribution of k-th iterate X_k and identify the limiting distribution π_γ.

Example

Sample normal distribution with ULA

Consider

$$\pi(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} .$$

Write down the sampling scheme with ULA starting from $X_0 = 0$, find the distribution of k-th iterate X_k and identify the limiting distribution π_γ.

Solution

$$X_{k+1} = (1-\gamma)X_k + \sqrt{2\gamma}\xi_{k+1}$$

Hence, the $(k+1)$-th iterate has normal distribution $\mathcal{N}(0, \sigma^2_{k+1})$, where

$$\sigma^2_{k+1} = (1-\gamma)^2\sigma^2_k + 2\gamma = 2\gamma \sum_{m=0}^{k-1} (1-\gamma)^{2m} .$$

Limiting distribution

The limiting distribution is normal $\mathcal{N}(0, \sigma_\gamma^2)$ with

$$\sigma_\gamma^2 = \frac{2\gamma}{1-(1-\gamma)^2} = \frac{1}{1-\gamma/2}\,.$$

Recap: Metropolis-Hastings algorithm

Let $Q(x, A) = \int_A q(x, y)\mathrm{d}y$ be some MK (e.g. Gaussian)

1. Choose X_0.
2. Given X_k, a candidate move Y_{k+1} is sampled from $Q(X_k, \cdot)$
3. $X_{k+1} = Y_{k+1}$ with probability $\alpha(X_k, Y_{k+1})$, otherwise $X_{k+1} = X_k$, where acceptance ratio

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right\}$$

Example: Random walk MH
Take $q(x, y) = \overline{q}(y - x)$, where $\overline{q}(x) = \overline{q}(-x)$. Then

$$Y_{k+1} = X_k + Z_{k+1}, \quad Z_{k+1} \sim \overline{q}$$

In this case

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}$$

MALA

- Metropolis-adjusted Langevin Algorithm (MALA): ULA + Metropolis-Hastings correction;
- Demo: https://chi-feng.github.io/mcmc-demo

Hamiltionian Monte-Carlo (HMC)

- Following Neal [2011], introduce an auxiliary momentum variable r_i for each model variable θ_i, $i \in \{1, \ldots, d\}$;
- Consider the (unnormalized) joint density

$$p(\theta, r) \propto \exp\{-U(\theta) - \tfrac{1}{2} r^\top r\}, (\theta, r) \in \mathbb{R}^{2d}. \tag{2}$$

- We aim at sampling from the joint density $p(\theta, r)$, despite we are interested only in the θ marginal;
- $\theta \in \mathbb{R}^d$ - particle's position; r - momentum; $U(\theta)$ - potential energy, $\frac{1}{2} r^\top r$ is the kinetic energy of the particle.

HMC

To simulate the evolution of the system over time, we can use the *Leapfrog integrator*

Leapfrog$(\theta_t, r_t, \epsilon)$

1. $r_{t+\epsilon/2} = r_t - (\epsilon/2)\nabla_\theta U(\theta_t)$;
2. $\theta_{t+\epsilon} = \theta_t + \epsilon r_{t+\epsilon/2}$;
3. $r_{t+\epsilon} = r_{t+\epsilon/2} - (\epsilon/2)\nabla_\theta U(\theta_{t+\epsilon})$.

In the above r_t and θ_t denote the values of the momentum and position variables r and θ at time t.

Hamiltionian Monte-Carlo (HMC): algorithm

Algorithm 2: Hamiltonian Monte Carlo

Input : θ^0, ϵ, L, $U(\theta)$, n:
Output: New sample Y_{j+1}

for $k = 1$ *to* n **do**
  Sample $r_0 \sim \mathcal{N}(0, \mathrm{I}_d)$;
  Set $\theta_k \leftarrow \theta_{k-1}, \tilde{\theta} \leftarrow \theta_{k-1}, \tilde{r} \leftarrow r_0$;
  for $i = 1$ *to* L **do**
    Set $\tilde{\theta}, \tilde{r} \leftarrow \mathit{Leapfrog}(\tilde{\theta}, \tilde{r}, \epsilon)$;
  With probability $\alpha = \min\left\{1, \frac{\exp\{-U(\tilde{\theta}) - \frac{1}{2}\tilde{r}^\top \tilde{r}\}}{\exp\{-U(\theta_{k-1}) - \frac{1}{2} r_0^\top r_0\}}\right\}$, accept
  $\theta_k \leftarrow \tilde{\theta}$, $r_k \leftarrow -\tilde{r}$.

HMC parameters

- What if ϵ is too large?

HMC parameters

- What if ϵ is too large?
- Acceptance rate is low, and the performance degrades;

HMC parameters

- What if ϵ is too large?
- Acceptance rate is low, and the performance degrades;
- What if ϵ is too small?

HMC parameters

- What if ϵ is too large?
- Acceptance rate is low, and the performance degrades;
- What if ϵ is too small?
- Same problems as ULA, HMC becomes computationally costly and produces correlated particles (can be partially compensated with L);
- Demo: https://chi-feng.github.io/mcmc-demo

To be continued...

Thank you!

References

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

Christophe Andrieu, Anthony Lee, Matti Vihola, et al. Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872, 2018.

Alain Durmus and Eric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 2017.

R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162, 2011.

Donald B Rubin. Comment: A noniterative Sampling/Importance Resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82 (398):542–543, 1987.