Lecture 27: Physics-informed deep neural networks

Professor Ilias Bilionis

The Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm

$$p(x) = \frac{\pi(x)}{Z}$$

- **Initialize:** x_0
- For $n = 1, 2, \ldots$
  - **Generate** candidate sample: $x' \sim q(x' \mid x_{n-1})$
  - **Calculate** the *acceptance ratio*:

$$a(x', x_{n-1}) = \max\left\{1, \frac{\pi(x')}{\pi(x_{n-1})} \cdot \frac{q(x_{n-1} \mid x')}{q(x' \mid x_{n-1})}\right\}$$

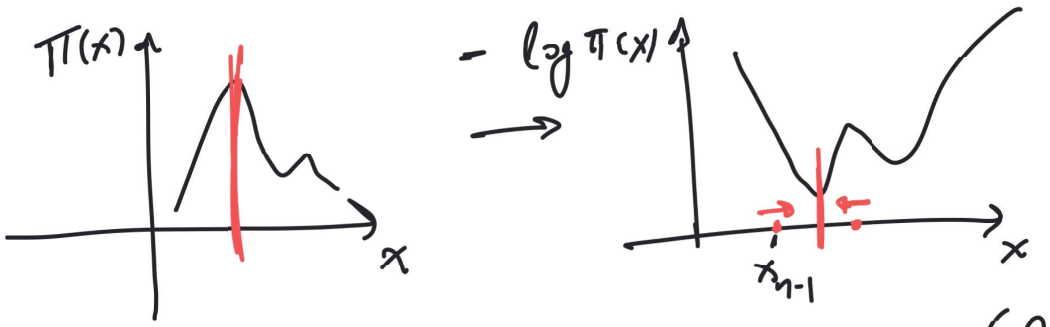
  - **Accept/Reject:**
    - Generate $u \sim U([0,1])$
    - If $u \leq a$, accept : $x_n \leftarrow x'$
    - If $u > a$, reject : $x_n \leftarrow x_{n-1}$

The Metropolis-Hastings algorithm is not one algorithm…

- For every choice of proposal, a different algorithm.
- Metropolis Adjusted Langevin Dynamics
- Hamiltonian Monte Carlo
- Riemannian Manifold Hamiltonian Monte Carlo
- …
- Meta-algorithms: No-U-Turn sampler (NUTS)

Building proposals from the gradient of the target density

$$p(x) = \frac{\pi(x)}{Z}$$



$$-V(x) = -\log \pi(x) \quad \text{(Potential energy)}$$

$$F = -\nabla_x V(x_{n-1}) = \nabla_x \log \pi(x_{n-1})$$

$$q(x' \mid x_{n-1}) = N(x' \mid \beta \nabla_x \log \pi(x_{n-1}), \Sigma)$$

$$\overset{\text{Common}}{=} N(x' \mid \Delta t \nabla_x \log \pi(x_{n-1}), \sqrt{2\Delta t}\, I)$$

Metropolis-Adjusted Langevin Dynamics