Efficient Sampling Techniques

Alexey Naumov

HDI Lab,
HSE University

NATIONAL RESEARCH
UNIVERSITY

June 20-25, 2022

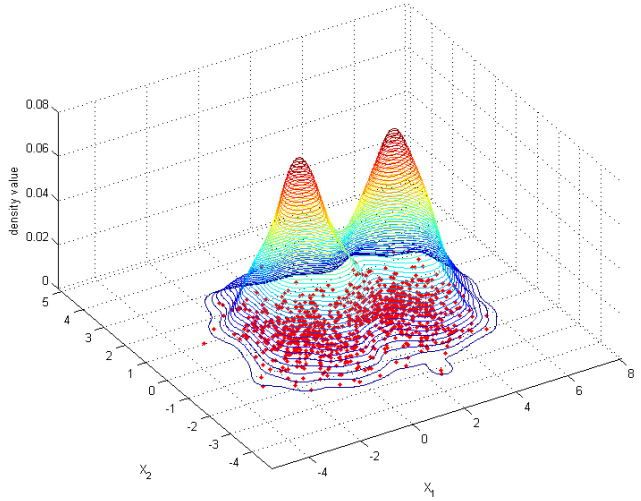
Summary

Density estimation

- Classical statistical problem:
  1. We have a sample $X_1, \ldots, X_n \in \mathbb{R}^d$ from a density $p_{\text{data}}(x)$.
  2. Aim: estimate $p_{\text{data}}(x)$ and sample from it
- Classical solution: kernel density estimation

$$\pi(x) = \frac{1}{n} \sum_{j=1}^{n} K_h(X_j - x),$$

where K_h – kernel, h – bandwidth.



- This approach work when $d = 1, 2, 3$.

Density estimation

- High dimension $d > 3$.
- Black and white pictures 1024×1024 pixels, dim $d = 2^{20} > 10^6$.
- Other object of interest: video, protein structure, ...
- We need other methods (e.g. GANs)
- How to sample from π?

Motivation

- Bayesian inference and learning. Let $\theta \in \Theta$ be an unknown variable (parameter) and $\mathbf{X} = (X_1, \ldots, X_N) \in \mathsf{X}$ be a data.
  1. Posterior distribution: given the prior $\mathrm{p}_0(\theta)$ and likelihood $p(X_i|\theta)$

$$\pi(\theta|\mathbf{X}) = \frac{\prod_{i=1}^{N} p(X_i|\theta)\, \mathrm{p}_0(\theta)}{\int\limits_{\Theta} \prod_{i=1}^{N} p(X_i|\theta)\, \mathrm{p}_0(\theta) \mathrm{d}\theta}$$

  2. Expectation w.r.t. $\pi(\theta|\mathbf{X})$

$$\mathbb{E}_{\pi(\cdot|\mathbf{X})}[f(\theta)] = \int_{\Theta} f(\theta)\pi(\theta|\mathbf{X})\mathrm{d}\theta$$

- Statistical mechanics. Here, one needs to compute the partition function Z of a system with states s and Hamiltonian $E(s)$

$$Z = \sum_{s} \exp\left\{-\frac{E(s)}{kT}\right\},$$

  where k is the Boltzmann's constant and T denotes the temperature of the system.

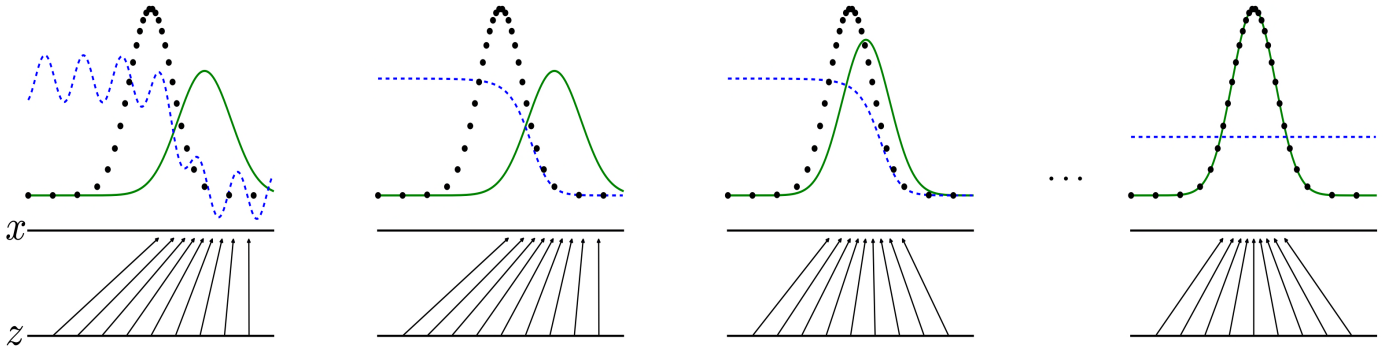
GANs framework

- Generator $G : \mathbb{R}^d \mapsto \mathbb{R}^{\mathrm{D}}$: takes a latent variable z from a prior density $p_0(z)$, $z \in \mathbb{R}^d$, produces $G(z) \in \mathbb{R}^{\mathrm{D}}$ in the observation space;
- Discriminator $D : \mathbb{R}^{\mathrm{D}} \mapsto [0, 1]$: takes a sample in the observation space, distinguishes between real examples and fake ones;

GAN training objective

$$L(g, D) := \mathbb{E}_{X \sim p_{\text{data}}}[\log(D(X))] + \mathbb{E}_{Z \sim p_0}[\log(1 - D(g(Z)))] \to \min_{g \in \mathcal{G}} \max_{D \in \mathcal{D}} .$$

- Let $p_d(x)$ and $p_g(x)$ be the densities of real and fake observations;



- $$\text{Optimal discriminator: } D^\star(x) = \frac{p_d(x)}{p_d(x) + p_g(x)} \tag{1}$$

GANs as an energy-based model

- Main drawback: information accumulated by discriminator is not used during the generation procedure;
- Let $d^\star(x) = \text{logit}\, D^\star(x)$, therefore:

$$\frac{p_d(x)}{p_d(x) + p_g(x)} = \frac{1}{1 + \frac{p_g(x)}{p_d(x)}} = \frac{1}{1 + \exp(-d^\star(x))}$$

  Hence, we can express

$$p_d(x) = p_g(x) e^{d^\star(x)} \,.$$

- Let us introduce $d(x) = \text{logit}\, D(x)$ and consider the corresponding energy-based model

$$\hat{p}_d(x) = p_g(x) e^{d(x)} / Z_0 \,,$$

  where Z_0 is the normalizing constant. If $D(x) \approx D^\star(x)$, $\hat{p}_d(x)$ is close to $p_d(x)$;
- Sample from $\hat{p}_d(x)$ using MCMC.

GANs as an energy-based model

- Similar idea considered in Turner et al. [2019]; main issue: MCMC in pixel space is highly inefficient;
- Che et al. [2020] suggested latent-space sampling from the model

$$\hat{p}_d(x) = p_0(z) \exp\left\{\text{logit}\big(D(G(z))\right\}, z \in \mathbb{R}^d,$$

  where $p_0(z)$ is the generator's prior distribution in the latent space;
- Sampling using Langevin-based algorithms, as suggested in Che et al. [2020], can be inefficient, especially if d is large.

Monte-Carlo method

- Get an i.i.d. sample $(X_k)_{k=0}^{\infty}$ from π, estimate $\pi(f)$ by

$$\pi_n(f) := \frac{1}{n}\sum_{k=0}^{n-1} f(X_k),$$

- Kolmogorov's strong law of large numbers: with probability 1

$$\lim_{n\to\infty} \pi_n(f) = \mathbb{E}[f(X_0)] = \pi(f)$$

- Advantage over deterministic integration: MC positions the integration grid (samples) in regions of high probability.
- Disadvantage: when $\pi(x)$ has standard form, e.g. Gaussian, it is straightforward to sample from it using easily available routines. However, when this is not the case, we need to introduce more sophisticated techniques.

Monte-Carlo method

- Variance:

$$\mathrm{Var}[\pi_n(f)] = \frac{1}{n^2}\sum_{k=0}^{n-1}\mathrm{Var}[f(X_k)] = \frac{\sigma_\pi^2(f)}{n}$$

  where $\sigma_\pi^2(f) = \mathrm{Var}[f(X_0] = \pi(f^2) - \pi^2(f)$.

- Central limit theorem (CLT)

$$\sqrt{n}(\pi_n(f) - \pi(f)) \xrightarrow{Law} \mathrm{N}(0, \sigma_\pi^2(\mathrm{f})) \quad \mathrm{n} \to \infty$$

  Indeed,

$$\sqrt{n}(\pi_n(f) - \pi(f)) = \frac{\sum_{k=0}^{n-1}(f(X_k) - \mathbb{E}[f(X_k)])}{\sqrt{n}}$$

- Length of confidence interval for $\pi_n(f)$ proportional to $\frac{\sigma_\pi(f)}{\sqrt{n}}$

Variance minimization

Variance

$$\mathrm{Var}[\pi_n(f)] = \frac{1}{n^2}\sum_{k=0}^{n-1}\mathrm{Var}[f(X_k)] = \frac{\sigma_\pi^2(f)}{n}$$

where $\sigma_\pi^2(f) = \mathrm{Var}[f(X_0)] = \pi(f^2) - \pi^2(f)$. How to decrease variance?

- Increase n. Not an option in many situations.
- Control variates: replace f by $f - g$, where $\pi(g) = 0$. Denote by $\mathcal{G} := \{g : \pi(g) = 0\}$. Find

$$\hat{g}_n := \arg\min_{g\in\mathcal{G}} V_n(f - g),$$

where

$$V_n(f - g) = \frac{1}{n-1}\sum_{k=0}^{n-1}(f(X_k) - g(X_k) - \pi_n(f) - \pi_n(g))^2$$

See [**?**].

Variance minimization

- Let $\pi(x) = \mathrm{e}^{-U(x)}$.
- Take Stein's control variates

$$g_\phi(x) = -\langle \phi(x), \nabla U(x) \rangle + \mathrm{div}\big(\phi(x)\big), \tag{2}$$

- Let $\mathsf{X} = \mathbb{R}$. Then (under some technical assumptions)

$$\begin{aligned}\int_{\mathbb{R}} g_\phi(x)\mathrm{d}x &= -\int \phi(x)U'(x)\mathrm{e}^{-U(x)}\mathrm{d}x + \int_{\mathbb{R}} \phi'(x)\mathrm{e}^{-U(x)}\mathrm{d}x \\ &= \int \phi(x)\mathrm{d}(\mathrm{e}^{-U(x)}) + \int_{\mathbb{R}} \phi'(x)\mathrm{e}^{-U(x)}\mathrm{d}x = 0\end{aligned}$$

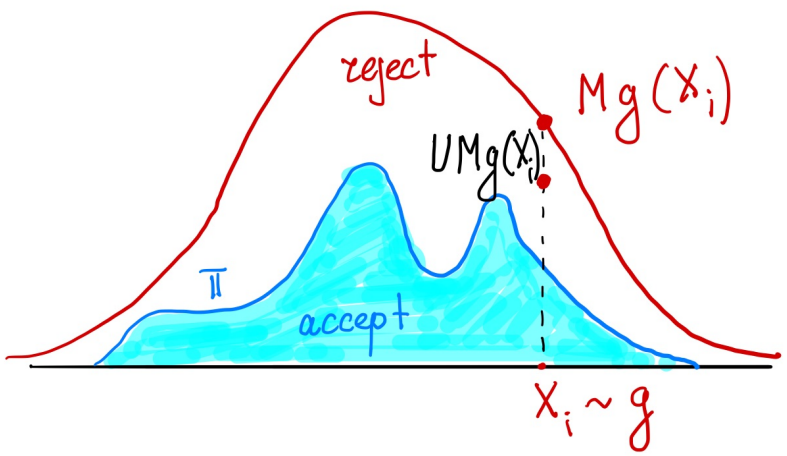
- Exercise: consider $\mathsf{X} = \mathbb{R}^{\mathsf{d}}, \mathsf{d} > 1$.
- More details on variance minimization in the talk by Leonid Iosipoi and Sergey Samsonov.

Rejection sampling

- Sample from a distribution π, which is known up to a proportionality constant, by sampling from another easy-to-sample proposal distribution g that satisfies $\pi(x) \leq Mg(x), M < \infty$.
- Algorithm:
  Set $k = 0$;
  Repeat until $k = n - 1$
  1. Sample $X_i \sim q$ and independent $U \sim \text{Uniform}[0, 1]$;
  2. Accept X_i and set $i := i + 1$, if

$$U < \frac{\pi(X_i)}{Mg(X_i)}.$$

  Otherwise, reject.



Rejection sampling

- Advantage: simple
- Disadvantage: impractical in high-dimensional scenarios.
  It is not always possible to bound $\pi(x)/g(x)$ with a reasonable constant M over the whole space X. If M is too large,

$$\begin{aligned}\mathsf{P}(X_i \text{ accepted}) &= \mathsf{P}\left(U < \frac{\pi(X_i)}{Mg(X_i)}\right) = \mathbb{E}\left[\mathsf{P}\left(U < \frac{\pi(X_i)}{Mg(X_i)}\right)\middle| X_i\right] \\ &= \mathbb{E}\left[\frac{\pi(X_i)}{Mg(X_i)}\right] = \int_{\mathsf{X}} \frac{\pi(x)}{Mg(x)} g(x)\mathrm{d}x = \frac{1}{M}\end{aligned}$$

  will be too small (here we also assume $g(x) > 0, x \in \mathsf{X}$)

Rejection sampling

We show that

$$\mathrm{P}\left(X_i \leq x | U < \frac{\pi(X_i)}{Mg(X_i)}\right) = \pi\{(-\infty, x]\}$$

Indeed, let $A = \{X_i \leq x\}, B = \left\{U < \frac{\pi(X_i)}{Mg(X_i)}\right\}$. Then

$$\mathrm{P}(A|B) = \mathrm{P}(B|A)\,\mathrm{P}(A)/\,\mathrm{P}(B).$$

We may check that

$$\begin{aligned}
\mathrm{P}(B|A) &= \frac{\mathrm{P}(A \cap B)}{G(x)} = \frac{1}{G(x)}\,\mathbb{E}[\mathbb{1}_{A\cap B}] \\
&= \frac{1}{G(x)}\,\mathbb{E}_{X_i}[\mathbb{1}_A]\,\mathbb{E}_U[\mathbb{1}_B] = \frac{1}{MG(x)}\,\mathbb{E}_{X_i}\left[\mathbb{1}_A \frac{\pi(X_i)}{g(X_i)}\right] \\
&= \frac{\pi\{(-\infty, x]\}}{MG(x)}.
\end{aligned}$$

Importance sampling

- Make change of measure: replace $\pi(x)$ by another easy-to-sample proposal distribution $\lambda(x)$:

$$\pi(f) = \int_{\mathsf{X}} f(x)\pi(x)\mathrm{d}x = \int_{\mathsf{X}} f(x)w(x)\lambda(x)\mathrm{d}x,$$

where $w(x)$ – importance weight (Radon-Nikodym derivative)

$$w(x) := \frac{\pi(x)}{\lambda(x)}$$

- Replace $\pi_n(f)$ by $\bar{\pi}_n(f)$,

$$\bar{\pi}_n(f) := \frac{1}{n}\sum_{k=0}^{n-1} f(X_i)w(X_i),$$

where $X_i \sim \lambda$.

Importance sampling

- Variance

$$\mathrm{Var}_\lambda[f(X_0)w(X_0)] = \mathbb{E}_\lambda[f^2(X_0)w^2(X_0)] - \pi^2(f)$$

- By Jensen's inequality

$$\mathbb{E}_\lambda[f^2(X_0)w^2(X_0)] \geq (\mathbb{E}_\lambda[|f(X_0)|w(X_0)])^2 = \left(\int_{\mathrm{X}} |f(x)|\pi(x)\mathrm{d}x\right)^2$$

- Lower bound is attained for

$$\lambda^*(x) = \frac{|f(x)|\pi(x)}{\int_{\mathrm{X}} |f(x)|\pi(x)\mathrm{d}x}$$

- High sampling efficiency is achieved when we focus on sampling from π in the importance regions where $|f(x)|\pi(x)$ is relatively large.

Self-Normalized Importance Sampling

- π is known up to a normalizing factor Z_π, $\pi(\mathrm{d}x) = \tilde{\pi}(\mathrm{d}x)/\mathsf{Z}_\Pi$;
- Define *importance weights* as $\tilde{w}(x) = \tilde{\pi}(x)/\lambda(x)$;
- Then

$$\pi(f) = \int f(x)\pi(x)\mathrm{d}x = \mathsf{Z}_\pi^{-1} \int f(x)\tilde{w}(x)\lambda(x)\mathrm{d}x$$
$$= \mathsf{Z}_\pi^{-1} \int f(x)\tilde{w}(x)\lambda(x)\mathrm{d}x \Big/ \left\{ \mathsf{Z}_\pi^{-1} \int \tilde{w}(x)\lambda(x)\mathrm{d}x \right\}$$

- 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

$$X_i \sim \lambda\,, \omega_N^i = \frac{\tilde{w}(X_i)}{\sum_{j=1}^{N} \tilde{w}(X_j)}\,, i \in \{1, \ldots, N\}\,.$$

MCMC

- What can be done if drawing i.i.d. samples from π is not an option?
- If we run the (ergodic) Markov chain $(Z_k)_{k\geq 0}$ for a long time (started from anywhere), then for large N the distribution of Z_N will be approximately invariant: $\mathrm{Law}(Z_N) \approx \pi$. We can then set $X_1 = Z_N$, and then restart and rerun the Markov chain to obtain $X_2, X_3, \ldots$, and then do estimates as in MC,

$$\pi_n(f) = \frac{1}{n} \sum_{k=0}^{n-1} f(X_k)$$

Important question

How to construct $\mathrm{P}(x, A)$ such that the distribution of X_n converges to invariant distribution π as quickly as possible for arbitrary initial distribution ξ?

Markov chains

What to read?

For more details see Douc et al. [2018]

Define a Markov chain (i.e., discrete time).

Ingredients of the definition:

- X – state space (e.g. $\mathsf{X} \subset \mathbb{R}^{\mathsf{d}}$), $\mathcal{X}$ – σ-algebra of X
- Initial distribution $X_0 \sim \xi$;
- Transition kernel $\mathsf{P}(x, A)$, where $x \in \mathsf{X}, \mathsf{A} \in \mathcal{X}$:

$$\mathsf{P}(X_{n+1} \in A | X_n = x) = \mathsf{P}(x, A)$$

- Markov property: X_{n+1} depends only on X_n;

Example: Model $X_0 \sim \xi$ and for $n \geq 1$

$$X_n = F(X_{n-1}, \varepsilon_n)$$

where $(\varepsilon_n)_{n \geq 1}$ is an i.i.d. sequence independent of $\sigma\{X_k, 0 \leq k \leq n-1\}$ and F is some function, $F : \mathsf{X} \times \mathbb{R}^{\mathsf{d}'} \to \mathsf{X}$

Markov chains: gym

- More about MK kernels
- Ergodicity (finite case)
- Ergodicity (not in this course:( )
- Ready for MCMC

Markov chains

Action on measures

Let μ be a probability measure on X

$$\mu P(A) = \int_{\mathsf{X}} \mu(\mathrm{d}x)\, \mathsf{P}(x, A)$$

Action on functions

$$\mathsf{P}\, f(x) = \int_{\mathsf{X}} f(y)\, \mathsf{P}(x, \mathrm{d}y)$$

Composition of kernels

$$\mathsf{P}^n(x, A) = \int_{\mathsf{X}} \mathsf{P}(x, \mathrm{d}y)\, \mathsf{P}^{n-1}(y, A)$$

(Kolmogorov-Chapman equation)

Markov chains

Tensor product (kernel $\otimes$ kernel)

$$\begin{aligned} \mathsf{P} \otimes \mathsf{P}\, f(x) &= \int_{\mathsf{X}} \mathsf{P}(x, \mathrm{d}y) \int_{\mathsf{X}} f(y,z)\, \mathsf{P}(y, \mathrm{d}z) \\ &= \int_{\mathsf{X}\times\mathsf{X}} f(y,z)\, \mathsf{P}(x, \mathrm{d}y)\, \mathsf{P}(y, \mathrm{d}z) \end{aligned}$$

Take $f(y,z) = 1(y \in A, z \in B)$. Then

$$\mathsf{P} \otimes \mathsf{P}\, f(x) = \mathsf{P}(X_1 \in A, X_2 \in B | X_0 = x) = \mathsf{P}^{\otimes 2}(x, A \times B)$$

Tensor product (measure $\otimes$ kernel)

$$\begin{aligned} \xi \otimes \mathsf{P}\, f &= \int_{\mathsf{X}} \xi(\mathrm{d}y) \int_{\mathsf{X}} f(y,z)\, \mathsf{P}(y, \mathrm{d}z) \\ &= \int_{\mathsf{X}\times\mathsf{X}} f(y,z) \xi(\mathrm{d}y)\, \mathsf{P}(y, \mathrm{d}z) \end{aligned}$$

Markov chains

Invariant distribution

Distrbution π is invariant w.r.t. P if

$$\pi P = \pi$$

Theorem

Let $(X_k)_{k=0}^{\infty}$ be a MC with initial distribution π and kernel P*. $(X_k)_{k=0}^{\infty}$ is stationary iff π is invariant.*

Proof.

Let $(X_k)_{k=0}^{\infty}$ be stationary. Then $\mathrm{Law}(X_1) = \mathrm{Law}(X_0)$. Hence,
$\pi\,\mathrm{P}(A) = \mathrm{P}_\pi(X_1 \in A) = \mathrm{P}(X_0 \in A) = \pi(A)$.
If π is invariant, then the distribution of $(X_n, \ldots, X_{n+k})$ is
$\pi\,\mathrm{P}^n \otimes \mathrm{P}^{\otimes k} = \pi \otimes \mathrm{P}^{\otimes k}$ is independent of n $\square$

Markov chains

Reversibility

Distribution ξ is reversible w.r.t. P if

$$\xi \otimes \mathsf{P}(A \times B) = \xi \otimes \mathsf{P}(B \times A)$$

- If X is countable,

$$\xi(x)\,\mathsf{P}(x, x') = \xi(x')\,\mathsf{P}(x', x)$$

  Detailed balance equation.

-

$$\begin{aligned}\mathbb{E}_\xi[f(X_0, X_1)] &= \int_{\mathsf{X}\times\mathsf{X}} \xi(\mathrm{d}x_0)\,\mathsf{P}(x_0, \mathrm{d}x_1) f(x_0, x_1) \\ &= \int_{\mathsf{X}\times\mathsf{X}} \xi(\mathrm{d}x_0)\,\mathsf{P}(x_0, \mathrm{d}x_1) f(x_1, x_0) = \mathbb{E}_\xi[f(X_1, X_0)]\end{aligned}$$

  Hence, $\mathsf{Law}(X_0, X_1) = \mathsf{Law}(X_1, X_0)$

Markov chains

Theorem

Let P *be a MK. If* ξ *is reversible w.r.t.* P *then* ξ *is invariant.*

Proof.

$$\begin{aligned}\xi\,\mathsf{P}(A) &= \xi \otimes \mathsf{P}(\mathsf{X} \times \mathsf{A}) = \xi \otimes \mathsf{P}(\mathsf{A} \times \mathsf{X}) \\ &= \int_{\mathsf{X}} \xi(\mathrm{d}x)\,\mathsf{P}(x, \mathsf{X}) 1_{\mathsf{A}}(\mathsf{x}) = \xi(\mathsf{A})\end{aligned}$$

□

Ergodicity, finite case

Let X be finite, $\mathsf{X} = [1, \ldots, \mathsf{r}]$

Total variation distance (finite case)

Let μ, ξ be probability measures on X. Define

$$\mathsf{d}_{\mathsf{TV}}(\xi, \mu) := \frac{1}{2} \sum_{\mathsf{i}=1}^{\mathsf{r}} |\mu(\mathsf{i}) - \xi(\mathsf{i})| = \sum_{\mathsf{i}:\mu(\mathsf{i})>\xi(\mathsf{i})} (\mu(\mathsf{i}) - \xi(\mathsf{i}))$$

Clearly, $\mathsf{d}_{\mathsf{TV}} \leq 1$.

- Denote $J := \{i : \mu\mathsf{Q}(\mathsf{i}) > \xi\mathsf{Q}(\mathsf{i})\}$. Let Q be an arbitrary MK. Then for any μ, ξ

$$\begin{aligned} \mathsf{d}_{\mathsf{TV}}(\mu\mathsf{Q}, \xi\mathsf{Q}) &= \sum_{j \in J} (\mu\mathsf{Q}(\mathsf{j}) - \xi\mathsf{Q}(\mathsf{j})) \\ &= \sum_{j \in J} \sum_{i \in \mathsf{X}} (\mu(i)\mathsf{Q}(\mathsf{i},\mathsf{j}) - \xi(\mathsf{i})\mathsf{Q}(\mathsf{i},\mathsf{j})) \\ &\leq \sum_{i:\mu(i)>\xi(i)} (\mu(i) - \xi(i)) \sum_{j \in J} \mathsf{Q}(\mathsf{i},\mathsf{j}) \leq \mathsf{d}_{\mathsf{TV}}(\mu, \xi) \end{aligned} \tag{3}$$

Ergodicity, finite case

- Let $\mathrm{Q(i,j)} \geq \mathrm{a} > 0$ for any $i, j \in \mathrm{X}$. Then $\exists\, j' \notin J$ and hence for any $i \in \mathrm{X}$
$$\sum_{j \in J} \mathrm{Q(i,j)} < 1 - \mathrm{a}$$
Eq. (3) my be improved:
$$\mathrm{d_{TV}}(\mu \mathrm{Q}, \xi \mathrm{Q}) < (1 - \mathrm{a}) \mathrm{d_{TV}}(\mu, \xi)$$
- Assume
$$\exists\, s : \mathrm{P}^s(x, x') > 0 \text{ for any } x, x' \in \mathrm{X} \tag{4}$$
- Let us fix arbitrary distribution μ_0 and denote $\mu_n = \mu_0 \mathrm{P}^n$. Then
$$\begin{aligned} \mathrm{d_{TV}}(\mu_\mathrm{n}, \mu_{\mathrm{n+k}}) &= \mathrm{d_{TV}}(\mu_0 \mathrm{P}^\mathrm{n}, \mu_0 \mathrm{P}^{\mathrm{n+k}}) \\ &\leq (1 - a) \mathrm{d_{TV}}(\mu_0 \mathrm{P}^{\mathrm{n-s}}, \mu_0 \mathrm{P}^{\mathrm{n+k-s}}) \\ &\leq (1 - a)^m \mathrm{d_{TV}}(\mu_0 \mathrm{P}^{\mathrm{n-ms}}, \mu_0 \mathrm{P}^{\mathrm{n+k-ms}}), \end{aligned} \tag{5}$$
where $m : 0 < n - ms \leq s$. Take n large such that $(1 - a)^m < \varepsilon$. Then $\{\mu_n\}_{n \geq 1}$ is a Cauchy sequence.

Ergodicity, finite case

- Set
$$\pi := \lim_{n\to\infty} \mu_n.$$
Then
$$\pi\,\mathsf{P} = \lim_{n\to\infty} \mu_n\,\mathsf{P} = \lim_{n\to\infty} \mu_0\,\mathsf{P}^{n+1} = \pi$$
- Uniqueness: Assume $\pi_1 \neq \pi_2$ such that $\pi_1\,\mathsf{P} = \pi_1, \pi_2\,\mathsf{P} = \pi_2$. Then $\pi_i = \pi_i\,\mathsf{P}^s, i = 1, 2$ and
$$\mathsf{d}_{\mathsf{TV}}(\pi_1, \pi_2) \leq (1 - \mathsf{a})\mathsf{d}_{\mathsf{TV}}(\pi_1, \pi_2)$$
Hence, $\pi_1 = \pi_2$.
- $$\begin{aligned}\mathsf{d}_{\mathsf{TV}}(\mu_0\,\mathsf{P}^{\mathsf{n}}, \pi) = \mathsf{d}_{\mathsf{TV}}(\mu_0\,\mathsf{P}^{\mathsf{n}}, \pi\,\mathsf{P}^{\mathsf{n}}) &\leq (1 - \mathsf{a})^{\mathsf{m}}\mathsf{d}_{\mathsf{TV}}(\mu_0\,\mathsf{P}^{\mathsf{n}-\mathsf{ms}}, \pi\,\mathsf{P}^{\mathsf{n}-\mathsf{ms}}) \\ &\leq (1 - a)^m \leq (1 - a)^{n/s-1} = (1 - a)^{-1}\beta^n,\end{aligned} \tag{6}$$
where $\beta = (1 - a)^{1/s} < 1$.

Ergodicity, finite case

Theorem

Assume (4) *and let* π *be an invariant distribution. Then for any* $f : \mathsf{X} \to \mathbb{R}$, *with probability* 1*:*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \pi(f)$$

- Compare with SLLN for i.i.d. sequence.

MCMC

- What can be done if drawing i.i.d. samples from π is not an option?
- If we run the (ergodic) Markov chain $(Z_k)_{k\geq 0}$ for a long time (started from anywhere), then for large N the distribution of Z_N will be approximately invariant: $\mathrm{Law}(Z_N) \approx \pi$. We can then set $X_1 = Z_N$, and then restart and rerun the Markov chain to obtain $X_2, X_3, \ldots$, and then do estimates as in MC,

$$\pi_n(f) = \frac{1}{n} \sum_{k=0}^{n-1} f(X_k)$$

Important question

How to construct $\mathrm{P}(x, A)$ such that the distribution of X_n converges to invariant distribution π as quickly as possible for arbitrary initial distribution ξ?

Example: 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

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

Example: Langevin Dynamics

Langevin Dynamics Itô SDE:

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

Invariant measure: $\pi(\theta) = e^{-U(\theta)}$ and $\mathrm{Law}(\theta_t) \to \pi$ as $t \to \infty$.

1. Take $\pi(\theta) = (2\pi)^{-1/2}\mathrm{e}^{-\theta^2/2}$.
2. SDE: $\mathrm{d}\theta_t = \theta_t\,\mathrm{d}t + \sqrt{2}\mathrm{d}W_t$, θ_0 is independent of W. This is Ornstein–Uhlenbeck process
3. Apply Ito's formula to obtain

$$\theta_t = \theta_0\mathrm{e}^{-t} + \sqrt{2}\int_0^t e^{-(t-s)}\mathrm{d}W_s$$

4. Since the Itô integral of deterministic integrand is normally distributed, we readily have

$$\mathrm{Law}(\theta_t) = \mathcal{N}(\theta_0\mathrm{e}^{-t}, 1-\mathrm{e}^{-2t}) \to \mathcal{N}(0,1)$$

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, $Y_{k+1} \sim \mathcal{N}(Y_k - \gamma\nabla U(Y_k), 2\gamma\,\mathrm{I})$
2. Metropolis-adjusted Langevin Algorithm (MALA):
   ULA + Metropolis-Hastings correction;
3. Demo: https://chi-feng.github.io/mcmc-demo
4. If we can't calculate ∇U replace it by its estimate over batch (SGLD, SGLD-FP, SAGA etc)

SGLD

1. Posterior distribution:

$$\pi(\theta|\mathbf{X}) = \frac{\prod_{i=1}^{N} p(X_i|\theta)\pi_0(\theta)}{\int\limits_{\mathbb{R}^d} \prod_{i=1}^{N} p(X_i|\theta)\pi_0(\theta)\, d\theta} \propto \mathrm{e}^{-U(\theta)},$$

   where $U = \log \pi_0(\theta) + \sum_{i=1}^{N} \log p(X_i|\theta)$;
2. A computational bottleneck: calculating the full gradient ∇U scaling proportionally to N can be very time consuming in the "big data" limit;
3. Replace $\nabla U(\theta)$ by an unbiased estimate. This gives rise to the SGLD algorithm, where the parameters are updated according to

$$\begin{aligned} \theta_{k+1} &= \theta_k - \gamma G(\theta_k, S_{k+1}) + \sqrt{2\gamma}\, \xi_{k+1}, \\ G(\theta, S) &= \nabla U_0(\theta) + KM^{-1} \sum\nolimits_{i \in S} \nabla U_i(\theta), \end{aligned} \tag{7}$$

   where each S_{k+1} is a random batch taking values in S_M (here S_M is the set of all subsets S of $\{1, \ldots, N\}$ with $|S| = M$) which is sampled from a uniform distribution over S_M independently of $\mathfrak{F}_k$ (here $(\mathfrak{F}_k)_{k \geq 0}$ is the filtration generated by $\{(\theta_\ell, S_\ell)\}_{\ell \geq 0}$).
4. Note that $\mathbb{E}[G(\theta_k, S_{k+1})|\mathfrak{F}_k] = \nabla U(\theta_k)$ and therefore $G(\theta_k, S_{k+1})$ is an unbiased estimate of $\nabla U(\theta_k)$.

Transition kernel of MH 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

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

MH transition kernel

$$\mathrm{P}(x, A) = \int_A \alpha(x, y)q(x, y)\mathrm{d}y + \overline{\alpha}(x)\delta_x(A),$$

where

$$\overline{\alpha}(x) = \int_{\mathrm{X}} (1 - \alpha(x, y))q(x, y)\mathrm{d}y.$$

Invariance of π

Theorem

Distribution π is reversible w.r.t. P.

Proof.

We need to show that for any $C \in \mathcal{X} \times \mathcal{X}$

$$\int_{\mathsf{X}\times\mathsf{X}} \pi(x)\mathrm{d}x\, \mathsf{P}(x, \mathrm{d}y) 1_C(x, y) = \int_{\mathsf{X}\times\mathsf{X}} \pi(y)\mathrm{d}y\, \mathsf{P}(y, \mathrm{d}x) 1_C(x, y)$$

For any $x, y \in \mathsf{X}$

$$\pi(x)\alpha(x, y)q(x, y) = \{\pi(x)q(x, y)\} \vee \{\pi(y)q(y, x)\} = \pi(y)\alpha(y, x)q(y, x)$$

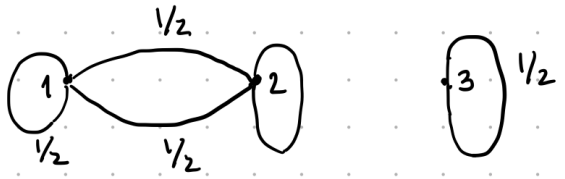
Moreover,

$$\int_{\mathsf{X}\times\mathsf{X}} \pi(x)\mathrm{d}x\delta_x(\mathrm{d}y)\overline{\alpha}(x)1_C(x, y) = \int_{\mathsf{X}} \pi(x)\mathrm{d}x\overline{\alpha}(x)1_C(x, x)$$
$$= \int_{\mathsf{X}} \pi(y)\mathrm{d}y\overline{\alpha}(y)1_C(y, y) = \int_{\mathsf{X}\times\mathsf{X}} \pi(y)\mathrm{d}y\delta_y(\mathrm{d}x)\overline{\alpha}(y)1_C(x, y)$$

□

Ergodicity. General case

- MC could have invariant distribution, but do not converge.
- Example: Let $X = \{1, 2, 3\}$, $\pi(1) = \pi(2) = \pi(3) = 1/3$. Let $P(1,1) = P(1,2) = P(2,1) = P(2,2) = 1/2$, and $P(3,3) = 1$. Then π is invariant.



$$\begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$$

$$\begin{pmatrix} 1/2 & 1/2 & 0 \end{pmatrix} \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix}$$

However, if $X_0 = 1$, then $X_n \in \{1, 2\}$ for all n, so $P_{\delta_1}(X_n = 3) = 0$ for all n, so

$$P_{\delta_1}(X_n = 3) \nrightarrow \pi(3)$$

- (In fact, X_n converges to $\pi(1) = \pi(2) = 1/2$)

Ergodicity. General case

ϕ-irreducubility

MC is ϕ-irreducible if there exists a non-zero σ-finite measure ϕ on X such that for all $A \in \mathcal{X}$ with $\phi(A) > 0$, and for all $x \in \mathrm{X}$, there exists a positive integer $n = n(x, A)$ such that $\mathrm{P}^n(x, A) > 0$.

ϕ-irreducibility of MH

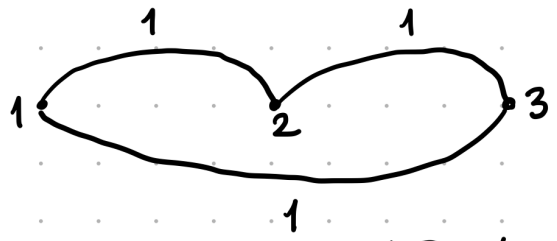
- Assume that $q(x, y) > 0$ is continuous and $\pi(A) = \int_A \pi(x)\,\mathrm{Leb}(\mathrm{d}x) = \int_A \pi(x)\mathrm{d}x$ ($\pi(A) = 0$ if $\mathrm{Leb}(A) = 0$)
- Let $\pi(A) > 0$. Then there exists $R > 0$ such that $\pi(A_R) > 0$, where $A_R = A \cap B_R(0)$, and $B_R(0)$ – the ball of radius R centred at 0. Then by continuity, for any $x \in \mathbb{R}^d$, $\inf_{y \in A_R} \min\{q(x, y), q(y, x)\} \geq \varepsilon$ for some $\varepsilon > 0$, and thus

$$
\begin{aligned}
\mathrm{P}(x, A) \geq \mathrm{P}(x, A_R) &\geq \int_{A_R} q(x, y) \min\left\{1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right\}\mathrm{d}y \\
&\geq \varepsilon\,\mathrm{Leb}\{y \in A_R : \pi(x) \leq \pi(y)\} + \frac{\varepsilon K}{\pi(x)}\pi(\{y \in A_R : \pi(x) > \pi(y)\}) \\
&> 0,
\end{aligned}
$$

where $K = \int_{\mathsf{X}} \pi(x)\mathrm{d}x$.

Ergodicity. General case

- Even ϕ-irreducible chains might not converge in distribution, due to periodicity problems.
- Example: take $\mathsf{X} = \{1, 2, 3\}$, with $\pi(1) = \pi(2) = \pi(3) = 1/3$. Let $\mathsf{P}(1, 2) = \mathsf{P}(2, 3) = \mathsf{P}(3, 1) = 1$. Then π is invariant, and the chain is ϕ-irreducible [e.g. with $\phi(\cdot) = \delta_1(\cdot)$].
  However, if $X_0 = 1$, then $X_n = 1$ whenever $n = 3m$. Hence, $\mathsf{P}_{\delta_0}(X_n = 1) \nrightarrow \pi(3)$, and there is again no convergence to π.



$$\begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$$

Ergodicity. General case

Aperiodicity

A MK P with invariant distribution π is aperiodic if $\nexists d \geq 2$ and disjoint subsets $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_\mathrm{d} \subset \mathrm{X}$ with $\mathrm{P}(x, \mathrm{X}_{\mathrm{i}+1}) = 1$ for all $x \in \mathrm{X}_\mathrm{i}$ $(1 \leq i \leq d-1)$, and $\mathrm{P}(x, \mathrm{X}_1) = 1$ for all $x \in \mathrm{X}_\mathrm{d}$, such that $\pi(\mathrm{X}_\mathrm{i}) > 0$. (Otherwise, the chain is periodic, with period d, and periodic decomposition $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_\mathrm{d}$).

Aperiodicity of MH algorithm

- To see this, suppose that X_1 and X_2 are disjoint subsets of X, $\pi(\mathsf{X}_\mathsf{i}) > 0$, with $\mathsf{P}(x, \mathsf{X}_2) = 1$ for all $x \in \mathsf{X}_1$.
- Take any $x \in \mathsf{X}_1$, then since X_1 must have positive Lebesgue measure,

$$\mathsf{P}(x, \mathsf{X}_1) \geq \int_{\mathsf{X}_1} \mathsf{q}(\mathsf{x}, \mathsf{y}) \alpha(\mathsf{x}, \mathsf{y}) \mathrm{d}\mathsf{y} > 0$$

for a contradiction. Therefore aperiodicity must hold.

Ergodicity. General case

Total variation distance

Let μ, ξ be probability measures on X. Define

$$\mathsf{d}_{\mathsf{TV}}(\xi, \mu) := \sup_{\mathsf{A} \in \mathcal{X}} |\mu(\mathsf{A}) - \xi(\mathsf{A})|$$

Clearly, $\mathsf{d}_{\mathsf{TV}} \leq 1$.

- For any μ, ξ

$$\mathsf{d}_{\mathsf{TV}}(\xi, \mu) = \sup_{\mathsf{f}:\mathsf{X} \to [0,1]} \left| \int_{\mathsf{X}} \mathsf{f}(\mathsf{x}) \mathrm{d}\xi(\mathsf{x}) - \int_{\mathsf{X}} \mathsf{f}(\mathsf{x}) \mathrm{d}\mu(\mathsf{x}) \right|$$

  Prove it.

Ergodicity. General case

Theorem

If a MC on a state space with countably generated σ- algebra is ϕ-irreducible and aperiodic, and has an invariant distribution π, then for π-a.e. $x \in \mathrm{X}$,

$$\lim_{n\to\infty} \mathrm{d_{TV}}(\mathrm{P^n}(\mathrm{x}, \mathrm{A}), \pi) = 0$$

In particular, $\lim_{n\to\infty} \mathrm{P}^n(x, A) = \pi(A)$ *for all* $A \in \mathcal{X}$.
If $\pi(f) < \infty$, *then with probability* 1

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} f(X_k) = \pi(f)$$

- We need more assumptions to quantify convergence (e.g. small set condition)

Analysis of ULA

- Let $\pi(x) = Z_d^{-1} \mathrm{e}^{-U(x)}$;

L–smooth potential

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

$$\|\nabla U(x) - \nabla U(y)\| \leq L\|x - y\|$$

for any $x, y \in \mathbb{R}^d$.

- Unadjusted Langevin Algorithm, ULA:

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

- Denote $\mathrm{P}_\gamma(x, \cdot) = \mathcal{N}(x - \gamma \nabla U(x), 2\gamma\, \mathrm{I})$.

Kantorovich–Wasserstein distance

Kantorovich–Wasserstein distance

For λ, ν, we denote their coupling set by $\Pi(\lambda, \nu)$, i.e. $\xi \in \Pi(\lambda, \nu)$ is the measure on $\mathsf{X} \times \mathsf{X}$ satisfying for all $A \in \mathcal{B}(\mathsf{X})$, $\xi(A, \mathsf{X}) = \lambda(\mathsf{A})$ and $\xi(\mathsf{X}, \mathsf{A}) = \nu(\mathsf{A})$. For $p \geq 1$ and λ, ν, let

$$W_{p,\mathsf{d}}(\lambda, \nu) := \inf_{\Pi(\lambda,\nu)} \left\{ \int_{\mathsf{X}\times\mathsf{X}} \mathsf{d}^p(x, y)\, \xi(\mathrm{d}x, \mathrm{d}y) \right\}^{1/p}$$

be the Kantorovich–Wasserstein distance of order p between λ and ν.

Analysis of ULA

A1

U is L–smooth and m–strongly convex:

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

Theorem

For any $\gamma \in (0, m/L^2)$ there exists invariant distribution π_γ:

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

Analysis of ULA

- Fix $x, \tilde{x} \in \mathbb{R}^d$. Synchronous coupling:

$$X_{k+1} = X_k - \gamma \nabla U(X_k) + \sqrt{2\gamma} Z_{k+1},$$
$$\tilde{X}_{k+1} = \tilde{X}_k - \gamma \nabla U(\tilde{X}_k) + \sqrt{2\gamma} Z_{k+1}$$

- Then

$$\begin{aligned} \|X_{k+1} - \tilde{X}_{k+1}\|^2 = \|X_k - \tilde{X}_k\|^2 \\ \gamma^2 \|\nabla U(X_k) - \nabla U(\tilde{X}_k)\|^2 \\ - 2\gamma \langle X_k - \tilde{X}_k, \nabla U(X_k) - \nabla U(\tilde{X}_k)[] \end{aligned}$$

- Use A1:

$$\begin{aligned} \|X_{k+1} - \tilde{X}_{k+1}\|^2 &\leq (1 + \gamma^2 L^2 - 2\gamma m) \|X_k - \tilde{X}_k\|^2 \\ &\leq (1 - \gamma m) \|X_k - \tilde{X}_k\|^2. \end{aligned}$$

- Hence

$$W_2^2(\delta_x \mathrm{P}_\gamma^k, \delta_{\tilde{x}} \mathrm{P}_\gamma^k) \leq (1 - m\gamma)^k W_2^2(\delta_x, \delta_{\tilde{x}})$$

- We may show that $(\lambda \mathrm{P}_\gamma^k)_{k \in \mathbb{N}}$ is a Cauchy sequence and there exists $\pi_\gamma^\lambda = \pi_\gamma$, moreover $\pi_\gamma \mathrm{P}_\gamma = \pi_\gamma$.

Variance of MCMC estimate

Let π be an invariant distribution. Assume $X_0 \sim \pi$, i.e. we start from the invariant distribution. Then

$$\mathrm{Var}_\pi\left[n^{-1}\sum_{k=0}^{n-1} f(X_k)\right] = \frac{\mathrm{Var}_\pi[f]}{n} + \frac{1}{n^2}\sum_{i\neq j}\mathbb{E}_\pi\left[(f(X_i)-\pi(f))(f(X_j)-\pi(f))\right] =$$

$$= \frac{\rho^{(f)}(0)}{n} + \frac{2}{n}\sum_{k=1}^{n-1}\left(1-\frac{k}{n}\right)\rho^{(f)}(k) \neq \frac{\mathrm{Var}_\pi[f]}{n}$$

where

$$\rho^{(f)}(k) = \mathbb{E}_\pi\left[(f(X_0)-\pi(f))(f(X_k)-\pi(f))\right]$$

Variance of MCMC estimate

- Under appropriate conditions (e.g. ϕ-irreducibility + apereodicity + existence of solution of Poisson eq.) CLT holds:

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} [f(X_i) - \pi(f)] \xrightarrow{Law} \mathcal{N}(0, V_\infty(f)),$$

where $V_\infty(f) := \lim_{n\to\infty} \mathrm{Var}_\pi \left[\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} (f(X_i) - \pi(f)) \right]$

- Length of confidence interval for $\pi_n(f)$ proportional to $\frac{\sqrt{V_\infty(f)}}{\sqrt{n}}$

Ex^{2}MCMC: Sampling through Exploration Exploitation

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 $Y_{j+1} = X_{j+1}^{I_{j+1}}$.

The Markov chain $\{Y_k,\, k \in \mathbb{N}\}$ generated by i-SIR has the following Markov kernel

$$\mathsf{P}_N(x, \mathsf{A}) = \int \delta_x(\mathrm{d}x^1) \sum_{i=1}^N \frac{\tilde{w}(x^i)}{\sum_{j=1}^N \tilde{w}(x^j)} \mathbb{1}_{\mathsf{A}}(x^i) \prod_{j=2}^N \lambda(\mathrm{d}x^j). \tag{8}$$

i-SIR algorithm

- Provided also that $|\tilde{w}|_\infty < \infty$, it was shown in Andrieu et al. [2018] that the Markov kernel P_N is uniformly geometrically ergodic. Namely, for any initial distribution ξ on $(\mathsf{X}, \mathcal{X})$ and $k \in \mathbb{N}$,

$$\|\xi \mathsf{P}_N^k - \pi\|_{\mathrm{TV}} \leq \kappa_N^k, \tag{9}$$

  with $\epsilon_N = \frac{N-1}{2\mathrm{L}+N-2}, \mathrm{L} = |\tilde{w}|_\infty / \lambda(\tilde{w})$ and $\kappa_N = 1 - \epsilon_N$.
- Note that the bound (9) relies significantly on the restrictive condition that weights are uniformly bounded $|\tilde{w}|_\infty < \infty$.
- Moreover, even when this condition is satisfied, the rate κ_N can be close to 1 when the dimension d is large.
- Indeed, consider a simple scenario $\pi(x) = \prod_{i=1}^d p(x_i)$ and $\lambda(x) = \prod_{i=1}^d q(x_i)$ for some densities $p(\cdot)$ and $q(\cdot)$ on $\mathbb{R}$. Then it is easy to see that $\mathrm{L} = (\sup_{y \in \mathbb{R}} p(y)/q(y))^d$ grows exponentially with d.

i-SIR algorithm

To illustrate this phenomenon, we consider a simple problem of sampling from the standard normal distribution $\mathcal{N}(0, \mathrm{I}_d)$ with the proposal $\mathcal{N}(0, 2\,\mathrm{I}_d)$ in increasing dimensions d up to 300.

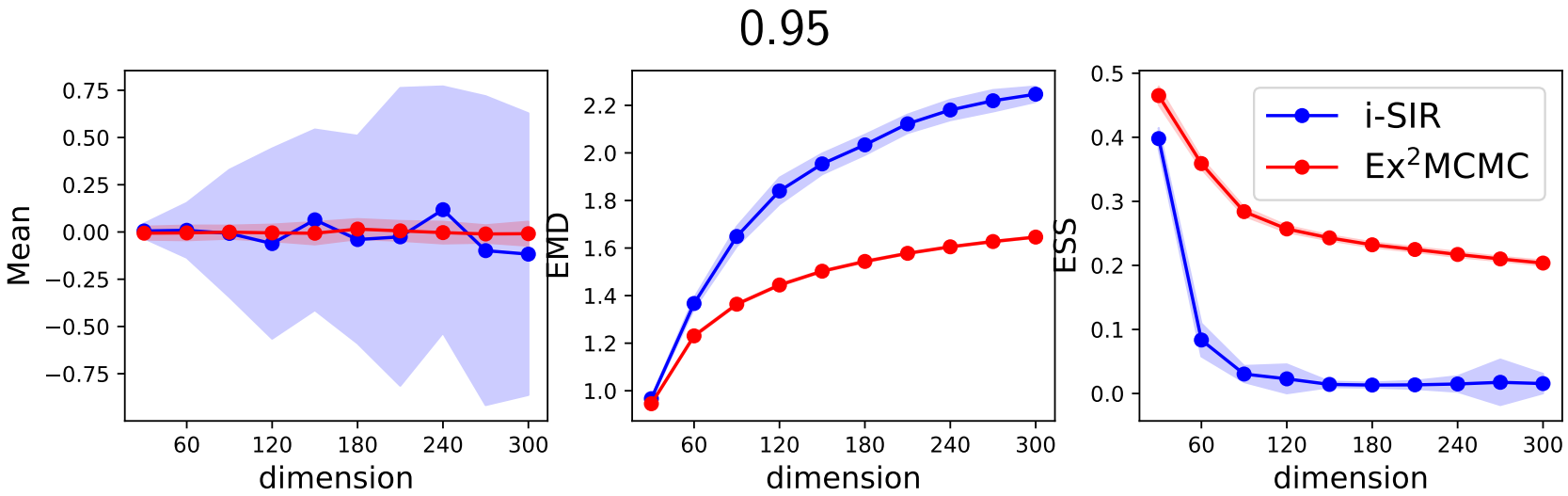


Figure: Sampling from $\mathcal{N}(0, \mathrm{I}_d)$ with the proposal $\mathcal{N}(0, 2\,\mathrm{I}_d)$. We display confidence intervals for i-SIR and Ex^{2}MCMC obtained from 100 independent runs as blue and red regions, respectively. Ex^{2}MCMC helps to achieve efficient sampling even in high dimensions.

Ex^{2}MCMC algorithm

- Main i-SIR drawback: absence of local exploration moves;
- Idea: apply a local MCMC kernel R (*rejuvenation kernel*) after each i-SIR step;
- R has π as invariant distribution;
- Here comes Ex^{2}MCMC : Exploration steps through i-SIR , Exploitation steps through $\mathrm{R}(x, \cdot)$;
- As our default choice we consider MALA as rejuvenation, but other ones (HMC, NUTS) are also possible.

Ex^{2}MCMC algorithm

Algorithm 2: Single stage of Ex^{2}MCMC algorithm with independent proposals

Procedure Ex^{2}MCMC $(Y_j, \Lambda, \mathrm{R})$**:**

Input : Previous sample Y_j;
proposal distribution Λ;
rejuvenation kernel R;

Output: New sample Y_{j+1};

Set $X_{j+1}^1 = Y_j$, 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 $Y_{j+1} \sim \mathrm{R}(X_{j+1}^{I_{j+1}}, \cdot)$.

Ex^{2}MCMC algorithm

V-geometric ergodicity

A Markov kernel Q with invariant probability measure π is V-geometrically ergodic if there exist constants $\rho \in (0,1)$ and $M < \infty$ such that, for all $x \in \mathsf{X}$ and $k \in \mathbb{N}$,

$$\|\mathrm{Q}^k(x,\cdot) - \pi\|_V \leq M\{V(x) + \pi(V)\}\rho^k.$$

Assumptions

A1

(i) R has π as its unique invariant distribution;
(ii) There exists a function $V\colon \mathsf{X} \to [1, \infty)$, such that for all $r \geq r_{\mathsf{R}} > 1$ there exist $\lambda_{\mathsf{R},r} \in [0, 1)$, $\mathsf{b}_{\mathsf{R},r} < \infty$, such that
$\mathsf{R}V(x) \leq \lambda_{\mathsf{R},r} V(x) + \mathsf{b}_{\mathsf{R},r} \mathbb{1}_{\mathsf{V}_r}$, where $\mathsf{V}_r = \{x\colon V(x) \leq r\}$;

A2

(i) For all $r \geq r_{\mathsf{R}}$, $\tilde{w}_{\infty,r} := \sup_{x \in \mathsf{V}_r} \{\tilde{w}(x)/\lambda(\tilde{w})\} < \infty$;
(ii) $\mathrm{Var}_\lambda[\tilde{w}]/\{\lambda(\tilde{w})\}^2 < \infty$.

Ex^{2}MCMC algorithm

Theorem

Let A1 and A2 hold. Then, for all $x \in \mathsf{X}$ and $k \in \mathbb{N}$,

$$\|\mathsf{K}_N^k(x,\cdot) - \pi\|_V \leq c_{\mathsf{K}_N}\{\pi(V) + V(x)\}\tilde{\kappa}_{\mathsf{K}_N}^k\,, \tag{10}$$

where c_{K_N}, $\tilde{\kappa}_{\mathsf{K}_N} \in [0,1)$ are some constants. In addition, $c_{\mathsf{K}_N} = c_{\mathsf{K}_\infty} + O(N^{-1})$ and $\tilde{\kappa}_{\mathsf{K}_N} = \tilde{\kappa}_{\mathsf{K}_\infty} + O(N^{-1})$.

Toy example

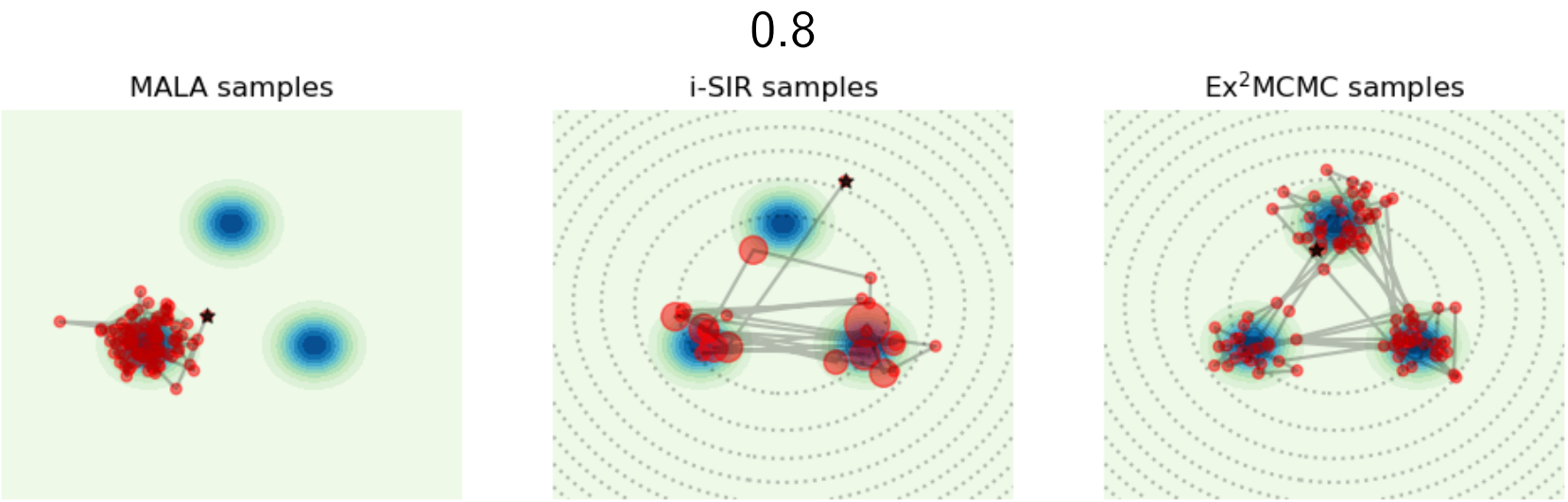


Figure: Single chain mixing visualization. – Blue color levels represent the target 2d density. Random chain initialization is noted in black, 100 steps are plotted per sampler: the size of each red dot corresponds to the number of consecutive steps the walkers remains at a given location. For MALA, we generate 300 samples and choose each 3-rd one for comparability. Note that the variance of the global proposal (dotted countour lines) should be relatively large to cover well all the modes. The step size of MALA also can not be increased much to keep reasonable acceptance ratio.

Adaptive proposals

- Consider family of proposals $\{\lambda_\theta\}, \theta \in \mathbb{R}^D$, chosen to match the target distribution $\tilde{\pi}$;
- Let $T\colon \mathbb{R}^d \to \mathbb{R}^d$ be smooth and invertible. Denote by $T\#\Lambda$ the distribution of $Y = T(X)$ with $X \sim \lambda$;
- The corresponding density is given by $\lambda_T(y) = \lambda\big(T^{-1}(y)\big)\, \mathrm{J}_{T^{-1}}(y)$, where J_T denotes the Jacobian determinant of T;

Adaptive proposals: learning procedure

- Disperancy measure: linear combination of forward and backward KL divergence (generalizations to [Papamakarios et al., 2021b] possible);
- Forward and backward KL:

$$\mathcal{L}^f(\theta) = \int \log \frac{\pi(x)}{\lambda_\theta(x)} \pi(x) \mathrm{d}x \,,$$

$$\mathcal{L}^b(\theta) = \int \log \frac{\lambda(x)}{\pi\big(T_\theta(x)\big)\, \mathrm{J}_{T_\theta}(x)} \lambda(x) \mathrm{d}x \,.$$

- Given a sample $Y_k \sim \pi$ and $Z_k \sim \lambda$ for $k \in [K]$, by

$$\widehat{\nabla \mathcal{L}^f}(Y^{1:K}, \theta) = -\frac{1}{K} \sum_{k=1}^{K} \nabla \log \lambda_\theta(Y_k) \,,$$

$$\widehat{\nabla \mathcal{L}^b}(Z^{1:K}, \theta) = -\frac{1}{K} \sum_{k=1}^{K} \nabla \log\big(\tilde{\pi}(T_\theta(Z_k)\, \mathrm{J}_{T_\theta}(Z_k)\big).$$

- Following Gabrié et al. [2021], we consider

$$\widehat{\mathcal{L}}(Y^{1:K}, Z^{1:K}, \theta) = \alpha \widehat{\mathcal{L}^f}(Y^{1:K}, \theta) + \beta \widehat{\mathcal{L}^b}(Z^{1:K}, \theta) \,.$$

FIEx^{2}MCMC algorithm with adaptive proposals

Algorithm 3: Single stage of FIEx^{2}MCMC. Steps of Ex^{2}MCMC are done in parallel with common values of proposal parameters θ_j. Step 4 updates the parameters using the gradient estimate obtained from all the chains.

Input : weights θ_j, batch $Y_j^{1:K}$

Output: new weights θ_{j+1}, batch $Y_{j+1}^{1:K}$

1 **for** $k \in [K]$ **do**

2 $\quad Y_{j+1,k} = \text{Ex}^2\text{MCMC}\ (Y_{j,k}, T_{\theta_j}\#\Lambda, \text{R})$

3 Draw $\bar{Z}^{1:K} \sim \lambda$.

4 Update $\theta_{j+1} = \theta_j - \gamma\widehat{\nabla\mathcal{L}}(Y_{j+1}, \bar{Z}, \theta_j)$.

Practical note

In our experiments: T_θ is modelled as a normalizing flow based on RealNVP architecture (Dinh et al. [2017]).

Lecture

Hamiltonian Monte-Carlo

I will follow Neal [2011].

Hamiltionian Monte-Carlo (HMC)

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

$$p(\theta, r) \propto \exp\{-U(\theta) - \tfrac{1}{2} r^\top r\}, (\theta, r) \in \mathbb{R}^{2d}. \tag{11}$$

- 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.
- $H(\theta, r) = U(\theta) + \frac{1}{2} r^\top r$ - *Hamiltonian*.

HMC dynamics

Now we consider the evolution of the particle according to the *Hamiltonian dynamics*

$$\begin{cases} \frac{d\theta_i}{dt} & = \frac{\partial H}{\partial r_i}, i \in \{1, \ldots, d\} \\ \frac{dr_i}{dt} & = -\frac{\partial H}{\partial \theta_i} \end{cases} \tag{12}$$

Properties of Hamiltonian dynamics

- Hamiltonian dynamics (12) is reversible: mapping $T_s : (\theta_t, r_t) \mapsto (\theta_{t+s}, r_{t+s})$ is bijective, and has an inverse T_{-s};
- Hamiltonian $H(\theta, r)$ is invariant for the dynamics (12);
- Hamiltonian dynamics is volume-preserving in (θ, r)-space (Liouville's theorem).

HMC

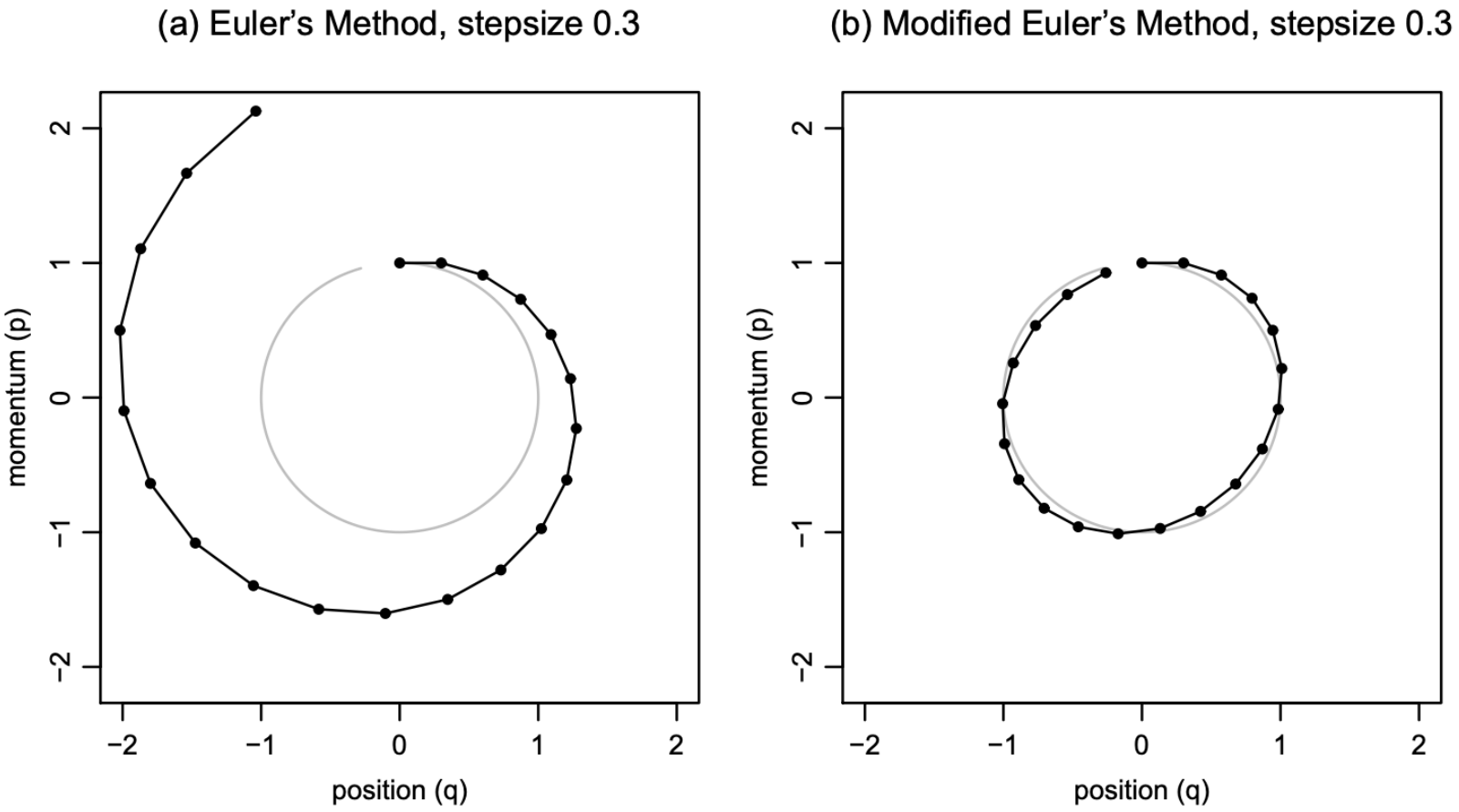
To simulate the evolution of the system over time, we can use the *Euler's method*

Euler's method$(\theta_t, r_t, \epsilon)$

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

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

Different discretizations, Neal [2011]



HMC

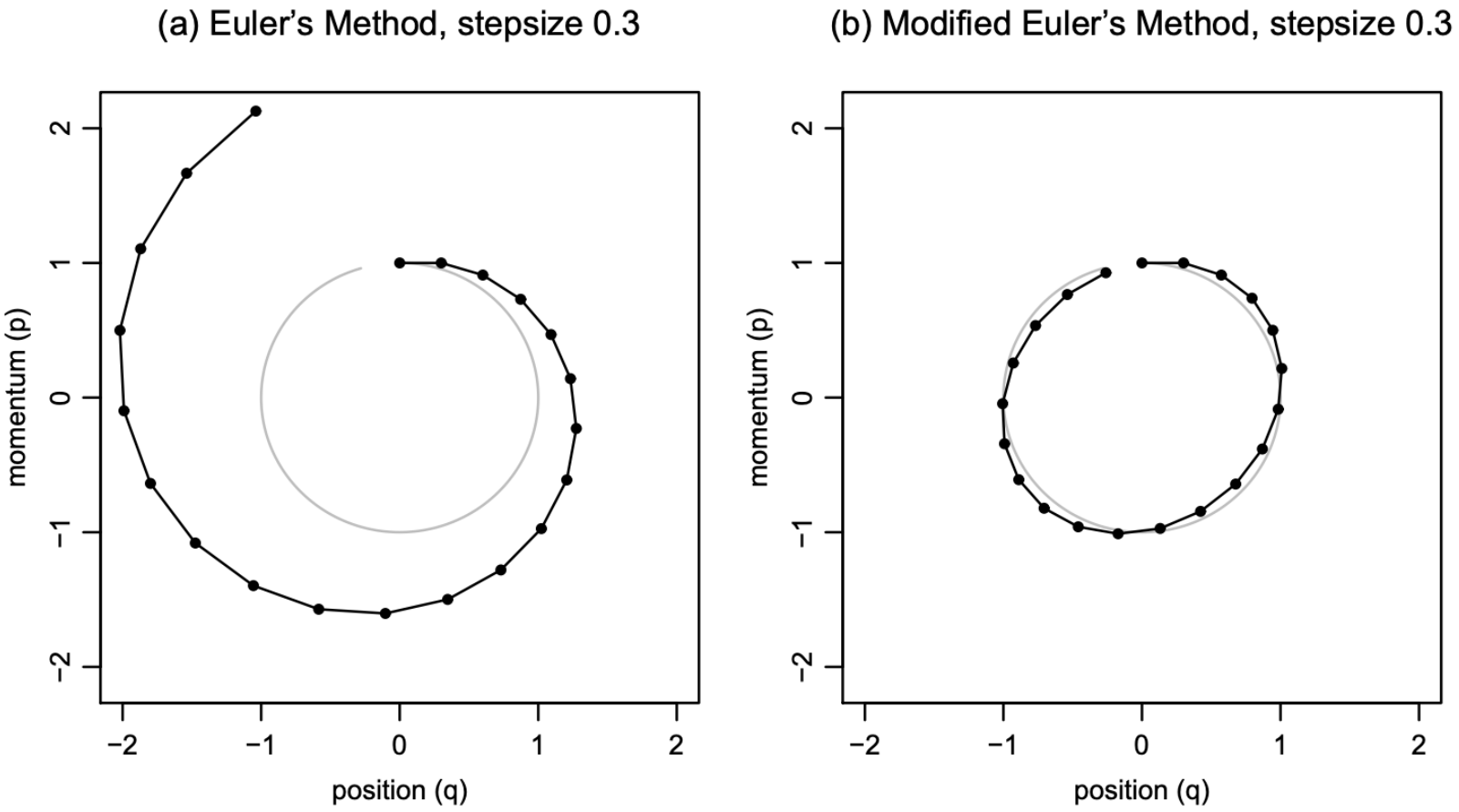
To simulate the evolution of the system over time, we can use the *modification of Euler's method*

Modification of Euler's method$(\theta_t, r_t, \epsilon)$

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

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

Different discretizations, Neal [2011]



HMC

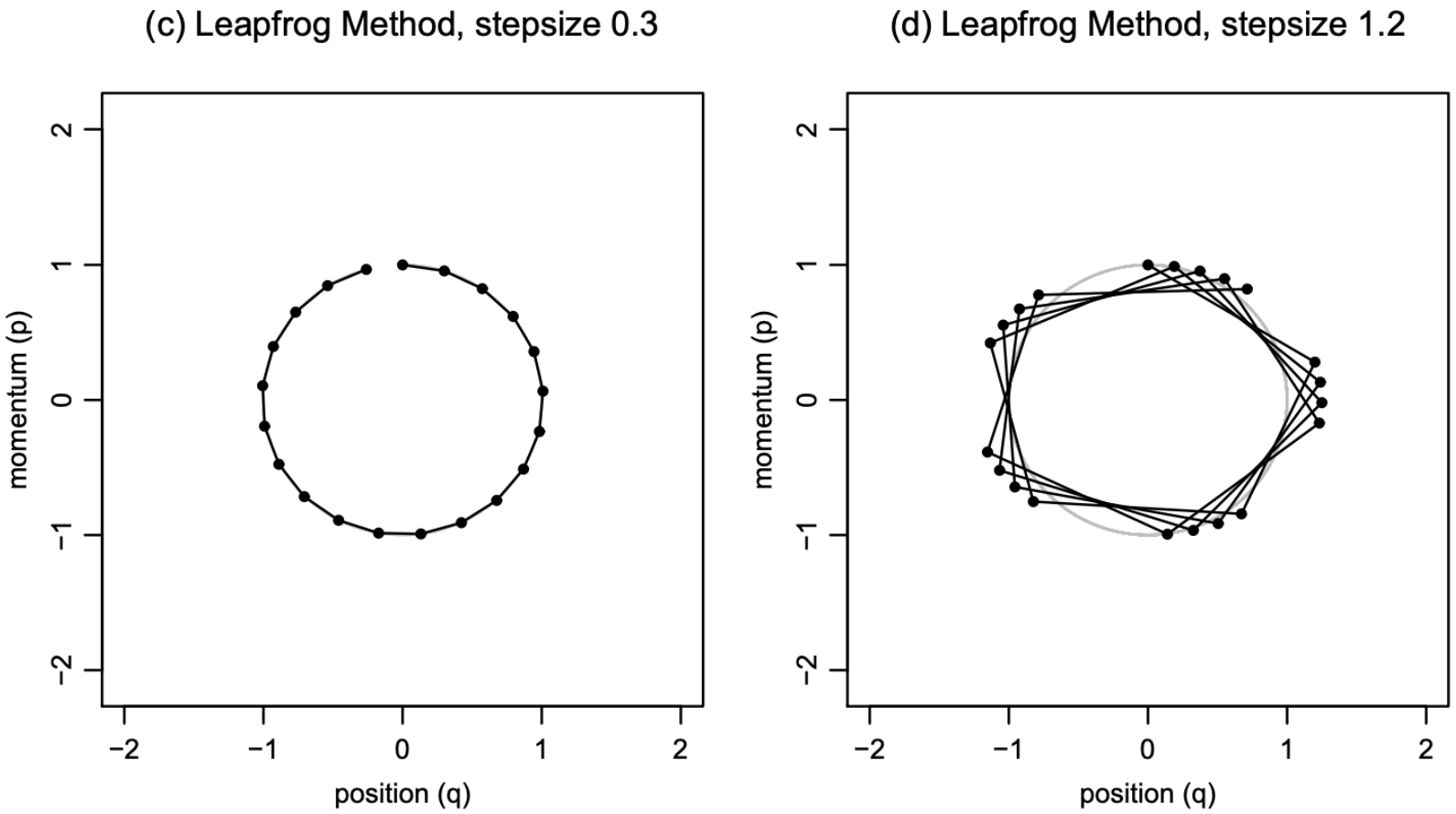
To simulate the evolution of the system over time, we can use the *Leapfrog integrator*

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.

Different discretizations, Neal [2011]



Hamiltionian Monte-Carlo (HMC): algorithm, Hoffman et al. [2014]

Algorithm 4: 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 $\tilde{\theta}, \tilde{r} \leftarrow Leapfrog(\tilde{\theta}, \tilde{r}, \epsilon)$;
  With probability

$$\alpha = 1 \wedge \frac{\exp\{-H(\tilde{\theta}, \tilde{r})\}}{\exp\{-H(\theta_{k-1}, r_{k-1})\}} = 1 \wedge \frac{\exp\{-U(\tilde{\theta}) - \frac{1}{2}\tilde{r}^\top \tilde{r}\}}{\exp\{-U(\theta_{k-1}) - \frac{1}{2} r_{k-1}^\top r_{k-1}\}},$$

  accept $\theta_k \leftarrow \tilde{\theta}$, $r_k \leftarrow -\tilde{r}$.

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

Lecture

Normalizing flows

I will follow Papamakarios et al. [2021a] and Kobyzev et al. [2021].

Definition and Basics

- Let x be a D-dimensional real vector, and suppose we would like to define a joint distribution over x.
- The main idea of flow-based modeling is to express x as a transformation T of a real vector u sampled from $p_u(u)$:

$$x = T(u), \quad u \sim p_u(u)$$

- We refer to $p_u(u)$ as the base distribution of the flow-based model
- The transformation T and the base distribution $p_u(u)$ can have parameters of their own (denote them as φ and ψ respectively); this induces a family of distributions over x parameterized by (φ, ψ).

Definition and Basics

- The defining property of flow-based models is that the transformation T must be invertible and both T and T^{-1} must be differentiable.
- Such transformations are known as diffeomorphisms and require that u be D-dimensional as well.
- Under these conditions, the density of x is well-defined and can be obtained by a change of variables

$$p_x(x) = p_u(u)|\det J_T(u)|^{-1}, \quad u = T^{-1}(x).$$

- Equivalently, we can also write $p_x(x)$ in terms of the Jacobian of T^1

$$p_x(x) = p_u(T^{-1}(x))|\det J_{T^{-1}}(x)|.$$

- The Jacobian $J_T(u)$ is the $D \times D$ matrix of all partial derivatives of T.

Definition and Basics

- An important property of invertible and differentiable transformations is that they are composable. Given two such transformations T_1 and T_2, their composition $T_2 \cdot T_1$ is also invertible and differentiable. Its inverse and Jacobian determinant are given by:

$$(T_2 \cdot T_1)^{-1} = T_2^{-1} \cdot T_1^{-1}, \quad \det J_{T_2 \cdot T_1}(u) = \det J_{T_2}(T_1(u)) \det J_{T_1}(u)$$

- In practice, it is common to chain together multiple transformations $T_1, \ldots, T_K$ to obtain $T = T_K \cdot \ldots \cdot T_1$, where each T_k transforms z_{k-1} into z_k, assuming $z_0 = u$ and $z_K = x$.

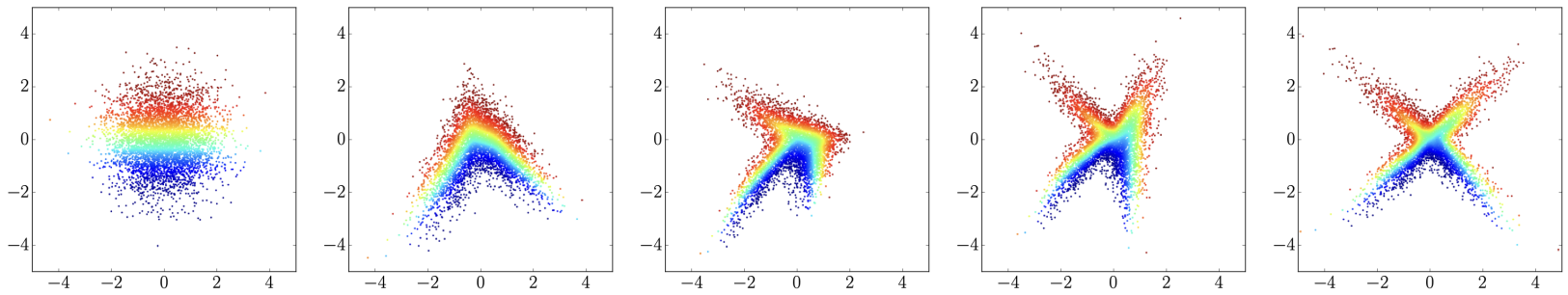


Figure: Example of a 4-step flow transforming samples from a standard-normal base density to a cross-shaped target density

Expressive Power of Flow-Based Models

- How expressive are flow-based models? Can they represent any distribution $p_x(x)$, even if the base distribution is restricted to be simple?
- We will show that for any pair of well-behaved distributions $p_x(x)$ (the target) and $p_u(u)$ (the base), there exists a diffeomorphism that can turn $p_u(u)$ into $p_x(x)$.

Expressive Power of Flow-Based Models

- Suppose that $p_x(x) > 0$ for all $x \in \mathbb{R}^D$, and assume that all conditional probabilities $\mathrm{P}(X_i \leq x_i | x_j, j < i)$ are differentiable with respect to $(x_i, x_j, j < i)$.
- Using the chain rule of probability

$$p_x(x) = \prod_{i=1}^{D} p_x(x_i | x_j, j < i)$$

- Since $p_x(x) > 0$ it follows that $p_x(x_i | x_j, j < i) > 0$ for all i and x.
- Define the transformation $F : x \to z \in (0,1)^D$ whose i-th element is given by the cumulative distribution function of the i-th conditional:

$$z_i = F_i(x_i, x_j, j < i) = \int_{-\infty}^{x_i} p_x(y_i | x_j, j < i) \mathrm{d}y_i = \mathrm{P}(X_i \leq x_i | x_j, j < i).$$

- Since F_i is differentiable w.r.t. its inputs, F is differentiable w.r.t. x. Moreover, each $F_i(\cdot, x_j, j < i) : \mathbb{R} \to (0,1)$ is invertible, since its derivative

$$\frac{\partial F_i}{\partial x_i} = p_x(x_i | x_j, j < i) > 0$$

Expressive Power of Flow-Based Models

- Because z_i doesn't depend on x_j for $i < j$, that implies we can invert F with its inverse F^{-1} given element-by-element as follows:

$$x_i = (F_i(\cdot, x_j, j < i))^{-1}(z_i), \quad i = 1, \ldots, D.$$

- The Jacobian of F is lower triangular since $\partial F_i / \partial x_j = 0$ for $i < j$. Hence, the Jacobian determinant of F is equal to the product of its diagonal elements:

$$\det J_F(x) = \prod_{i=1}^{D} \frac{\partial F_i}{\partial x_i} = \prod_{i=1}^{D} p_x(x_i | x_j, j < i) = p_x(x)$$

- Since $p_x(x) > 0$, the Jacobian determinant is non-zero everywhere. Therefore, the inverse of $J_F(x)$ exists, and is equal to the Jacobian of F^{-1}, so F is a diffeomorphism. Using a change of variables, we can calculate the density of z as follows:

$$p_z(z) = p_x(x) |\det J_F(x)|^{-1} = p_x(x) |p_x(x)|^{-1} = 1$$

- which implies z is distributed uniformly in the open unit cube $(0, 1)^D$.

Using Flows for Modeling and Inference

- Similarly to fitting any probabilistic model, fitting a flow-based model $p_x(x;\theta)$ to a target distribution $p_{x^\star}(x)$ can be done by minimizing some divergence or discrepancy between them.
- This minimization is performed with respect to the model's parameters $\theta = (\varphi, \psi)$, where φ are the parameters of T and ψ are the parameters of $p_u(u)$.
- We discuss a number of divergences for fitting flow-based models, with a particular focus on the Kullback–Leibler (KL) divergence as it is one of the most popular choices.

Forward KL Divergence and Maximum Likelihood Estimation

- The forward KL divergence between the target distribution $p_{x^\star}(x)$ and the flow-based model $p_x(x;\theta)$ can be written as follows:

$$\begin{aligned}\mathcal{L}(\theta) &= \mathrm{KL}(p_{x^\star}|p_x(\cdot;\theta)) = \mathbb{E}_{p_{x^\star}}[\log(p_{x^\star}/p_x(\cdot;\theta))] \\ &= -\mathbb{E}_{p_{x^\star}}[\log(p_x(\cdot;\theta))] + C \\ &= -\mathbb{E}_{p_{x^\star}}[\log(p_u(T^{-1}(\cdot,\varphi);\psi)) + \log\det J_{T^{-1}}(\cdot,\varphi)] + C\end{aligned}$$

- Assuming we have a set of samples $(x_n)_{n=1}^N$ from $p_{x^\star}(x)$, we can estimate the expectation over $p_{x^\star}(x)$ by Monte Carlo as follows:

$$\mathcal{L}(\theta) \approx -\frac{1}{N}\sum_{i=1}^{N}[\log(p_u(T^{-1}(x_i,\varphi);\psi)) + \log|\det J_{T^{-1}}(x_i,\varphi)|] + C$$

Forward KL Divergence and Maximum Likelihood Estimation

- In practice, we often optimize the parameters θ iteratively with stochastic gradient-based methods. We can obtain an unbiased estimate of the gradient of the KL divergence with respect to the parameters as follows:

$$\nabla_\varphi \mathcal{L}(\theta) \approx -\frac{1}{N}\sum_{i=1}^{N}[\nabla_\varphi \log(p_u(T^{-1}(x_i,\varphi);\psi)) + \nabla_\varphi \log|\det J_{T^{-1}}(x_i,\varphi)|],$$

$$\nabla_\psi \mathcal{L}(\theta) \approx -\frac{1}{N}\sum_{i=1}^{N}[\nabla_\psi \log(p_u(T^{-1}(x_i,\varphi);\psi)) + \nabla_\psi \log|\det J_{T^{-1}}(x_i,\varphi)|]$$

- The update with respect to may also be done in closed form if $p_u(u;\psi)$ admits closed-form maximum likelihood estimates, as is the case for example with Gaussian distributions.
- We can train a flow model with maximum likelihood even if we are not able to compute T or sample from $p_u(u;\psi)$. Yet these operations will be needed if we want to sample from the model after it is fitted.

Reverse KL divergence

- We may fit the flow-based model by minimizing the reverse KL divergence, which can be written as follows

$$\begin{aligned}\mathcal{L}(\theta) &= \mathrm{KL}(p_x(\cdot;\theta)|p_{x^\star}) = \mathbb{E}_{p_x(\cdot,\theta)}[\log(p_x(\cdot;\theta)/p_{x^\star})] \\ &= \mathbb{E}_{p_u(\cdot,\psi)}[\log(p_u(\cdot;\psi)) - \log|\det J_T(\cdot,\varphi)| - \log p^\star(T(\cdot,\varphi))]\end{aligned}$$

- Let $(u_n)_{n=1}^N$ be a set of samples from $p_u(u;\psi)$. The gradient of $\mathcal{L}(\theta)$ with respect to φ can be estimated as follows:

$$\nabla_\varphi \mathcal{L}(\theta) \approx -\frac{1}{N}\sum_{i=1}^{N}[\nabla_\varphi \log|\det J_T(u_i,\varphi)| + \nabla_\varphi \log p^\star(T(u_i,\varphi))]$$

- Similarly, we can estimate the gradient with respect to ψ by reparameterizing u as:

$$u = T'(u',\psi), \quad u' \sim p_{u'}(u')$$

  and then writing the expectation with respect to $p_{u'}(u')$
- We can fit a flow-based model by minimizing the reverse KL divergence even if we cannot evaluate the base density or compute the inverse transformation T^{-1}. However, we will need these operations if we would like to evaluate the density of the trained model.

Constructing Flows

- Normalizing flows are composable:

$$T = T_K \cdot \ldots T_1$$

- Use simple transformations as building blocks (each having a tractable inverse and Jacobian determinant) to define a complex transformation with more expressive power than any of its constituent components. Importantly, the flow's forward and inverse evaluation and Jacobian-determinant computation can be localized to the sub-flows. Assuming $z_0 = u$ and $z_K = x$, the forward evaluation is:

$$z_k = T_k(z_{k-1}),$$

  the inverse evaluation is:

$$z_{k-1} = T_k^{-1}(z_k)$$

- The Jacobian-determinant computation (in the log domain) is:

$$\log|\det J_T(z_0)| = \sum_{k=1}^{K} \log|\det J_{T_k}(z_k)|$$

  Increasing the 'depth' of the transformation crucially results in only $O(K)$ growth in the computational complexity

Constructing Flows

- In practice we implement either T_k or T_k^{-1} using a model (such as a neural network) with parameters φ_k, which we will denote as f_{φ_k}.
- In either case, we must ensure that the model is invertible and has a tractable Jacobian determinant. Ensuring that fk is invertible and explicitly calculating its inverse are not synonymous.
- In many implementations, even though the inverse of f_{φ_k} is guaranteed to exist, it can be expensive or even intractable to compute exactly.
- As discussed, the forward transformation T is used when sampling, and the inverse transformation T^{-1} is used when evaluating densities.
- If the inverse of f_{φ_k} is not efficient, either density evaluation or sampling will be inefficient or even intractable.

Autoregressive Flows

- We saw that, under mild conditions, we can transform any distribution $p_x(x)$ into a uniform distribution in $(0,1)^D$ using maps with a triangular Jacobian. Autoregressive flows are a direct implementation of this construction, specifying f_φ to have the following form

$$z_i' = \tau(z_i, h_i), h_i = c_i(z_j, j < i),$$

  where τ is termed the transformer and c_i the i-th conditioner.
- The transformer is a strictly monotonic function of z_i (and therefore invertible), is parameterized by h_i, and specifies how the flow acts on z_i in order to output z_i.
- The conditioner determines the parameters of the transformer, and in turn, can modify the transformer's behavior. The conditioner does not need to be a bijection. Its one constraint is that the i-th conditioner can take as input only the variables with dimension indices less that i.
- The parameters φ of f_φ are typically the parameters of the conditioner (not shown above for notational simplicity), but sometimes the transformer has its own parameters too (in addition to h_i).

Autoregressive Flows

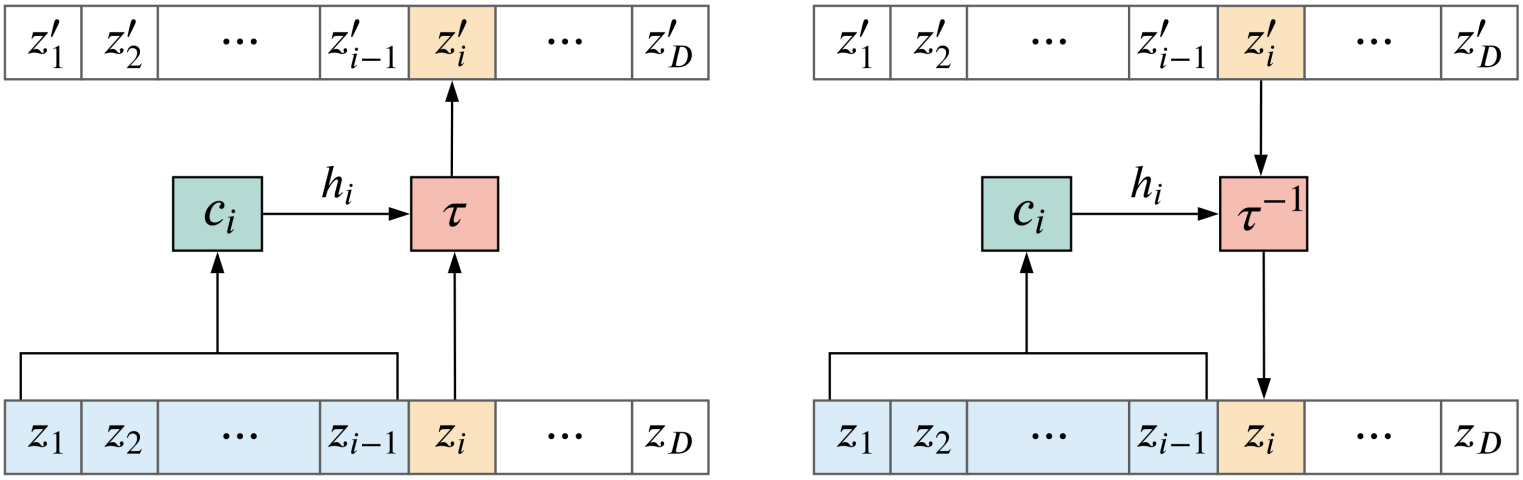
- It is easy to check that the above construction is invertible for any choice of τ and c_i as long as the transformer is invertible. Given z, we can compute z iteratively as follows:

$$z_i = \tau^{-1}(z_i', h_i), h_i = c_i(z_j, j < i).$$

- Jacobian of transformation is triangular and

$$\log|\det J_{f_\varphi}(z)| = \sum_{i=1}^{D} \log\left|\frac{\partial \tau}{\partial z_i}(z_i, h_i)\right|$$

Autoregressive Flows



(a) Forward (b) Inverse

Figure: Illustration of the i-th step of an autoregressive flow.

Affine transformers

- One of the simplest possible choices for the transformer is the class of affine functions:

$$\tau(z_i, h_i) = \alpha_i z_i + \beta_i, \quad h_i = (\alpha_i, \beta_i)$$

- The above can be thought of as a location-scale transformation, where α_i controls the scale and β_i controls the location. Invertibility is guaranteed if $\alpha_i \neq 0$.
- Log absolute Jacobian determinant is

$$\log |\det J_{f_\varphi}(z)| = \sum_{i=1}^{D} \log |\alpha_i|.$$

- Autoregressive flows with affine transformers are attractive because of their simplicity and analytical tractability, but their expressivity is limited.
- Affine transformers are popular in the literature, having been used in models such as NICE, RealNVP etc.

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.

Tong Che, Ruixiang ZHANG, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12275–12287. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/90525e70b7842930586545c6f1c9310c-Paper.pdf.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

R. Douc, E. Moulines, P. Priouret, and P. Soulier. *Markov chains*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, 2018. ISBN 978-3-319-97703-4; 978-3-319-97704-1. doi: 10.1007/978-3-319-97704-1. URL https://doi.org/10.1007/978-3-319-97704-1.

Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *arXiv preprint arXiv:2105.12603*, 2021.

Matthew D Hoffman, Andrew Gelman, et al. The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, nov 2021. doi: 10.1109/tpami.2020.2992934. URL https://doi.org/10.1109%2Ftpami.2020.2992934.

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

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn.*