Path Integral Sampler: A Stochastic Control Approach for Sampling

Qinsheng Zhang   Yongxin Chen

Georgia Institute of Technology

International Conference on Learning Representations

Sampling

Problem Statement

Draw samples from a given target density $\hat{\mu} = Z\mu$

- $\hat{\mu}$ known up to a normalizing constant Z

Stochastic control

Stochastic system

$$d\mathbf{x}_t = \mathbf{u}_t dt + d\mathbf{w}_t, \;\; \mathbf{x}_0 \sim \nu,$$

Notations:

- $\mathcal{Q}^{\mathbf{u}}$: path measure associated with control policy $\mathbf{u}$
- $\mu^{\mathbf{u}}$: terminal distribution of $\mathbf{x}_T$ driven by $\mathbf{u}$
- $\mathcal{Q}^0$: path measure associated with $\mathbf{u} = 0$
- μ^0: terminal distribution of $\mathbf{x}_T$ driven by $\mathbf{u} = 0$

Stochastic control

Stochastic system

$$d\mathbf{x}_t = \mathbf{u}_t dt + d\mathbf{w}_t, \ \mathbf{x}_0 \sim \nu,$$

Notations:

- $\mathcal{Q}^{\mathbf{u}}$: path measure associated with control policy $\mathbf{u}$
- $\mu^{\mathbf{u}}$: terminal distribution of $\mathbf{x}_T$ driven by $\mathbf{u}$
- $\mathcal{Q}^0$: path measure associated with $\mathbf{u} = 0$
- μ^0: terminal distribution of $\mathbf{x}_T$ driven by $\mathbf{u} = 0$

Control cost

$$\mathbb{E}\left[\int_0^T \frac{1}{2} \|\mathbf{u}_t\|^2 dt + \Psi(\mathbf{x}_T) \mid \mathbf{x}_0 \sim \nu\right]$$

Sampling as a stochastic control problem

Goal

Find a control policy to drive particles from ν to μ

$$\mathrm{d}\mathbf{x}_t = \mathbf{u}_t \mathrm{d}t + \mathrm{d}\mathbf{w}_t, \ \ \mathbf{x}_0 \sim \nu$$

Warning

- $\{\mathbf{u} : \mu^{\mathbf{u}} = \mu\}$ can be infinite

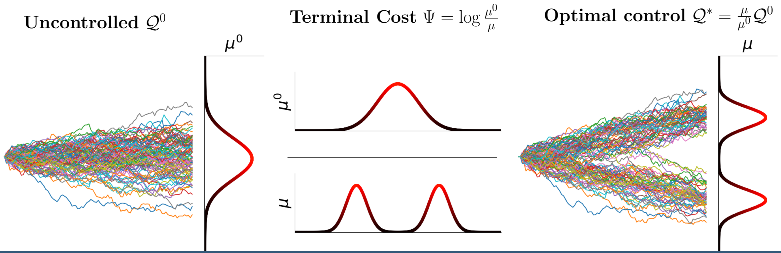
Sampling as a stochastic control problem

Theorem 1

When ν is a Dirac distribution and terminal loss is chosen as $\Psi(\mathbf{x}_T) = \log \frac{\mu^0(\mathbf{x}_T)}{\mu(\mathbf{x}_T)}$, the distribution $\mathcal{Q}^*$ induced by the optimal control policy is

$$\mathcal{Q}^*(\tau) = \mathcal{Q}^0(\tau|\mathbf{x}_T)\mu(\mathbf{x}_T).$$

Moreover, $\mathcal{Q}^*(\mathbf{x}_T) = \mu(\mathbf{x}_T)$.



Path Integral Sampler (PIS)

Optimal control via Path Integral

$$\mathbf{u}_t^*(\mathbf{x}) = \nabla \log \phi_t(\mathbf{x}) \quad \phi_t(\mathbf{x}) = \mathbb{E}_{\mathcal{Q}^0}[\exp(-\Psi(\mathbf{x}_T))|\mathbf{x}_t = \mathbf{x}]$$

Path Integral Sampler (PIS)

Optimal control via Path Integral

$$\mathbf{u}_t^*(\mathbf{x}) = \nabla \log \phi_t(\mathbf{x}) \quad \phi_t(\mathbf{x}) = \mathbb{E}_{\mathcal{Q}^0}[\exp(-\Psi(\mathbf{x}_T))|\mathbf{x}_t = \mathbf{x}]$$

Wasserstein-2 distance

Under mild condition, with sampling step size Δt, if $\left\|\mathbf{u}_t^* - \mathbf{u}_t\right\|^2 \leq d\epsilon$ for any t, then

$$W_2(\mathcal{Q}^u(\mathbf{x}_T), \mu(\mathbf{x}_T)) = \mathcal{O}(\sqrt{Td(\Delta t + \epsilon)}).$$

Path Integral Sampler (PIS)

Optimal control via Path Integral

$$\mathbf{u}_t^*(\mathbf{x}) = \nabla \log \phi_t(\mathbf{x}) \quad \phi_t(\mathbf{x}) = \mathbb{E}_{\mathcal{Q}^0}[\exp(-\Psi(\mathbf{x}_T))|\mathbf{x}_t = \mathbf{x}]$$

Wasserstein-2 distance

Under mild condition, with sampling step size Δt, if $\left\|\mathbf{u}_t^* - \mathbf{u}_t\right\|^2 \leq d\epsilon$ for any t, then

$$W_2(\mathcal{Q}^u(\mathbf{x}_T), \mu(\mathbf{x}_T)) = \mathcal{O}(\sqrt{Td(\Delta t + \epsilon)}).$$

- Accessing ϕ_t can be expensive unless Ψ is simple

Path Integral Sampler (PIS)

Neural control policy $\mathbf{u}_\theta$:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \mathbb{E}_{\mathcal{Q}^u}\left[\int_0^T \frac{1}{2}\|\mathbf{u}_t\|^2 \, dt + \log\frac{\mu^0(\mathbf{x}_T)}{\mu(\mathbf{x}_T)}\right].$$

Algorithm 1 Training

Input: Vector: $\mathbf{x}_0 = 0$, Scalar: $y_0 = 0$
Output: $\mathbf{u}_t(\mathbf{x})$ parameterized by θ
Define: SDE drift $\mathbf{f}(t, [\mathbf{x}_t, y_t]) = [\mathbf{u}_{\theta t}(\mathbf{x}_t), \frac{1}{2}\|\mathbf{u}_{\theta t}(\mathbf{x}_t)\|^2]$, diffusion $\mathbf{g}(t, [\mathbf{x}_t, y_t]) = [1, 0]$
loop epoches
 $\mathbf{x}_T, y_T = \text{sdeint}(\mathbf{f}, \mathbf{g}, [\mathbf{x}_0, y_0], [0, T])$ # Integrate SDE from 0 to T with Neural SDE
 Gradient descent step $\nabla_\theta[y_T + \log\frac{\mu^0(\mathbf{x}_T)}{\mu(\mathbf{x}_T)}]$ # Optimize control policy
done

Path Integral Sampler (PIS)

Neural control policy $\mathbf{u}_\theta$:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \mathbb{E}_{\mathcal{Q}^u}\left[\int_0^T \frac{1}{2}\|\mathbf{u}_t\|^2 \, \mathrm{d}t + \log\frac{\mu^0(\mathbf{x}_T)}{\mu(\mathbf{x}_T)}\right].$$

Algorithm 1 Training

Input: Vector: $\mathbf{x}_0 = 0$, Scalar: $y_0 = 0$
Output: $\mathbf{u}_t(\mathbf{x})$ parameterized by θ
Define: SDE drift $\mathbf{f}(t, [\mathbf{x}_t, y_t]) = [\mathbf{u}_{\theta t}(\mathbf{x}_t), \frac{1}{2}\|\mathbf{u}_{\theta t}(\mathbf{x}_t)\|^2]$, diffusion $\mathbf{g}(t, [\mathbf{x}_t, y_t]) = [1, 0]$
loop epoches
  $\mathbf{x}_T, y_T = \text{sdeint}(\mathbf{f}, \mathbf{g}, [\mathbf{x}_0, y_0], [0, T])$ # Integrate SDE from 0 to T with Neural SDE
  Gradient descent step $\nabla_\theta[y_T + \log\frac{\mu^0(\mathbf{x}_T)}{\mu(\mathbf{x}_T)}]$ # Optimize control policy
done

Warning:

- Global optimal policy is impossible to train

Path Integral Sampler (PIS)

How to get unbiased samples?

Importance sampling for suboptimal policy

$$w^u(\tau) = \frac{d\mathcal{Q}^*(\tau)}{d\mathcal{Q}^u(\tau)} = \exp\Big(\int_0^T -\frac{1}{2}\|\mathbf{u}_t\|^2 dt - \mathbf{u}_t' d\mathbf{w}_t - \Psi(\mathbf{x}_T)\Big)$$

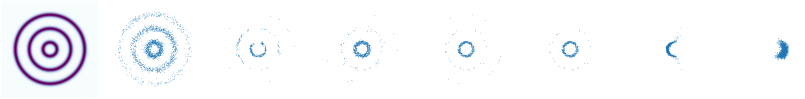
Benchmark

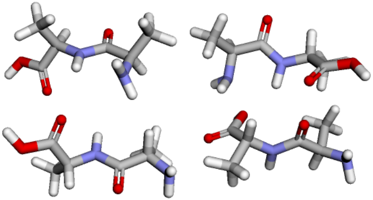
Normalization constant estimation

	MG ($d = 2$)			Funnel ($d = 10$)			LGCP ($d = 1600$)		
	B	S	A	B	S	A	B	S	A
PIS_{RW}-GT	-0.012	0.013	0.018	-	-	-	-	-	-
PIS-NN	-1.691	0.370	1.731	-0.098	**5e-3**	0.098	-92.4	6.4	92.62
PIS-Grad	-0.440	0.024	0.441	-0.103	9e-3	0.104	-13.2	3.21	13.58
PIS_{RW}-NN	-1.192	0.482	1.285	-0.018	7e-3	0.02	-60.8	4.81	60.99
PIS_{RW}-Grad	**-0.021**	**0.030**	**0.037**	**-0.008**	9e-3	**0.012**	**-1.94**	**0.91**	**2.14**
AFT	-0.509	0.24	0.562	-0.208	0.193	0.284	-3.08	1.59	3.46
SMC	-0.362	0.293	0.466	-0.216	0.157	0.267	-435	14.7	436
NUTS	-1.871	0.527	1.943	-0.835	0.257	0.874	-1.3e3	8.01	1.3e3
HMC	-1.876	0.527	1.948	-0.835	0.257	0.874	-1.3e3	8.01	1.3e3
VI-NF	-1.632	0.965	1.896	-0.236	0.0591	0.243	-77.9	5.6	78.2

Table 1: Benchmarking on mode separated mixture of Gaussian (MG), Funnel distribution and Log Gaussian Cox Process (LGCP) for estimation log normalization constants. B and S stand for estimation bias and standard deviation among 100 runs and $A^2 = B^2 + S^2$.

More experiments





Alanine dipeptide molecules

	B	S	$\sqrt{B^2 + S^2}$
VI-NF	-2.3	0.76	2.42
AFT	-1.7	0.95	1.96
SMC	-10.6	2.01	10.79
PIS_{RW}-NN	-1.9	0.81	2.06
PIS_{RW}-Grad	**-0.87**	**0.31**	**0.92**

Estimation of $\log p_\theta(x)$ of a trained VAE.

- Paper: `https://arxiv.org/abs/2111.15141`
- Code: `https://github.com/qsh-zh/pis`