Sampling via Controlled Stochastic Dynamical Systems

Benjamin J. Zhang[1], **Tuhin Sahai**[2], **and Youssef M. Marzouk**[1]

[1]*Center for Computational Science and Engineering, Massachusetts Institute of Technology*

[2]*Raytheon Technologies Research Center*

Abstract

We present a framework for constructing controlled stochastic differential equations (SDEs) that exactly sample from a class of probability distributions with Gaussian tails. By choosing the reference process to be a linear SDE, we can find the optimal control that guides the system to a target distribution by only solving a static optimization problem. In practice, the method lacks robustness due to the high sensitivity to the algorithm's parameters.

Motivation

- Computing expectations with respect to complex probability distributions is ubiquitous in statistics and ML
- Efficient sampling methods for distributions yield estimators for approximating expectations
- Finding controlled SDEs enables exact sampling of target distributions

Problem setting

- **Given an unnormalized target density** $\pi(x)$, **design reference SDE**

$$\mathrm{d}X_t = \mathbf{A}(X_t)\,\mathrm{d}t + \mathbf{B}(X_t)\,\mathrm{d}W_t, \quad X_0 = x_0 \tag{1}$$

  and **design optimal feedback control** $u(t,x)$ such that the controlled diffusion process

$$\mathrm{d}Y_t = [\boldsymbol{A}(Y_t) + \boldsymbol{B}u(t,Y_t)]\,\mathrm{d}t + \boldsymbol{B}(Y_t)\,\mathrm{d}W_t, \quad Y_0 = x_0 \tag{2}$$

  has its time T marginal equal to the target distribution, $Y_T \sim \pi$
- Independent simulations of Y_t produce samples of π

Background on controlled SDEs

Markov generator

$$\mathcal{A}\psi = \langle \boldsymbol{A}(x), \nabla\psi\rangle + \frac{1}{2}\mathrm{Tr}\left[\boldsymbol{B}(x)\boldsymbol{B}(x)^*\nabla^2\psi\right] \tag{3}$$

- *Linear* operator on $\mathcal{C}^2(\mathbb{R}^d)$ that describes evolution of statistics of SDE

Kolmogorov backward equation

Let $f \in \mathcal{C}^2(\mathbb{R}^d)$ be strictly positive over $\mathbb{R}^d$ and $\Phi(t,x) = \mathbb{E}[f(X_T)|X_t = x]$ be the solution to the *Kolmogorov backward equation*:

$$\begin{cases} \partial_t\Phi + \mathcal{A}\Phi = 0 \\ \Phi(T,x) = f(x). \end{cases}$$

Doob h-transform

If $u(t,x) = \mathbf{B}(x)^*\nabla\log\Phi(t,x)$ is the controller, then the density of Y_T is

$$\eta_T^u(x) = \frac{f(x)\eta_T(x)}{\Phi(0,x)}$$

where $\eta_T(x)$ is the reference distribution.

- If $f(x) = \pi(x)/\eta_T(x)$, then $\eta_T^u(x) = \pi(x)$ exactly!

Construction of the controlled SDEs sampler

Choosing a reference process

Desiderata

- Ability to compute time T marginal exactly
- Access solutions to KBE cheaply

Choose linear SDEs of the form:

$$\begin{cases} \mathrm{d}X_t &= -X_t\mathrm{d}t + \boldsymbol{B}\mathrm{d}W_t \\ X_0 &= x_0. \end{cases} \tag{4}$$

Features

- Exact expression of terminal $X_T \sim \mathcal{N}(x_0e^{-t}, \boldsymbol{\Sigma}_t)$, $\boldsymbol{\Sigma}_t = \frac{1}{2}(1-e^{-2t})\mathbf{B}\mathbf{B}^*$
- Eigenfunctions of Markov generator **can be found exactly** and are of the form

$$\phi_{\boldsymbol{n}}(x) = \prod_{i=1}^d \mathrm{He}_{n_i}\left(\frac{\sqrt{2}}{\mu_i}\langle x, e_i\rangle\right) \tag{5}$$

  with eigenvalues $\lambda_{\boldsymbol{n}} = -\sum_{i=1}^d n_i$, $\mathbf{B}^*e_i = \mu_ie_i$.
- Cheap solutions to the KBE:

$$f(x) = \sum_{\boldsymbol{n}} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}(x), \quad \text{then } \Phi(t,x) = \sum_{\boldsymbol{n}} c_{\boldsymbol{n}} e^{\lambda_{\boldsymbol{n}}(T-t)}\phi_{\boldsymbol{n}}(x). \tag{6}$$

Projecting onto eigenfunctions

Project $f(x) = \pi(x)/\eta_T(x)$ **onto eigenfunctions**

- Define $\tilde{f}(x,c) = \sum_{\boldsymbol{n}\in\mathcal{I}} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}(x)$, where $\mathcal{I} \subset \mathbb{N}_0^d$ is some set of multi-indices.
- Write the exact and approximate densities as

$$\pi_0(x) = \frac{f(x)\eta_T(x)}{\gamma}, \quad \tilde{\pi}(x) = \frac{\tilde{f}(x,c)\eta_T(x)}{\tilde{\gamma}}, \tag{7}$$

  γ (unknown) and $\tilde{\gamma} = \sum_{\boldsymbol{n}\in\mathcal{I}} c_{\boldsymbol{n}} e^{\lambda_{\boldsymbol{n}}T}\phi_{\boldsymbol{n}}(x_0)$ are the normalizing constants
- **Minimize KL divergence** from $\tilde{\pi}$ to π_0: $\min_{c\in\mathbb{R}^{|\mathcal{I}|}} \mathbb{E}_{\pi_0}\left[\log\pi_0(x) - \log\tilde{\pi}(x)\right]$.
- Resulting optimization problem:

$$\max_{c\in\mathbb{R}^{|\mathcal{I}|}} \mathbb{E}_{\eta_T}\left[f(x)\log\frac{\tilde{f}(x,c)}{\tilde{\gamma}(c)}\right]. \tag{8}$$

- Objective function evaluated by sample average approximation

Choosing the terminal marginal η_T and initial condition x_0

- Choosing terminal marginal is an open question
- In numerical examples, η_T found *ad hoc*
- Initial condition is always deterministic and based on terminal marginal

Rules of thumb

- Choose η_T such that objective function estimated with low variance
- Heuristic: find a normal approximation of π via Laplace approximation, expectation propagation or mean-field variational Bayes.
- Center π so that η_T of the form $\mathcal{N}(0,\boldsymbol{\Sigma})$

Algorithm summary

Input: Unnormalized target density $\pi(x)$, set of multi-indices $\mathcal{I} \subset \mathbb{N}_0^d$

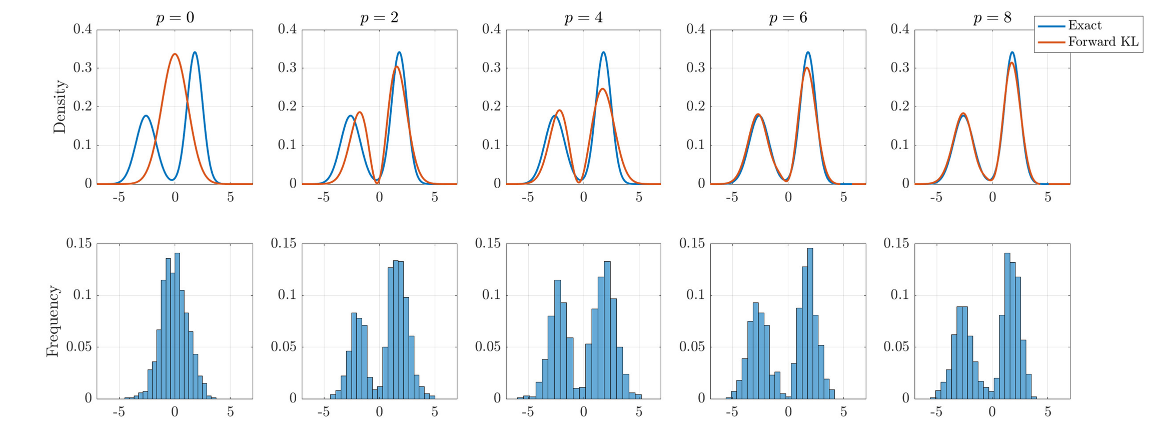
Output: Optimal control $u(t,x)$

1. Find an approximation $\eta(x) = \mathcal{N}(0,\boldsymbol{\Sigma})$ to $\pi(x)$, define $f(x) = \pi(x)/\eta(x)$
2. Compute $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^*$
3. Set $\boldsymbol{B} = \sqrt{\frac{2\boldsymbol{\Lambda}}{1-e^{-2T}}}\boldsymbol{V}$
4. Construct eigenfunctions $\{\phi_{\boldsymbol{n}}(x)\}_{\boldsymbol{n}\in\mathcal{I}}$
5. Draw M independent $X^{(i)} \sim \mathcal{N}(0,\boldsymbol{\Sigma})$
6. Solve $c^* = \arg\max_{c\in\mathbb{R}^{|\mathcal{I}|}} \frac{1}{M}\sum_{i=1}^M f(X^{(i)})\log\frac{\tilde{f}(X^{(i)},c)}{\tilde{\gamma}(c)}$ where $\tilde{f}(X^{(i)},c) = \sum_{\boldsymbol{n}\in\mathcal{I}} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}(X^{(i)})$, and $\gamma(c) = \sum_{\boldsymbol{n}\in\mathcal{I}} c_{\boldsymbol{n}}e^{\lambda_{\boldsymbol{n}}(T-t)}\phi_{\boldsymbol{n}}(x_0)$
7. Doob h-transform is $u(t,x) = \boldsymbol{B}^*\nabla\log\sum_{\boldsymbol{n}\in\mathcal{I}} c^*_{\boldsymbol{n}}e^{\lambda_{\boldsymbol{n}}(T-t)}\phi_{\boldsymbol{n}}(x)$.

Numerical examples when the method works

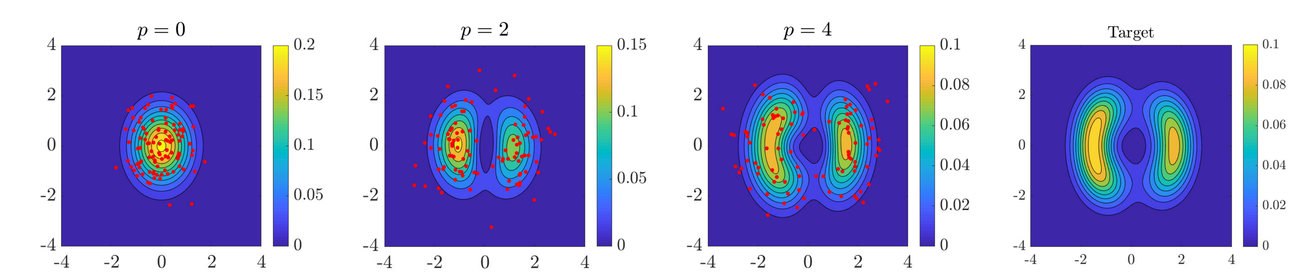
1D example

- $\pi(x) = 0.6\mathcal{N}(x; 1.8, 0.7^2) + 0.4\mathcal{N}(x; -2.6, 0.9^2)$
- $\eta_T = \mathcal{N}(x; 0, 1.4^2)$



2D example

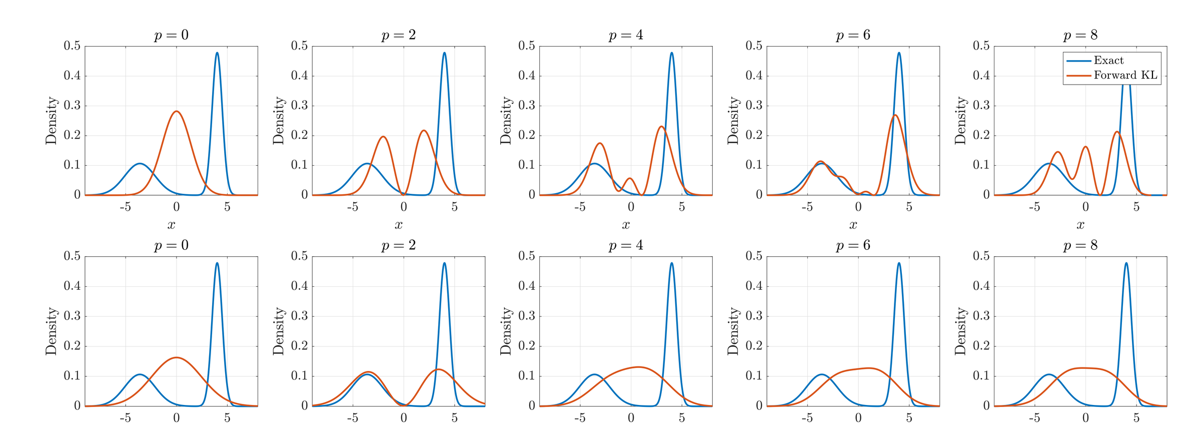
- $-\log\pi(x) = 0.5\left(\frac{\|x\|_2 - 1.5}{0.7}\right)^2 - \log\left[\exp\left(-\left(\frac{x_1-2}{0.8\sqrt{2}}\right)^2\right) + \exp\left(-\left(\frac{x_1+1.5}{0.8\sqrt{2}}\right)^2\right)\right]$
- η_T is normal with $\boldsymbol{\Sigma} = \mathrm{diag}(0.6, 1)$



Numerical examples when the method fails

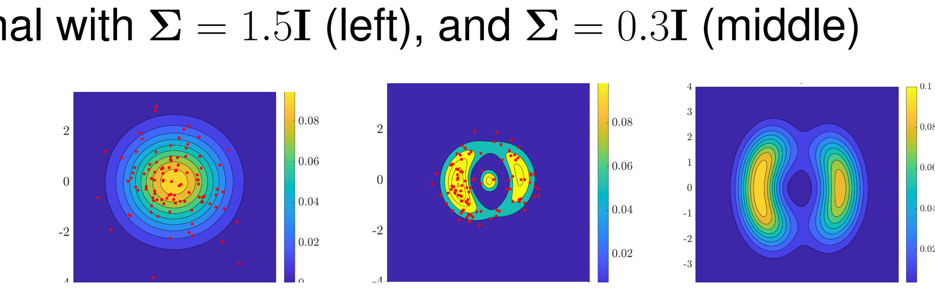
1D example

- $\pi(x) = 0.6\mathcal{N}(x; 4, 0.5^2) + 0.4\mathcal{N}(x; -3.6, 1.5^2)$
- $\eta_T(x) = \mathcal{N}(x; 0, 4)$ (top row) and $\eta_T(x) = \mathcal{N}(x; 0, 36)$ (bottom row)



2D example: $\pi(x)$ same as in the above 2D example

- η_T is normal with $\boldsymbol{\Sigma} = 1.5\mathbf{I}$ (left), and $\boldsymbol{\Sigma} = 0.3\mathbf{I}$ (middle)



Why is it not better?

Fundamental issues

- Method is highly sensitive to parameters – need to be carefully tuned
- When marginal η_T is chosen poorly, then approximating class cannot capture target properly in practice
- These issues are independent of the nature of the resulting optimization problem or how well the objective can be evaluated

Other issues hindering implementation

- Robust evaluation of the objective functions
- Scaling the approach to higher dimensions— requires judiciously choosing basis functions