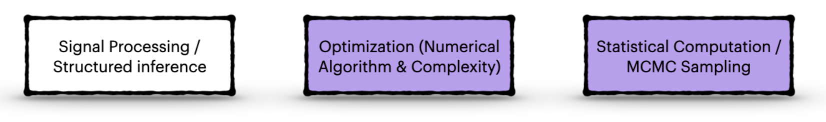
Dynamics in Algorithm Design: Optimization, Sampling and Diffusion

Qijia Jiang

March 2024

Berkeley Lab

Research Overview and Roadmap



Today:

(1) information-theoretic complexity in optimization

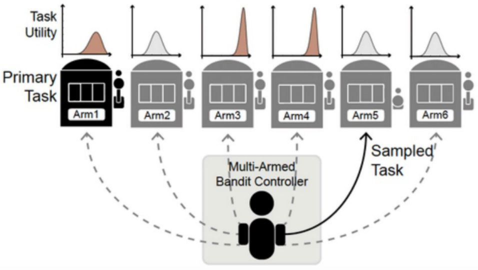
(2) ∞-dim optimization in general metric space $\rightsquigarrow$ sampling and diffusion!

Optimization

Optimization comes in many different flavors:

- online

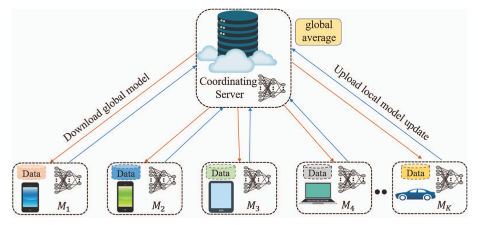
Decision making under uncertainty



Optimization comes in many different flavors:

- online
- distributed

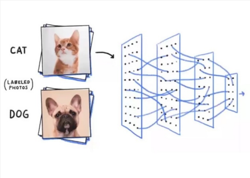
Federated Learning

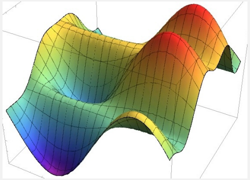


Optimization comes in many different flavors:

- online
- distributed
- non-convex

Neural Network Training



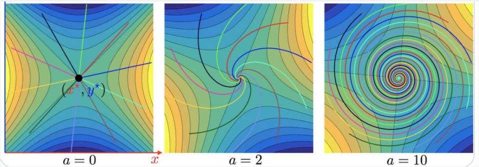


Optimization comes in many different flavors:

- online
- distributed
- non-convex
- min-max

Game Dynamics and Equilibrium

$$\min_x \max_y f(x,y) = axy + x^2 - y^2, \; a = \text{interaction}$$



Optimization comes in many different flavors:

- online
- distributed
- non-convex
- min-max
- combinatorial

Optimization and Data Science

Optimization comes in many different flavors:

- online
- distributed
- non-convex
- min-max
- combinatorial
- robust

Optimization and Data Science

Optimization comes in many different flavors:

- online
- distributed
- non-convex
- min-max
- combinatorial
- robust
- stochastic

Optimization and Data Science

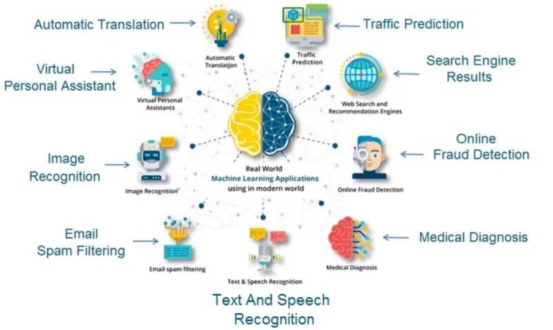
Optimization comes in many different flavors:

- online
- distributed
- non-convex
- min-max
- combinatorial
- robust
- stochastic
- non-Euclidean

Optimization and Data Science

Optimization comes in many different flavors:

- online
- distributed
- non-convex
- min-max
- combinatorial
- robust
- stochastic
- non-Euclidean
- ...



and is integral to machine learning success stories

Outline

Goal:

> Study # rounds of interaction (k) with an oracle $\mathcal{O}$, such that for functions f in certain convex function class $\mathcal{F}$,
> $$f(x_k) - f^* \leq \epsilon$$
> for the output x_k.

Two stops:

- Function with smooth higher-order derivatives ($\mathcal{F}$) and higher-order oracle ($\mathcal{O}$) [BJLLS COLT '19]
- Non-smooth function ($\mathcal{F}$) with parallel gradient oracle ($\mathcal{O}$) [BJLLS NeurIPS '19]

Setting

- Function class $\mathcal{F}$: Lipschitz gradient, i.e., $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$
- Gradient Oracle $\mathcal{O}$: access to $\{f(\cdot), \nabla f(\cdot)\}$ at any query point x
- Ex: linear system $f(x) = \|Ax - b\|_2^2$

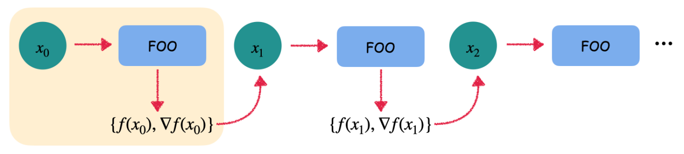


Figure 1: Classical First-Order Oracle Model

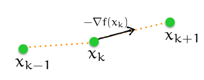
Curved arrow is where algorithm design comes in (Ex: $x_1 \leftarrow x_0 - h \cdot \nabla f(x_0)$)

Gradient Descent and Accelerated Gradient Descent

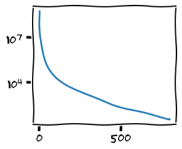
Gradient Descent

$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

one gradient call per iteration



Rate $\mathcal{O}(1/k)$, dimension-free

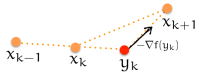


ODE: $\dot{X}_t = -\nabla f(X_t)$

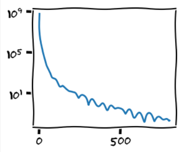
Accelerated Gradient Descent

$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$

$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$



Rate $\mathcal{O}(1/k^2)$, not a descent method



ODE: $\ddot{X}_t + 3/t \cdot \dot{X}_t + \nabla f(X_t) = 0$

Formalism of oracle model led to the discovery of AGD and it is the best one can do. [Nemirovski & Yudin ’83]

Generalization: Higher Order Oracle Model

Setting

- Function class $\mathcal{F}$: p-times differentiable & p-th order smooth

$$\|\nabla^p f(x) - \nabla^p f(y)\| := \max_{\|v\|=1} \left|\nabla^p f(x)[v]^p - \nabla^p f(y)[v]^p\right| \leq L_p\|x-y\|$$

- p-th Order Oracle $\mathcal{O}$: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Example: ℓ_p-regression $\frac{1}{p}\|Ax-b\|_p^p$

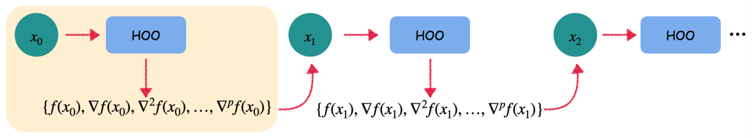


Figure 2: Higher order oracle

Generalization: Higher Order Oracle Model

Setting

- Function class $\mathcal{F}$: p-times differentiable & p-th order smooth
- p-th Order Oracle $\mathcal{O}$: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Example: ℓ_p-regression $\frac{1}{p}\|Ax - b\|_p^p$

Prior Art [Agarwal & Hazan '18, Nesterov '18]

Under mild assumption on the algorithm, one has

$$\min_{0 \le t \le k} f(x_t) - f^* \ge \Omega\left(\frac{L_p}{k^{\frac{3p+1}{2}}}\right).$$

There is a family of algorithm that achieve convergence rate $\mathcal{O}(\frac{1}{k^{p+1}})$.

Generalization: Higher Order Oracle Model

Setting

- Function class $\mathcal{F}$: p-times differentiable & p-th order smooth
- p-th Order Oracle $\mathcal{O}$: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Example: ℓ_p-regression $\frac{1}{p}\|Ax - b\|_p^p$

Prior Art [Agarwal & Hazan '18, Nesterov '18]

Under mild assumption on the algorithm, one has

$$\min_{0 \le t \le k} f(x_t) - f^* \ge \Omega\left(\frac{L_p}{k^{\frac{3p+1}{2}}}\right).$$

There is a family of algorithm that achieve convergence rate $\mathcal{O}(\frac{1}{k^{p+1}})$.

Coincide when $p = 1$.

For $p = 2$: Accelerated Cubic-Regularized Newton [Nesterov & Polyak '08].

Generalization: Higher Order Oracle Model

Setting

- Function class $\mathcal{F}$: p-times differentiable & p-th order smooth
- p-th Order Oracle $\mathcal{O}$: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Example: ℓ_p-regression $\frac{1}{p}\|Ax - b\|_p^p$

Prior Art [Agarwal & Hazan '18, Nesterov '18]

Under mild assumption on the algorithm, one has

$$\min_{0\le t\le k} f(x_t) - f^* \ge \Omega\left(\frac{L_p}{k^{\frac{3p+1}{2}}}\right).$$

There is a family of algorithm that achieve convergence rate $\mathcal{O}(\frac{1}{k^{p+1}})$.

Coincide when $p = 1$.

For $p = 2$: Accelerated Cubic-Regularized Newton [Nesterov & Polyak '08].

Gap between upper & lower bound? Better algorithm?

The Iteration-Complexity Optimal Algorithm

Convergence Guarantee [BJLLS, COLT'19]

There is an algorithm with error decrease as $\tilde{\mathcal{O}}(k^{-\frac{3p+1}{2}})$.

Still leverage interpolation of past iterates, **but** each iteration of the algorithm requires solving a tensor minimization problem:

$$y_{k+1} = \operatorname*{arg\,min}_{y} \left\{ f_p(y; x_k) + \frac{L_p}{p!} \|y - x_k\|^{p+1} \right\}$$

When $p = 1, 2, 3$ efficiently solvable.

The Iteration-Complexity Optimal Algorithm

Convergence Guarantee [BJLLS, COLT'19]

There is an algorithm with error decrease as $\tilde{\mathcal{O}}(k^{-\frac{3p+1}{2}})$.

Still leverage interpolation of past iterates, **but** each iteration of the algorithm requires solving a tensor minimization problem:

$$y_{k+1} = \arg\min_{y} \left\{ f_p(y; x_k) + \frac{L_p}{p!} \|y - x_k\|^{p+1} \right\}$$

When $p = 1, 2, 3$ efficiently solvable.

These are quite powerful oracles ...

Broadly useful beyond scientific curiosity?

Black-box Oracle Complexity: Non-Smooth Function

Setting

- Function class $\mathcal{F}$: Lipschitz, i.e., $|f(x) - f(y)| \leq L_0 \cdot \|x - y\|$
- First Order Oracle $\mathcal{O}$: return $\{f(x), \partial f(x)\}$
- Example: ℓ_1-penalty $\|\cdot\|_1$, hinge loss

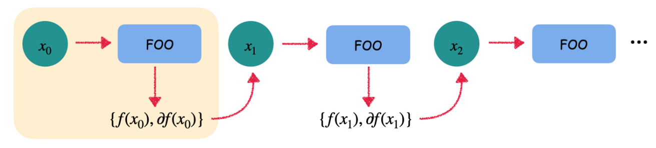
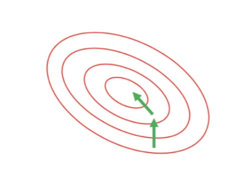


Figure 2: Classical Sequential Setup (non-smooth f)

Black-box Oracle Complexity: Non-Smooth Function



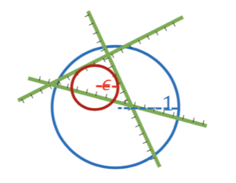
(Sub)gradient Descent

At iteration k,

$x_{k+1} \leftarrow x_k - h \cdot \nabla f(x_k)$

Output: $\bar{x}_K = \frac{1}{K}\sum x_k$, rate $\mathcal{O}(1/\sqrt{K})$

$$\mathcal{O}\left(\frac{1}{\epsilon^2}\right) \text{ queries suffice}$$



Cutting Plane Methods

High-dimensional binary search

$\rightsquigarrow$ separation oracle implementable by ∇f thanks to convexity

$$\mathcal{O}\left(d \log\left(\frac{1}{\epsilon}\right)\right) \text{ queries suffice}$$

Sequential Setup ($K = 1$):

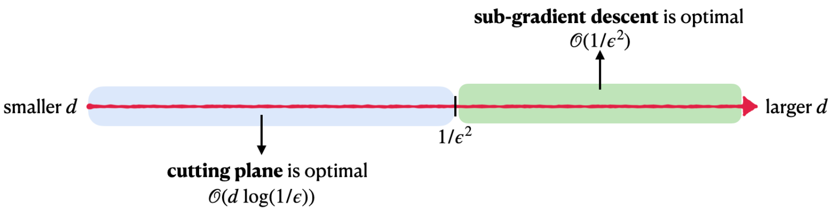


Figure 2: Upper & Lower Bound for non-smooth f

Allowed to submit K gradient queries in parallel [Nemirovski '94].

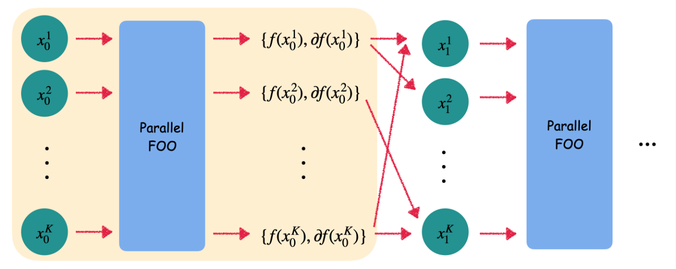


Figure 3: Schematic for Parallel Setup

Call **Depth** the # queries to parallel oracle $\mathcal{O}$

Generalization: Parallel Oracle

Allowed to submit K gradient queries in parallel [Nemirovski '94].

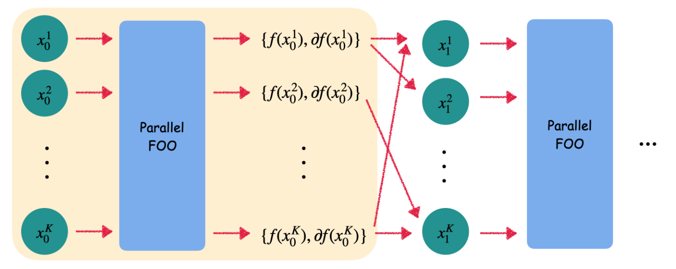
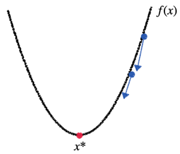


Figure 3: Schematic for Parallel Setup



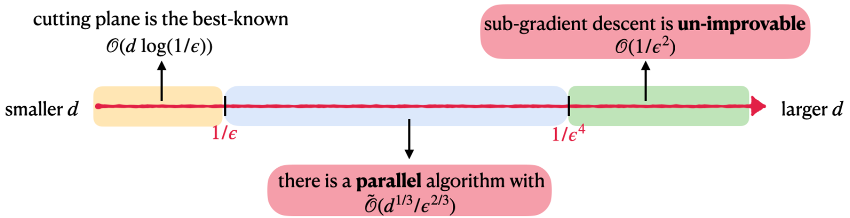
Call **Depth** the # queries to parallel oracle $\mathcal{O}$

For $K = \text{poly}(d)$, best possible depth?

Power of non-adaptive information in convex optimization?

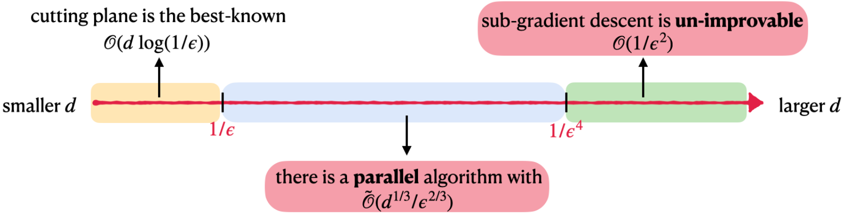
Our Result

Upper & Lower Bound on Parallel Complexity [BJLLS, NeurIPS '19]



Our Result

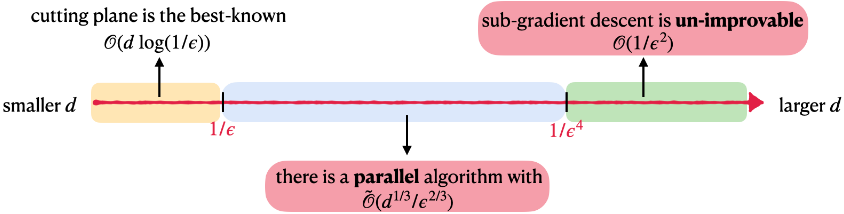
Upper & Lower Bound on Parallel Complexity [BJLLS, NeurIPS '19]



Randomized smoothing of non-smooth f as $g = f * \gamma_r$, **parallel** computation of gradient by sampling $x_i \sim \mathcal{N}(y, r \cdot I)$ and $\hat{\nabla} g(y) = \frac{1}{m}\sum_{i=1}^{m} \nabla f(x_i) \rightsquigarrow$ leverage highly smooth acceleration result on the smoothed $g(\cdot)$

Our Result

Upper & Lower Bound on Parallel Complexity [BJLLS, NeurIPS ’19]

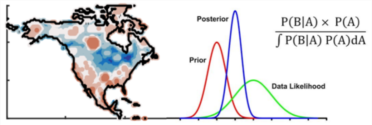


Reality check: binary classification $b_i \in \{\pm 1\}$, $a_i \in \mathbb{R}^{300}$, $\epsilon \sim 10^{-2}$, SVM loss with 5000 samples $\min_x f(x) = \sum_{i=1}^{5000} [1 - b_i \cdot a_i^\top x]_+$

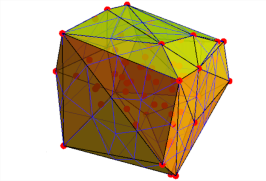
- (Sub)gradient descent: $\sim$ 650 iterations
- Parallel Stochastic method: $\sim$ 250 iterations

Statistical Computation and Sampling

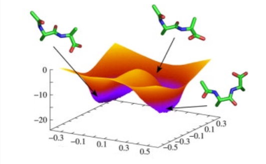
Sampling as an important algorithmic primitive



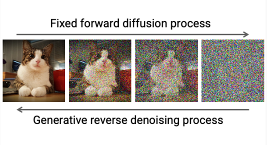
(a) Bayesian statistics / inverse problem



(b) Volume computation / counting



(c) Computational physics and chemistry



(d) Diffusion Generative Modeling

Goal:

> Draw samples from
>
> $$\pi \propto e^{-f} \text{ target density known up to normalizing constant}$$
>
> Design a process to gradually transform simple $\nu \to$ complicated π.

Two stops

- Optimization in $\mathcal{P}_2(\mathbb{R}^d)$ [J NeurIPS '21]: Mirror Langevin as geometry-aware MCMC sampling algorithm
- Borrow ideas from generative modeling [JN '24]: optimal stochastic control / optimal transport to steer a trajectory from ν to π using machine learning

Optimization in $\mathcal{P}_2(\mathbb{R}^d)$ and JKO Scheme

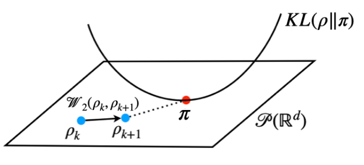
Deterministic Optimization in the space of probability measures

$$(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2)$$

Conceptually,

$$\underbrace{\boxed{\rho_{k+1}}}_{\text{“Prox step”}} = \arg\min_{\rho} \underbrace{\int \rho(x) \log \frac{\rho(x)}{\pi(x)}\, dx}_{\text{KL objective}} + \frac{1}{2h} \times \underbrace{\mathcal{W}_2^2(\rho, \rho_k)}_{\text{geometry}}$$

take h small, iterates $(\rho_k)_k$ trace out a curve of measures $(\rho_t)_t$ in $\mathcal{P}_2(\mathbb{R}^d)$ converging to π.



Optimization in $\mathcal{P}_2(\mathbb{R}^d)$ and JKO Scheme

Deterministic Optimization in the space of probability measures

$$(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2)$$

Conceptually,

$$\boxed{\rho_{k+1}} = \arg\min_{\rho} \underbrace{\int \rho(x) \log \frac{\rho(x)}{\pi(x)}\, dx}_{\text{KL objective}} + \frac{1}{2h} \times \underbrace{\mathcal{W}_2^2(\rho, \rho_k)}_{\text{geometry}}$$

"Prox step"

take h small, iterates $(\rho_k)_k$ trace out a curve of measures $(\rho_t)_t$ in $\mathcal{P}_2(\mathbb{R}^d)$ converging to π.

[JKO '98] Coincide with stochastic SDE dynamics $\rho_t = \mathrm{Law}(X_t)$:

$$dX_t = -\nabla f(X_t)\, dt + \sqrt{2}\, dW_t$$

Have $\pi \propto e^{-f}$ as long-time equilibrium and easy to discretize:

$$\boxed{x_{k+1} = x_k - h \cdot \nabla f(x_k) + \sqrt{2h} \cdot z_{k+1}}$$

Langevin MCMC

Converges to $\pi_h \neq \pi$ but $\pi_h \to \pi$ as $h \to 0$.

[JKO '98] Density $X_t \sim \rho_t$ along Langevin SDE dynamics

$$dX_t = -\nabla f(X_t)\, dt + \sqrt{2}\, dW_t$$

follows gradient flow of minimizing KL functional with $\mathcal{W}_2$ metric in $\mathcal{P}_2(\mathbb{R}^d)$

$$\boxed{\text{“}\dot{\rho}_t = -\nabla_{W_2} KL(\rho_t \| \pi)\text{”}}$$

Optimization in $\mathcal{P}_2(\mathbb{R}^d)$ and JKO Scheme

[JKO '98] Density $X_t \sim \rho_t$ along Langevin SDE dynamics

$$dX_t = -\nabla f(X_t)\,dt + \sqrt{2}\,dW_t$$

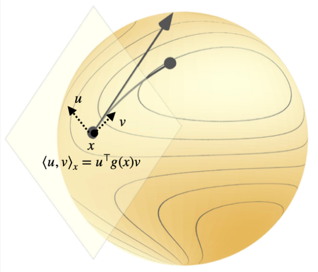
follows gradient flow of minimizing KL functional with $\mathcal{W}_2$ metric in $\mathcal{P}_2(\mathbb{R}^d)$

$$\boxed{\text{“}\dot{\rho}_t = -\nabla_{W_2} KL(\rho_t \| \pi)\text{”}}$$

? We know one can go from

$$(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{X}, g)$$

via mirror descent in optimization.



Is there a mirror flow analogue of Langevin?

$$(\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2) \to (\mathcal{P}_2(\mathcal{X}),\ \mathcal{W}_{2,g})$$

replace ground cost:
$\|\cdot\|_2 \to$ geodesic distance under g

Convergence and stable discretization?

Mirror Flow and Mirror Descent

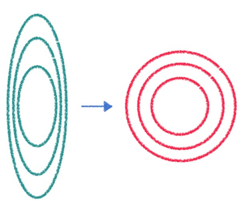
Mirror flow (in dual space) for bijective mapping $\nabla\phi \colon \mathcal{X} \to \mathbb{R}^d$, $\nabla^2\phi \succ 0$:

$$dY_t = -\nabla f(X_t)\, dt, \quad Y_t = \nabla\phi(X_t) \tag{1}$$

Same as (in primal space) Riemannian gradient flow over $(\mathcal{X}, \nabla^2\phi)$:

$$dX_t = -\underbrace{(\nabla^2\phi(X_t))^{-1}\nabla f(X_t)}_{\text{grad } f \text{ under metric } \nabla^2\phi}\, dt \tag{2}$$

⭐ Precondition for local geometry through choice of mirror map ϕ



Mirror Flow and Mirror Descent

Mirror flow (in dual space) for bijective mapping $\nabla\phi \colon \mathcal{X} \to \mathbb{R}^d$, $\nabla^2\phi \succ 0$:

$$dY_t = -\nabla f(X_t)\,dt, \quad Y_t = \nabla\phi(X_t) \tag{1}$$

Same as (in primal space) Riemannian gradient flow over $(\mathcal{X}, \nabla^2\phi)$:

$$dX_t = -\underbrace{(\nabla^2\phi(X_t))^{-1}\nabla f(X_t)}_{\text{grad } f \text{ under metric } \nabla^2\phi}\,dt \tag{2}$$

☆ Precondition for local geometry through choice of mirror map ϕ

Mirror descent discretizes (1):

$$x_{k+1} = \nabla\phi^*(\nabla\phi(x_k) - h \cdot \nabla f(x_k)) \tag{3}$$

Can invert $\nabla\phi^*$ numerically, i.e., convex optimization.

Ex: $\phi(x) = \frac{1}{2}\|x\|_2^2$ GD; $\phi(x) = \sum_i x_i \log(x_i)$ multiplicative weight. If $\phi = f$ Newton.

☆ E.g., $\min_{x\in\mathbb{R}^d} f(x)$: (3) allow regularity w.r.t norms beyond $\|\cdot\|_2$ without ∇^2

Mirror Descent: Application to Constrained Setup

Optimize $\min_{x \in \mathcal{X}} f(x)$: turn into Riemannian manifold by endowing $\mathcal{X}$ with metric $\nabla^2 \phi$ where $\|\nabla \phi(x)\| \to \infty$ as $x \to \partial \mathcal{X}$.

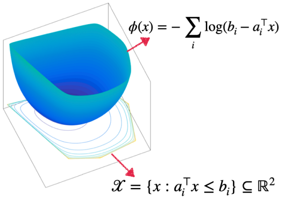


Figure 5: Log-barrier metric supported on a polytope

Mirror Descent: Application to Constrained Setup

Optimize $\min_{x \in \mathcal{X}} f(x)$: turn into Riemannian manifold by endowing $\mathcal{X}$ with metric $\nabla^2 \phi$ where $\|\nabla \phi(x)\| \to \infty$ as $x \to \partial \mathcal{X}$.

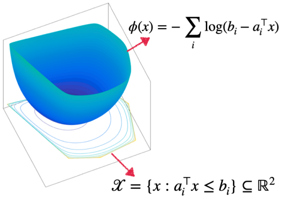


Figure 5: Log-barrier metric supported on a polytope

Primal $X \in \mathcal{X}$ constrained

$$\dot{X}_t = -(\nabla^2 \phi(X_t))^{-1} \nabla f(X_t) \leftarrow \text{Riemannian GF}$$

$$x_{k+1} = x_k - h \underbrace{(\nabla^2 \phi(x_k))^{-1} \nabla f(x_k)}_{\to 0} \text{ as } x_k \to \partial \mathcal{X}$$

[-] Can go out if $h \neq 0$, need $\nabla^2 \phi(\cdot)$

Optimize $\min_{x\in\mathcal{X}} f(x)$: turn into Riemannian manifold by endowing $\mathcal{X}$ with metric $\nabla^2\phi$ where $\|\nabla\phi(x)\| \to \infty$ as $x \to \partial\mathcal{X}$.

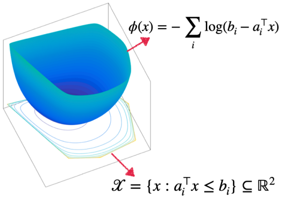


Figure 5: Log-barrier metric supported on a polytope

Primal $X \in \mathcal{X}$ constrained

$$\dot{X}_t = -(\nabla^2\phi(X_t))^{-1}\nabla f(X_t)$$

$$x_{k+1} = x_k - h\underbrace{(\nabla^2\phi(x_k))^{-1}\nabla f(x_k)}_{\to 0} \text{ as } x_k \to \partial\mathcal{X}$$

[-] Can go out if $h \neq 0$, need $\nabla^2\phi(\cdot)$

$$\xrightarrow[\nabla\phi:\ \mathcal{X} \to \mathbb{R}^d]{Y_t = \nabla\phi(X_t)}$$

Dual $Y \in \mathbb{R}^d$ un-constrained

$$\dot{Y}_t = -\nabla f(X_t) \leftarrow \text{Mirror Flow}$$

$$y_{k+1} = y_k - h\nabla f(x_k),\ x_{k+1} = \nabla\phi^*(y_{k+1})$$

[+] Never leave $\mathcal{X}$

[+] No need to evaluate $\nabla^2\phi(\cdot)$

Mirror Langevin: Continuous Time

Sample $\pi \propto e^{-f}$ supported on $\mathcal{X} \subseteq \mathbb{R}^d$.

Going from $(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{X}, g)$ to $(\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2) \to (\mathcal{P}_2(\mathcal{X}), \mathcal{W}_{2,g})$

Mirror Langevin SDE in **dual** space:

$$dY_t = -\nabla f(\nabla \phi^*(Y_t))\, dt + \sqrt{2[\nabla^2 \phi^*(Y_t)]^{-1}}\, dW_t, \quad Y_t = \nabla \phi(X_t)$$

Equivalent to Riemannian Langevin dynamics in **primal** space:

$$dX_t = (\nabla \cdot (\nabla^2 \phi(X_t)^{-1}) - \nabla^2 \phi(X_t)^{-1} \nabla f(X_t))\, dt + \sqrt{2 \nabla^2 \phi(X_t)^{-1}}\, dW_t$$

Mirror Langevin: Continuous Time

Sample $\pi \propto e^{-f}$ supported on $\mathcal{X} \subseteq \mathbb{R}^d$.

Going from $(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{X}, g)$ to $(\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2) \to (\mathcal{P}_2(\mathcal{X}), \mathcal{W}_{2,g})$

Mirror Langevin SDE in **dual** space:

$$dY_t = -\nabla f(\nabla \phi^*(Y_t))\, dt + \sqrt{2[\nabla^2 \phi^*(Y_t)]^{-1}}\, dW_t, \quad Y_t = \nabla \phi(X_t)$$

Equivalent to Riemannian Langevin dynamics in **primal** space:

$$dX_t = (\nabla \cdot (\nabla^2 \phi(X_t)^{-1}) - \nabla^2 \phi(X_t)^{-1} \nabla f(X_t))\, dt + \sqrt{2\nabla^2 \phi(X_t)^{-1}}\, dW_t$$

Recall $\nabla^2 \phi(X)^{-1} \to 0$ as $X \to \partial \mathcal{X}$ so $X_t \in \mathcal{X}$ always.

Mirror Langevin: Continuous Time

Sample $\pi \propto e^{-f}$ supported on $\mathcal{X} \subseteq \mathbb{R}^d$.

Going from $(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{X}, g)$ to $(\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2) \to (\mathcal{P}_2(\mathcal{X}), \mathcal{W}_{2,g})$

Mirror Langevin SDE in **dual** space:

$$dY_t = -\nabla f(\nabla \phi^*(Y_t))\, dt + \sqrt{2[\nabla^2 \phi^*(Y_t)]^{-1}}\, dW_t, \quad Y_t = \nabla \phi(X_t)$$

Equivalent to Riemannian Langevin dynamics in **primal** space:

$$dX_t = (\nabla \cdot (\nabla^2 \phi(X_t)^{-1}) - \nabla^2 \phi(X_t)^{-1} \nabla f(X_t))\, dt + \sqrt{2 \nabla^2 \phi(X_t)^{-1}}\, dW_t$$

Recall $\nabla^2 \phi(X)^{-1} \to 0$ as $X \to \partial \mathcal{X}$ so $X_t \in \mathcal{X}$ always.

GF interpretation of D_{KL} under $\mathcal{W}_{2,\nabla^2 \phi}$ $\rightsquigarrow$ “Wasserstein mirror flow” [Chewi et al ’20]

same objective

more general metric

Mirror Langevin: Continuous Time

Sample $\pi \propto e^{-f}$ supported on $\mathcal{X} \subseteq \mathbb{R}^d$.

Going from $(\mathbb{R}^d, \|\cdot\|_2) \to (\mathcal{X}, g)$ to $(\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2) \to (\mathcal{P}_2(\mathcal{X}), \mathcal{W}_{2,g})$

Mirror Langevin SDE in **dual** space:

$$dY_t = -\nabla f(\nabla\phi^*(Y_t))\,dt + \sqrt{2[\nabla^2\phi^*(Y_t)]^{-1}}\,dW_t, \quad Y_t = \nabla\phi(X_t)$$

Equivalent to Riemannian Langevin dynamics in **primal** space:

$$dX_t = (\nabla\cdot(\nabla^2\phi(X_t)^{-1}) - \nabla^2\phi(X_t)^{-1}\nabla f(X_t))\,dt + \sqrt{2\nabla^2\phi(X_t)^{-1}}\,dW_t$$

Mapping the diffusion process to dual space: a tractable SDE-dynamics that
(1) enjoy better geometric property for mixing;
(2) perform constrained sampling on compact, convex set $\mathcal{X}$

Mirror Langevin: Discretization

SDE in dual space:

$$dY_t = -\nabla f(X_t)\, dt + \sqrt{2[\nabla^2 \phi(X_t)]}\, dW_t, \quad Y_t = \nabla \phi(X_t)$$

Euler-Maruyama [Zhang, Peyré et al. ’20]

$$x_{k+1} = \nabla \phi^* \left(\nabla \phi(x_k) - h \cdot \nabla f(x_k) + \sqrt{2h} \cdot [\nabla^2 \phi(x_k)]^{1/2} \cdot z_{k+1} \right)$$

Asymptotic irreducible bias w.r.t diminishing step size $h \to 0$ generally.

$$dY_t = \underbrace{-\nabla f(X_t)\,dt}_{\text{deterministic, need to query } \nabla f} + \underbrace{\sqrt{2[\nabla^2\phi^*(Y_t)]^{-1}}\,dW_t}_{\text{stochastic, only involve } \phi}, \qquad Y_t = \nabla\phi(X_t)$$

Splitting Schemes (discretize objective but not geometry)

Forward Discretization:

$$\begin{cases} \bar{y} = \nabla\phi(x_k) - h \cdot \nabla f(x_k) \\ \text{solve } dy_t = \sqrt{2[\nabla^2\phi^*(y_t)]^{-1}}dW_t \text{ from initial } y_0 = \bar{y} \qquad (\clubsuit) \\ x_{k+1} = \nabla\phi^*(y_h) \end{cases}$$

Brownian motion ($\clubsuit$) can be solved approximately. Guarantee $x_k \in \mathcal{X}$ $\forall k$.

Bias-free Discretization Schemes [J NeurIPS '21]

deterministic, need to query ∇f

stochastic, only involve ϕ

$$dY_t = -\nabla f(X_t)\,dt + \sqrt{2[\nabla^2\phi^*(Y_t)]^{-1}}\,dW_t, \qquad Y_t = \nabla\phi(X_t)$$

Splitting Schemes (discretize objective but not geometry)

Forward Discretization:

$$\begin{cases} \bar{y} = \nabla\phi(x_k) - h\cdot\nabla f(x_k) \\ \text{solve } dy_t = \sqrt{2[\nabla^2\phi^*(y_t)]^{-1}}dW_t \ \text{ from initial } \ y_0 = \bar{y} \\ x_{k+1} = \nabla\phi^*(y_h) \end{cases} \qquad (\clubsuit)$$

Can also consider backward discretization: $\nabla f(x_k) \to \nabla f(x_{k+1})$.

Both bias-free as $h \to 0$.

Numerical Experiments

1. Ill-conditioned Gaussian ($d = 50$, $\kappa = 100$)

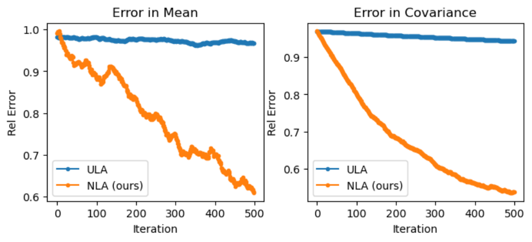
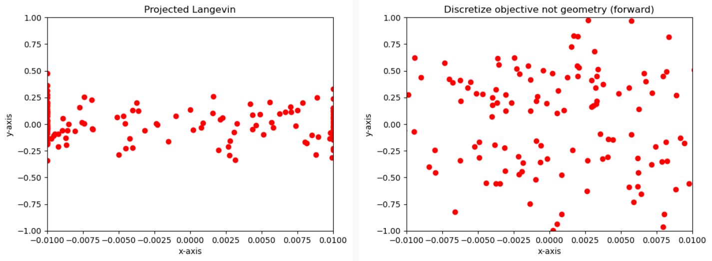


Figure 6: Error averaged over 100 parallel chains (mixing time $\frac{d}{\epsilon^2}$ vs. $\frac{\kappa d}{\epsilon^2}$ unadjusted Langevin)

2. Uniform sampling from 2D constrained ill-conditioned box $[-0.01, 0.01] \times [-1, 1]$



MCMC struggles with **multi-modality** in the target distribution. Alternatives?

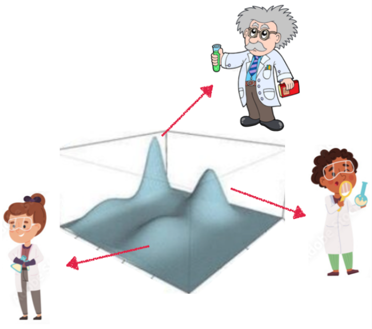
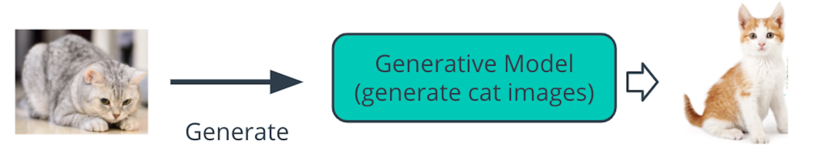


Figure 7: Probability distribution corresponding to image of scientists

Setup

Given many samples from a complex distribution π, generate more samples from it.



Diffusion Generative Modeling and Time Reversal SDE

With two path measures represented as (π is target, ν simple)

$$dX_t = \sigma u_t(X_t)\,dt + \sigma d\overrightarrow{W}_t,\ X_0 \sim \nu \Rightarrow (X_t)_t \sim \overrightarrow{\mathbb{P}}^{\nu,\sigma u}$$

$$X_{t+h} \approx X_t + h\sigma u_t(X_t) + \sqrt{h}\sigma z_t,\ X_0 \sim \nu$$

$$dX_t = \sigma v_t(X_t)\,dt + \sigma d\overleftarrow{W}_t,\ X_T \sim \pi \Rightarrow (X_t)_t \sim \overleftarrow{\mathbb{P}}^{\pi,\sigma v}$$

$$X_{t-h} \approx X_t + h\sigma v_t(X_t) + \sqrt{h}\sigma z_t,\ X_T \sim \pi$$

Interested in learning drifts u, v such that $D_{KL}(\overrightarrow{\mathbb{P}}^{\nu,\sigma u} || \overleftarrow{\mathbb{P}}^{\pi,\sigma v}) = 0$ or $D_{KL}(\overleftarrow{\mathbb{P}}^{\pi,\sigma v} || \overrightarrow{\mathbb{P}}^{\nu,\sigma u}) = 0$:

$$\boxed{\text{simple } \nu(x) \underset{\mathbb{P}^{\pi,\sigma v}}{\overset{\mathbb{P}^{\nu,\sigma u}}{\rightleftarrows}} \pi(x) \text{ target}}$$

Diffusion Generative Modeling and Time Reversal SDE

With two path measures represented as (π is target, ν simple)

$$dX_t = \sigma u_t(X_t)\,dt + \sigma \overrightarrow{dW_t},\ X_0 \sim \nu \Rightarrow (X_t)_t \sim \overrightarrow{\mathbb{P}}^{\nu,\sigma u}$$

$$dX_t = \sigma v_t(X_t)\,dt + \sigma \overleftarrow{dW_t},\ X_T \sim \pi \Rightarrow (X_t)_t \sim \overleftarrow{\mathbb{P}}^{\pi,\sigma v}$$

Interested in learning drifts u, v such that $D_{KL}(\overrightarrow{\mathbb{P}}^{\nu,\sigma u} \| \overleftarrow{\mathbb{P}}^{\pi,\sigma v}) = 0$ or $D_{KL}(\overleftarrow{\mathbb{P}}^{\pi,\sigma v} \| \overrightarrow{\mathbb{P}}^{\nu,\sigma u}) = 0$:

$$\text{simple } \nu(x) \xrightleftharpoons[\mathbb{P}^{\pi,\sigma v}]{\mathbb{P}^{\nu,\sigma u}} \pi(x) \text{ target}$$

Generative models: fix noising part $\overleftarrow{\mathbb{P}}^{\pi,\sigma v}$ (e.g., OU), learn NN-parameterized denoiser u using data from $\pi \rightsquigarrow \min_u D_{KL}(\overleftarrow{\mathbb{P}}^{\pi,\sigma v} \| \overrightarrow{\mathbb{P}}^{\nu,\sigma u})$ [Song et al '21]

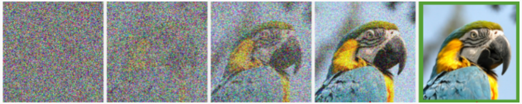


Figure 7: Generative Model: learning to denoise

Sampling by learning transition path

$$\boxed{\text{simple } \nu(x) \underset{\mathbb{P}^{\pi,\sigma \mathbf{v}}}{\overset{\mathbb{P}^{\nu,\sigma \mathbf{u}}}{\rightleftarrows}} \pi(x) \text{ target}}$$

Don't have samples from π: reverse KL, still fix v

Sampling by learning transition path

simple $\nu(x) \underset{\mathbb{P}^{\pi,\sigma v}}{\overset{\mathbb{P}^{\nu,\sigma u}}{\rightleftarrows}} \pi(x)$ target

Don't have samples from π: reverse KL, still fix v

$$D_{KL}(\overrightarrow{\mathbb{P}}^{\nu,\sigma u} \| \overleftarrow{\mathbb{P}}^{\pi,\sigma v}) = \mathbb{E}_{\overrightarrow{\mathbb{P}}^{\nu,\sigma u}}\left[\log\left(\frac{d\overrightarrow{\mathbb{P}}^{\nu,\sigma u}}{d\overleftarrow{\mathbb{P}}^{\pi,\sigma v}}\right)\right] = \mathbb{E}_{X\sim\overrightarrow{\mathbb{P}}^{\nu,\sigma u}}\left[\int_0^T \ldots (X_t)\, dt\right] =: \mathcal{L}_{KL}(u)$$

$\rightsquigarrow$ solution $\min_u \mathcal{L}_{KL}(u)$ is unique, resulting u^* can be used to transport ν to π [VGD '23]

Sampling by learning transition path

$$\text{simple } \nu(x) \overset{\mathbb{P}^{\nu,\sigma \mathbf{u}}}{\underset{\mathbb{P}^{\pi,\sigma \mathbf{v}}}{\rightleftarrows}} \pi(x) \text{ target}$$

Don’t have samples from π: reverse KL, still fix v

$$D_{KL}(\overrightarrow{\mathbb{P}}^{\nu,\sigma u} \| \overleftarrow{\mathbb{P}}^{\pi,\sigma v}) = \mathbb{E}_{\overrightarrow{\mathbb{P}}^{\nu,\sigma u}}\left[\log\left(\frac{d\overrightarrow{\mathbb{P}}^{\nu,\sigma u}}{d\overleftarrow{\mathbb{P}}^{\pi,\sigma v}}\right)\right] = \mathbb{E}_{X\sim\overrightarrow{\mathbb{P}}^{\nu,\sigma u}}\left[\int_0^T \dots(X_t)\,dt\right] =: \mathcal{L}_{KL}(u)$$

$\rightsquigarrow$ solution $\min_u \mathcal{L}_{KL}(u)$ is unique, resulting u^* can be used to transport ν to π [VGD ’23]

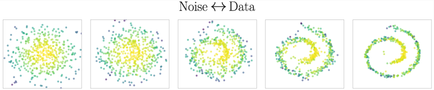


Figure 8: Interpolating Flow between ν and π

But $\overrightarrow{\mathbb{P}}_T^{\nu,\sigma u} = \pi$ only if $T \to \infty$.

Such forward/backward process is not unique, a better choice of $\overrightarrow{\mathbb{P}}^{\nu,\sigma u^*}$ corresponds to

stochastic optimal control

$$\min_u \mathbb{E}_u \left[\int_0^T \frac{1}{2} \|u_t(X_t)\|^2 dt \right]$$

$$\text{s.t.} \quad dX_t = \sigma u_t(X_t)\, dt + \sigma\, dW_t, \; X_0 \sim \nu, X_T \sim \pi$$

$\rightsquigarrow$ minimum control effort steering ν to π. Dynamics reaches target in finite time.

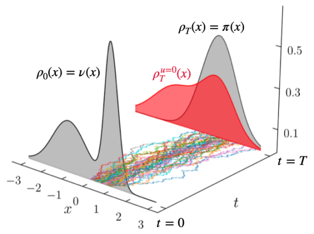


Figure 9: (Constrained) Optimization over path measure $\mathcal{P}_C([0,T],\mathbb{R})$

Such forward/backward process is not unique, a better choice of $\overrightarrow{\mathbb{P}}^{\nu,\sigma u^*}$ corresponds to

stochastic optimal control

$$\min_u \mathbb{E}_u \left[\int_0^T \frac{1}{2} \|u_t(X_t)\|^2 dt \right]$$

$$\text{s.t.} \quad dX_t = \sigma u_t(X_t)\, dt + \sigma\, dW_t, \; X_0 \sim \nu, X_T \sim \pi$$

$\rightsquigarrow$ minimum control effort steering ν to π. Dynamics reaches target in finite time.

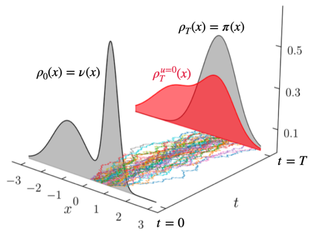


Figure 9: (Constrained) Optimization over path measure $\mathcal{P}_C([0,T],\mathbb{R})$

? Losses that can be used to train for a control u that follows an optimal trajectory w/o access to data from π?

Add regularizer to D_{KL} $\rightsquigarrow$ This imposes terminal marginals, uniqueness, and fulfills a reversible noising/denoising in an optimal way:

$$\boxed{\arg\min_{\nabla u, \nabla v} D_{KL}(\overrightarrow{\mathbb{P}}^{\nu,\sigma\nabla u} \| \overleftarrow{\mathbb{P}}^{\pi,\sigma\nabla v}) + \mathrm{Reg}(\nabla u) \text{ or } \mathrm{Reg}(\nabla v)}$$

Regularizer on the forward/backward control $\nabla u, \nabla v$ can be done in various ways using different perspectives on the SB problem: PDE, FBSDE, Optimal Transport [JN '24].

PINN

Feynman-Kac

Schrödinger system

Sampling as optimal control / transport of measure over path-space

Add regularizer to D_{KL} $\rightsquigarrow$ This imposes terminal marginals, uniqueness, and fulfills a reversible noising/denoising in an optimal way:

$$\arg\min_{\nabla u, \nabla v} D_{KL}(\overrightarrow{\mathbb{P}}^{\nu,\sigma\nabla u} \| \overleftarrow{\mathbb{P}}^{\pi,\sigma\nabla v}) + \mathrm{Reg}(\nabla u) \text{ or } \mathrm{Reg}(\nabla v)$$

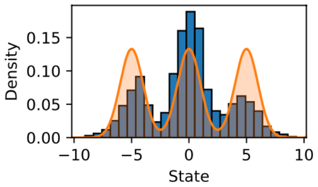
Regularizer on the forward/backward control $\nabla u, \nabla v$ can be done in various ways using different perspectives on the SB problem: PDE, FBSDE, Optimal Transport [JN '24].

PINN Feynman-Kac Schrödinger system

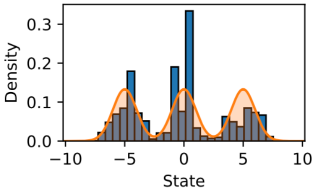
Algorithm

Alternate between:
(1) simulate trajectory $\overrightarrow{\mathbb{P}}^{\nu,\sigma\nabla u}$ with current control ∇u from ν;
(2) estimate loss $\mathcal{L}(\nabla u, \nabla v)$ above & update NN-parameterized controls $\nabla u, \nabla v$
$\rightsquigarrow$ if loss = 0, the controls found must be optimal

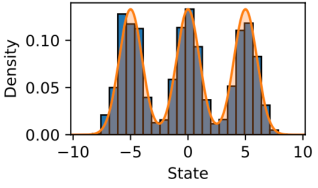
Experiment: Gaussian Mixture Model



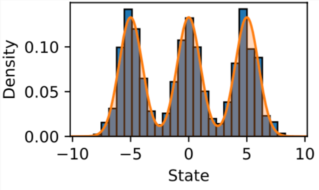
(a) No optimality enforced (Reg=0) [CLT '22]



(b) PDE-based Loss [VN '23]



(c) SDE-based Loss (ours)



(d) OT-based Loss (ours)

☆ This approach: reduce sampling to ERM with neural network.

Conclusion

I am particularly excited about:

- **Theoretically,** the connection between optimization, sampling, physics-inspired dynamical system (e.g., HMC, momentum), mean-field game goes much deeper

  interacting particle system

Conclusion

I am particularly excited about:

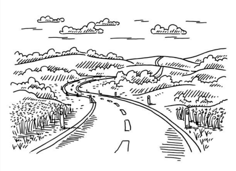
- **Theoretically,** the connection between optimization, sampling, physics-inspired dynamical system (e.g., HMC, momentum), mean-field game goes much deeper
- **Computationally,** bring powerful function fitting NN-architecture to solve more traditional tasks in sampling, control, PDE etc., is changing many areas of science

operator learning & harmonic analysis

Conclusion

I am particularly excited about:

- **Theoretically,** the connection between optimization, sampling, physics-inspired dynamical system (e.g., HMC, momentum), mean-field game goes much deeper
- **Computationally,** bring powerful function fitting NN-architecture to solve more traditional tasks in sampling, control, PDE etc., is changing many areas of science
- **Applications** in climate modeling (PDE), drug discovery & material design (sampling, generative modeling), single-cell genomics (optimal transport), ...



Thanks!

Questions?

Dissipation of Hamiltonian Monte Carlo Sampler

Motivation

Stan

Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

Radford Neal (2011) on Hamiltonian Monte Carlo:

> *“One practical impediment to the use of Hamiltonian Monte Carlo is* ***the need to select suitable values*** *for the leapfrog stepsize h, and the number of leapfrog steps K … Tuning HMC will usually require preliminary runs with trial values for h and K … Unfortunately, preliminary runs can be misleading …”*

Anatomy of HMC dynamics

Classical HMC alternates between:

(1) Follow deterministic Newtonian mechanics $\ddot{X}_t = -\nabla f(X_t)$

$$\boxed{\begin{cases} dX_t &= V_t\,dt \\ dV_t &= -\nabla f(X_t)\,dt \end{cases}}$$

for time T: define flow map $\phi_T(X_0, V_0) = (X_T, V_T)$

(2) Redraw the velocity $V_T \leftarrow Z \sim \mathcal{N}(0, I)$

$\rightsquigarrow$ **Piece-wise deterministic Markov process**

Anatomy of HMC dynamics

Classical HMC alternates between:

(1) Follow deterministic Newtonian mechanics $\ddot{X}_t = -\nabla f(X_t)$

$$\boxed{\begin{cases} dX_t &= V_t\, dt \\ dV_t &= -\nabla f(X_t)\, dt \end{cases}}$$

for time T: define flow map $\phi_T(X_0, V_0) = (X_T, V_T)$

(2) Redraw the velocity $V_T \leftarrow Z \sim \mathcal{N}(0, I)$

$\rightsquigarrow$ **Piece-wise deterministic Markov process**

Along dynamics (1), conservation of Hamiltonian $H(X, V) = f(X) + \frac{1}{2}\|V\|_2^2$ as

$$\frac{d}{dt}\left(f(X_t) + \frac{1}{2}\|V_t\|^2\right) = \nabla f(X_t)^\top V_t + V_t^\top(-\nabla f(X_t)) = 0$$

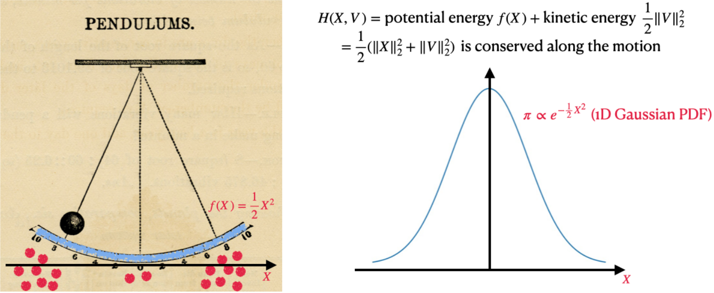
Stochasticity in (2) is needed for the dynamics to be a valid sampler, i.e.,

$$\mathrm{Law}(X_t, V_t) \to \pi(X) \otimes \mathcal{N}(0, I) \propto e^{-H(X,V)}$$

Ergodic: unique invariant measure (initial ρ_0 is eventually forgotten), or equivalently $\forall f$

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T f(x_t)\,dt = \int_{\mathbb{R}^d} f(x)\pi(x)\,dx$$

Imagine ensemble of particles (Ex: harmonic oscillator with potential $f(x) = \frac{1}{2}\|x\|^2$):



–If we initialize ρ_0 out of equlibrium (i.e., low-density region), with most particles at the tails, most will likely stay at the two ends

–If most are initialized around the center (i.e., ρ_0 near stationary π), one can show the distribution of particles will stay the same

Implies $\rho_T(X) \nrightarrow \pi(X)$ for all ρ_0 if the dynamics simply follow $\ddot{X}_t = -\nabla f(X_t)$

Parameter Tuning and Connection to Optimization

Two extremes:

- T too short: short deterministic dynamics $\rightsquigarrow$ random-walk-like diffusive behavior
- T too long: periodic $\rightsquigarrow$ backtrack on the progress made

Assuming quadratic potential with

$$\mu \cdot I \preceq \nabla^2 f \preceq L \cdot I, \quad \kappa := L/\mu$$

[Chen-Vempala '22] show for $T \asymp 1/\sqrt{L}$, mixing time in $\mathcal{W}_2$ is

$$\asymp \kappa \log(1/\epsilon) \times \frac{1}{\sqrt{L}} \asymp \frac{\sqrt{L}}{\mu} \log(1/\epsilon)$$

and this is tight.

Parameter Tuning and Connection to Optimization

Optimization

$\min_x f(x)$

Gradient Flow

$$dX_t = -\nabla f(X_t)\,dt$$

$$X_t \to X^*$$

ODE dissipates $f(X)$

rate $\kappa \to \sqrt{\kappa}$

Gradient Flow w/ Momentum

$$\begin{cases} dX_t &= V_t\,dt \\ dV_t &= -\nabla f(X_t) - \gamma V_t\,dt \end{cases}$$

$$(X_t, V_t) \to (X^*, 0)$$

2nd-order ODE dissipates $H(X,V) + \nabla D(Z)^\top S\,\nabla D(Z)$

$$D(Z) = \frac{1}{2}\|X - X^*\|_2^2 + \frac{1}{2}\|V\|_2^2$$

? What if we do **partial refreshment** (as inspired by accelerated gradient descent)?

1. Follow deterministic flow ϕ_T for time T
2. Redraw the velocity $V_T \leftarrow \eta V_T + \sqrt{1-\eta^2}Z$ for some $\eta > 0$

? What if we **randomize the integration time**?

1. Follow deterministic flow ϕ_T for time $T \sim \mathrm{Pois}(\lambda^{-1}) \leftarrow$ jump process
2. Redraw the velocity $V_T \leftarrow Z$

Dissipation of the Dynamics

Key Observation

For quadratic potential, both give improved performance by $\sqrt{\kappa}$ factor, i.e.,

$$\frac{\sqrt{L}}{\mu}\log(1/\epsilon) \to \frac{1}{\sqrt{\mu}}\log(1/\epsilon)$$

The crucial quantity is

$$\overset{\text{trajectory length}}{\lambda}^{-1}(1-\underset{\text{momentum}}{\eta}^{2}) \approx \sqrt{\mu}$$

with either $\eta = 0, \lambda^{-1} = \sqrt{\mu}$ or $1-\eta^2 = \sqrt{\mu}/\sqrt{L}, \lambda^{-1} = \sqrt{L}$, which compared to classical scaling

$$\lambda^{-1}(1-\eta^2) \approx \sqrt{L}$$

when $\eta = 0, \lambda^{-1} = \sqrt{L}$ can be much smaller, i.e., more inertia.

Argument based on (synchronous) coupling of two chains, challenge is using the right Lyapunov function over extended state-space $\mathbb{R}^{2d}$ for contraction.

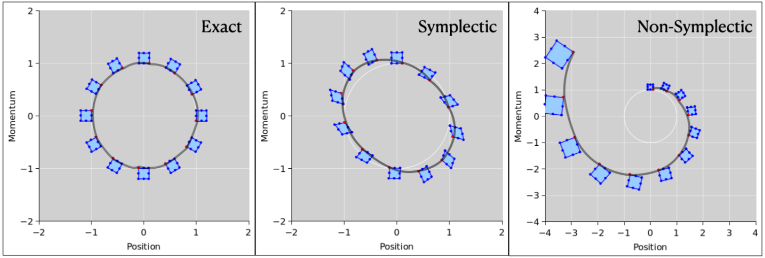
Discretization of Hamiltonian Dynamics

One gradient call, leapfrog (i.e., Verlet) for discretizing $\dot{X}_t = V_t, \dot{V}_t = -\nabla f(X_t)$:

$$
\begin{aligned}
x_{k+1/2} &= x_k + h/2 \cdot v_k \\
v_{k+1} &= v_k - h \cdot \nabla f(x_{k+1/2}) \\
x_{k+1} &= x_{k+1/2} + h/2 \cdot v_{k+1}
\end{aligned}
$$

Symplectic integrator:

- Simulate long trajectory w/o incur much err (flow preserve phase space volume)



- For quadratic there's a "shadow Hamiltonian" the discrete dynamics preserve $\rightsquigarrow$ invariant measure is another quadratic with shifted mean $\rightsquigarrow$ bias $\mathcal{O}(L\sqrt{d}h^2)$ in $\mathcal{W}_2$

Putting everything together

Dissipation-reduced HMC

$$\text{K-times, deterministic}\begin{cases} x_{k+1/2} = x_k + h/2 \cdot v_k \\ v_{k+1} = v_k - h \cdot \nabla f(x_{k+1/2}) \\ x_{k+1} = x_{k+1/2} + h/2 \cdot v_{k+1} \end{cases}$$

$$v_{k+1} = \eta \cdot v_{k+1} + \sqrt{1-\eta^2} \cdot Z$$

Stepsize $h \asymp \frac{\sqrt{\epsilon}}{\sqrt{L}d^{1/4}}$ determined by bias of deterministic part, $K = T/h$ steps of leapfrog, together w/ momentum η satisfies $1/Kh \cdot (1-\eta^2) \approx \sqrt{\mu}$ $\rightsquigarrow$ Total # gradient call:

$$\text{from} \quad \frac{\sqrt{L}}{\mu \cdot h} = \frac{\kappa d^{1/4}}{\sqrt{\epsilon}} \quad \text{to} \quad \frac{1}{\sqrt{\mu} \cdot h} = \frac{\sqrt{\kappa} d^{1/4}}{\sqrt{\epsilon}}$$

Improve on 1st-order over-damped Langevin: ($\frac{1}{\mu \cdot h} = \frac{\kappa d}{\epsilon^2}$ in $\mathcal{W}_2$)

$$dX_t = -\nabla f(X_t)\,dt + \sqrt{2}\,dW_t \quad \text{with } \mathrm{Law}(X_t) \to \pi.$$