Proximal algorithms for sampling and variational inference

Sinho Chewi
Institute for Advanced Study

September 14, 2023
University of Pennsylvania

Today's theme

The *entropy functional* is non-smooth, and therefore benefits from the use of *proximal methods*.

Outline:

- Review of proximal methods in optimization
- Sampling as optimization
- Gaussian variational inference via proximal gradient
- Log-concave sampling via the proximal sampler

Proximal discretization

For $f : \mathbb{R}^d \to \mathbb{R}$, the *gradient flow* of f is

$$\dot{x}_t = -\nabla f(x_t)\,.$$

We can discretize *explicitly*, via *gradient descent* (GD):

$$x_{k+1}^{\mathrm{GD}} = x_k^{\mathrm{GD}} - h\,\nabla f(x_k^{\mathrm{GD}})\,,$$

or *implicitly*, via the *proximal point method* (PPM):

$$x_{k+1}^{\mathrm{PPM}} = x_k^{\mathrm{PPM}} - h\,\nabla f(x_{k+1}^{\mathrm{PPM}})\,.$$

Unlike GD, the PPM does not require smoothness.

Proximal discretization

Reformulation of the PPM:

$$x_{k+1}^{\mathrm{PPM}} = \operatorname*{arg\,min}_{y \in \mathbb{R}^d} \left[f(y) + \frac{1}{2h} \|y - x_k^{\mathrm{PPM}}\|^2 \right] =: \operatorname{prox}_{hf}(x_k^{\mathrm{PPM}}) \,.$$

Proximal discretization

Reformulation of the PPM:

$$x_{k+1}^{\mathrm{PPM}} = \operatorname*{arg\,min}_{y \in \mathbb{R}^d} \left[f(y) + \frac{1}{2h} \|y - x_k^{\mathrm{PPM}}\|^2 \right] =: \operatorname{prox}_{hf}(x_k^{\mathrm{PPM}}) .$$

- For small h, implementation of the PPM is easier, but the overall convergence is slower.
- For large h, implementation of the PPM is harder, but we can nearly minimize f in one step (even non-convex f).

Convergence rate for the PPM

Suppose that f is α-convex. The gradient flow converges with rate

$$\|x_t - x_\star\|^2 \leq \exp(-2\alpha t)\, \|x_0 - x_\star\|^2$$

and the PPM converges with rate

$$\boxed{\|x_k^{\mathrm{PPM}} - x_\star\|^2 \leq \frac{1}{(1+\alpha h)^{2k}}\, \|x_0 - x_\star\|^2 .}$$

Sampling

If π is a probability distribution on $\mathbb{R}^d$ with known functional form, how do we efficiently draw samples from π?

- cornerstone of Bayesian sampling, high-dimensional integration, randomized algorithms, etc.
- typical approach: *Markov chain Monte Carlo* (MCMC)

Sampling

Suppose we write the density π in the form $\pi \propto \exp(-V)$, where $V : \mathbb{R}^d \to \mathbb{R}$. The *Langevin diffusion*

$$\mathrm{d}X_t = -\nabla V(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,, \qquad (B_t)_{t\geq 0} = \text{Brownian motion}\,,$$

converges in law to π as $t \to \infty$.

Sampling

Suppose we write the density π in the form $\pi \propto \exp(-V)$, where $V : \mathbb{R}^d \to \mathbb{R}$. The *Langevin diffusion*

$$\mathrm{d}X_t = -\nabla V(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\,, \qquad (B_t)_{t\geq 0} = \text{Brownian motion}\,,$$

converges in law to π as $t \to \infty$.

Key insight: Sampling is an optimization problem over the space of probability measures, and the Langevin diffusion is a *gradient flow*.

Sampling as optimization

Step one: Pass to the space of measures.

$$X_t \qquad \rightsquigarrow \qquad \mu_t := \mathsf{law}(X_t)\,.$$

Sampling as optimization

Step one: Pass to the space of measures.

$$X_t \qquad \rightsquigarrow \qquad \mu_t := \mathsf{law}(X_t)\,.$$

Kolmogorov's equations give us the evolution of $(\mu_t)_{t\geq 0}$, known as the *Fokker–Planck equation*.

$$\partial_t \mu_t = \underbrace{\operatorname{div}(\mu_t \nabla V)}_{\text{drift}} + \underbrace{\Delta \mu_t}_{\text{diffusion}}\,.$$

Sampling as optimization

Step one: Pass to the space of measures.

$$X_t \qquad \rightsquigarrow \qquad \mu_t := \mathsf{law}(X_t)\,.$$

Kolmogorov's equations give us the evolution of $(\mu_t)_{t\geq 0}$, known as the *Fokker–Planck equation*.

$$\partial_t \mu_t = \underbrace{\operatorname{div}(\mu_t \nabla V)}_{\text{drift}} + \underbrace{\Delta \mu_t}_{\text{diffusion}}\,.$$

How do we understand dynamics on the space of measures?

Sampling as optimization

Step two: Forget stochastic dynamics for the moment. Consider the *deterministic* dynamics

$$\dot{X}_t = v_t(X_t)\,, \qquad X_0 \sim \mu_0\,.$$

What is the evolution of $\mu_t = \mathrm{law}(X_t)$?

Sampling as optimization

Step two: Forget stochastic dynamics for the moment. Consider the *deterministic* dynamics

$$\dot{X}_t = v_t(X_t)\,, \qquad X_0 \sim \mu_0\,.$$

What is the evolution of $\mu_t = \mathrm{law}(X_t)$?

dynamics over $\mathbb{R}^d$	$\rightsquigarrow$	dynamics over $\mathcal{P}(\mathbb{R}^d)$
$\boxed{\dot{X}_t = v_t(X_t)}$	$\rightsquigarrow$	$\boxed{\partial_t \mu_t + \mathrm{div}(\mu_t v_t) = 0}$

This is known as the *continuity equation*.

Sampling as optimization

Step three: We can interpret the Fokker–Planck equation as a continuity equation:

$$\partial_t \mu_t = \operatorname{div}(\mu_t \nabla V) + \Delta \mu_t = \operatorname{div}(\mu_t (\nabla V + \nabla \log \mu_t)) .$$

Sampling as optimization

Step three: We can interpret the Fokker–Planck equation as a continuity equation:

$$\partial_t \mu_t = \operatorname{div}(\mu_t \nabla V) + \Delta \mu_t = \operatorname{div}(\mu_t (\nabla V + \nabla \log \mu_t)) \,.$$

$$\mathrm{d}X_t = -\nabla V(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

$$\downarrow$$

same evolution of law
$\partial_t \mu_t = \operatorname{div}(\mu_t (\nabla V + \nabla \log \mu_t))$

$$\uparrow$$

$$\dot{X}_t = -\nabla V(X_t) - \nabla \log \mu_t(X_t)$$

Sampling as optimization

Step four: We endow the space of measures with geometry.

At time t, the *kinetic energy* associated with $\dot{X}_t = v_t(X_t)$ is

$$\frac{1}{2}\int \underbrace{\|v_t\|^2}_{\text{squared velocity}} \underbrace{\mathrm{d}\mu_t}_{\text{mass density}} = \frac{1}{2}\|v_t\|^2_{L^2(\mu_t)}.$$

We can think of v_t as a *tangent vector* at μ_t, with norm $\|v_t\|_{L^2(\mu_t)}$.

Sampling as optimization

An energy-minimizing curve, i.e.,

$$(\mu_t, v_t)_{t\in[0,1]} \quad \text{minimizes} \quad \int_0^1 \|v_t\|^2_{L^2(\mu_t)}\,\mathrm{d}t \quad \text{with } \mu_0, \mu_1 \text{ fixed}$$

are *geodesics* or *shortest paths* in the space of measures, with

$$\boxed{\int_0^1 \|v_t\|^2_{L^2(\mu_t)}\,\mathrm{d}t = W_2^2(\mu_0, \mu_1)\,.}$$

Here, W_2 is the Wasserstein distance from *optimal transport*.

Sampling as optimization

Step five: Given a functional $\mathcal{F}$ over $\mathcal{P}(\mathbb{R}^d)$, we can now look for the *direction of steepest descent*:

$$-\nabla_{W_2}\mathcal{F}(\mu) := \underset{\|v_0\|_{L^2(\mu)}\leq 1}{\arg\min} \left\{ \partial_t \mathcal{F}(\mu_t)\big|_{t=0} \;\middle|\; \partial_t \mu_t + \operatorname{div}(\mu_t v_t) = 0,\ \mu_0 = \mu \right\}$$

Sampling as optimization

Step five: Given a functional $\mathcal{F}$ over $\mathcal{P}(\mathbb{R}^d)$, we can now look for the *direction of steepest descent*:

$$-\nabla_{W_2}\mathcal{F}(\mu) := \underset{\|v_0\|_{L^2(\mu)}\leq 1}{\arg\min} \left\{ \partial_t \mathcal{F}(\mu_t)\big|_{t=0} \;\middle|\; \partial_t \mu_t + \operatorname{div}(\mu_t v_t) = 0,\ \mu_0 = \mu \right\}$$

The *Wasserstein gradient flow* for $\mathcal{F}$ follows this direction:

$$\partial_t \mu_t = \operatorname{div}(\mu_t \nabla_{W_2}\mathcal{F}(\mu_t)) \,.$$

Sampling as optimization

Step six: If we compute the Wasserstein gradient for the *KL divergence* w.r.t. $\pi \propto \exp(-V)$,

$$\mathrm{KL}(\mu \,\|\, \pi) = \int \mu \log \frac{\mu}{\pi} = \int V \,\mathrm{d}\mu + \int \mu \log \mu + \text{const.},$$

one can show that

$$[\nabla_{W_2} \mathrm{KL}(\cdot \,\|\, \pi)](\mu) = \nabla V + \nabla \log \mu \,.$$

Sampling as optimization

Step six: If we compute the Wasserstein gradient for the *KL divergence* w.r.t. $\pi \propto \exp(-V)$,

$$\mathrm{KL}(\mu \,\|\, \pi) = \int \mu \log \frac{\mu}{\pi} = \int V \,\mathrm{d}\mu + \int \mu \log \mu + \text{const.},$$

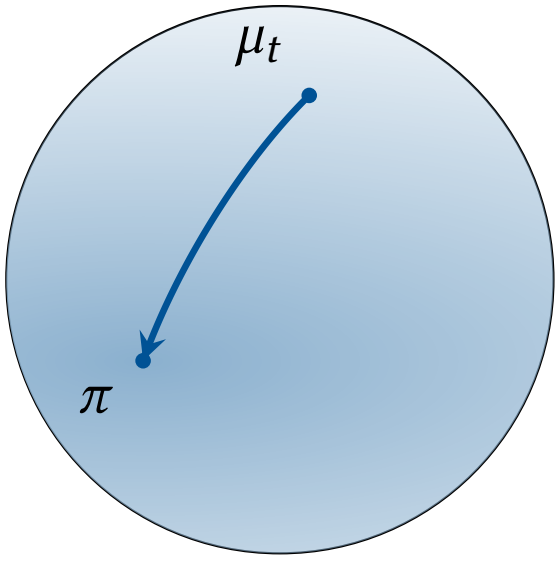
one can show that

$$[\nabla_{W_2} \mathrm{KL}(\cdot \,\|\, \pi)](\mu) = \nabla V + \nabla \log \mu \,.$$

The Wasserstein gradient flow for the KL agrees with the Fokker–Planck equation:

$$\partial_t \mu_t = \mathrm{div}(\mu_t \,(\nabla V + \nabla \log \mu_t)) \,.$$

Sampling as optimization



When we equip $\mathcal{P}(\mathbb{R}^d)$ with the geometry of optimal transport, the Langevin diffusion becomes a *gradient flow* of $\mathrm{KL}(\cdot \,\|\, \pi)$.

[Jordan, Kinderlehrer, Otto '98, *The variational formulation of the Fokker–Planck equation.*]

Sampling as optimization

In the aftermath of JKO…

- *algorithm analysis* [Dalalyan '17; Wibisono '18; Durmus, Majewski, Miasojedow '19; Ahn, **C.** '21; Altschuler, Talwar '22; *etc.*]
- *algorithm design* [**C.**, Le Gouic, Lu, Maunu, Rigollet, Stromme '20; Zhang, Peyré, Fadili, Pereyra '20; Ding, Li '21; Lee, Shen, Tian '21; *etc.*]
- *theory of complexity* [**C.**, Gerber, Lu, Le Gouic, Rigollet '22; **C.**, de Dios Pont, Li, Lu, Narayanan '23; **C.**, Gerber, Lee, Lu '23; *etc.*]

Sampling as optimization

In the aftermath of JKO…

- *algorithm analysis* [Dalalyan '17; Wibisono '18; Durmus, Majewski, Miasojedow '19; Ahn, **C.** '21; Altschuler, Talwar '22; *etc.*]
- *algorithm design* [**C.**, Le Gouic, Lu, Maunu, Rigollet, Stromme '20; Zhang, Peyré, Fadili, Pereyra '20; Ding, Li '21; Lee, Shen, Tian '21; *etc.*]
- *theory of complexity* [**C.**, Gerber, Lu, Le Gouic, Rigollet '22; **C.**, de Dios Pont, Li, Lu, Narayanan '23; **C.**, Gerber, Lee, Lu '23; *etc.*]

  ⊵ See my book draft if you are interested.

Sampling as optimization

Langevin diffusion
$$\mathrm{d}X_t = -\nabla V(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

$\rightsquigarrow$

gradient flow of KL
$$\mu \mapsto \int V\,\mathrm{d}\mu + \int \mu \log \mu$$

How do we design sampling algorithms in discrete time?

Potential energy

First, consider the *potential energy* functional

$$\boxed{\mathcal{V}(\mu) := \int V \, \mathrm{d}\mu \, .}$$

Potential energy

First, consider the *potential energy* functional

$$\boxed{\mathcal{V}(\mu) := \int V \, \mathrm{d}\mu \, .}$$

- V is α-convex $\iff$ $\mathcal{V}$ is α-convex on Wasserstein space

Potential energy

First, consider the *potential energy* functional

$$\boxed{\mathcal{V}(\mu) := \int V \, \mathrm{d}\mu \, .}$$

- V is α-convex $\iff$ $\mathcal{V}$ is α-convex on Wasserstein space
- V is β-smooth $\iff$ $\mathcal{V}$ is β-smooth on Wasserstein space

Potential energy

First, consider the *potential energy* functional

$$\boxed{\mathcal{V}(\mu) := \int V \, \mathrm{d}\mu \,.}$$

- V is α-convex $\iff$ $\mathcal{V}$ is α-convex on Wasserstein space
- V is β-smooth $\iff$ $\mathcal{V}$ is β-smooth on Wasserstein space
- GD step on V $\iff$ GD step on $\mathcal{V}$ in Wasserstein space

Entropy functional

The *entropy functional* is

$$\boxed{\mathcal{H}(\mu) := \int \mu \log \mu \, .}$$

The entropy is convex on Wasserstein space, but $\mathcal{H}(\mu) = \infty$ if $\mu \not\ll$ Lebesgue. Hence, the entropy is non-smooth.

Sampling as optimization

Langevin diffusion
$$\mathrm{d}X_t = -\nabla V(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$$

$\rightsquigarrow$

gradient flow of KL
$$\mu \mapsto \int V\,\mathrm{d}\mu + \int \mu \log \mu$$

After *discretization*,

$$X_{t+h} = X_t - h\nabla V(X_t) + \sqrt{2}\,(B_{t+h} - B_t) \quad \rightsquigarrow \quad ???$$

Sampling as optimization

$$\begin{array}{c}\text{Langevin diffusion} \\ \mathrm{d}X_t = -\nabla V(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t\end{array} \quad \rightsquigarrow \quad \begin{array}{c}\text{gradient flow of KL} \\ \mu \mapsto \int V\,\mathrm{d}\mu + \int \mu \log \mu\end{array}$$

After *discretization*, [Wibisono '18] showed:

$$X_t^+ = X_t - h\nabla V(X_t) \quad \rightsquigarrow \quad \text{gradient descent for } \mathcal{V}$$

$$X_{t+h} = X_t^+ + \sqrt{2}\,(B_{t+h} - B_t) \quad \rightsquigarrow \quad \text{gradient flow for } \mathcal{H}$$

He called this a *forward–flow* discretization.

Forward–flow is biased

In other words, for positive step size $h > 0$,

$$\mu_t \not\to \pi \,.$$

How can we remedy this issue?

Confronting the non-smoothness of entropy

In the original JKO paper, they proposed using the PPM for KL:

$$\mu_{t+h} = \mathrm{prox}_{h\,\mathrm{KL}}(\mu_t) := \operatorname*{arg\,min}_{\mu \in \mathcal{P}(\mathbb{R}^d)} \left\{ \mathrm{KL}(\mu \,\|\, \pi) + \frac{1}{2h} W_2^2(\mu, \mu_t) \right\}$$

Confronting the non-smoothness of entropy

In the original JKO paper, they proposed using the PPM for KL:

$$\mu_{t+h} = \operatorname{prox}_{h\,\mathsf{KL}}(\mu_t) := \operatorname*{arg\,min}_{\mu \in \mathcal{P}(\mathbb{R}^d)} \left\{ \mathsf{KL}(\mu \,\|\, \pi) + \frac{1}{2h}\, W_2^2(\mu, \mu_t) \right\}$$

Alternatively, one can split the objective and use proximal gradient (see analysis in [Salim, Korba, Luise '20]):

$$\mu_{t+h} = \operatorname{prox}_{h\,\mathcal{H}}\Big(\underbrace{(\mathrm{id} - h\nabla V)_{\#}\mu_t}_{\text{gradient step on } \mathcal{V}} \Big)$$

Confronting the non-smoothness of entropy

In the original JKO paper, they proposed using the PPM for KL:

$$\mu_{t+h} = \operatorname{prox}_{h\,\mathrm{KL}}(\mu_t) := \operatorname*{arg\,min}_{\mu \in \mathcal{P}(\mathbb{R}^d)} \left\{ \mathrm{KL}(\mu \,\|\, \pi) + \frac{1}{2h}\, W_2^2(\mu, \mu_t) \right\}$$

Alternatively, one can split the objective and use proximal gradient (see analysis in [Salim, Korba, Luise '20]):

$$\begin{aligned} \mu_{t+h} &= \operatorname{prox}_{h\,\mathcal{H}}\Big(\underbrace{(\mathrm{id} - h\nabla V)_{\#}\mu_t}_{\text{gradient step on } \mathcal{V}} \Big) \\ &= \operatorname*{arg\,min}_{\mu \in \mathcal{P}(\mathbb{R}^d)} \left\{ \mathcal{H}(\mu) + \frac{1}{2h}\, W_2^2\big(\mu, (\mathrm{id} - h\nabla V)_{\#}\mu_t\big) \right\}. \end{aligned}$$

Wasserstein proximal algorithms

However, these proximal operators are computationally **intractable**.
Is all hope for a Wasserstein proximal algorithm lost?

Restricting to a parametric class

Instead of minimizing over all probability measures, what if we minimize over a *parametric family* $\mathcal{P}$?

Restricting to a parametric class

Instead of minimizing over all probability measures, what if we minimize over a *parametric family* $\mathcal{P}$?

The parametric family must be specific:

- It should be convex in the Wasserstein geometry.
- The proximal operator should be computable over $\mathcal{P}$.
- Projections of gradients to $\mathcal{P}$ should be computable.

Gaussian variational inference

Problem: Compute the best Gaussian approximation to π in the sense of KL divergence $\mathrm{KL}(\cdot \,\|\, \pi)$.

This corresponds to $\mathcal{P} = \{\text{Gaussians over } \mathbb{R}^d\}$.

Gaussian VI is used to provide approximations to the mean and covariance of π which is hopefully cheaper than MCMC sampling.

Prior work: Lambert, **C.**, Bach, Bonnabel, Rigollet '22, *Variational inference via Wasserstein gradient flows.*

Proximal gradient for Gaussian VI

The set of Gaussians $\mathcal{P}$ is a *geodesically convex submanifold* of Wasserstein space. We consider the iteration

$$\begin{aligned}\mu_{k+1} &= \mathcal{P}\text{-prox}_{h\mathcal{H}}\left(\left[\mathrm{id} - h\,\mathrm{proj}_{T_{\mu_k}\mathcal{P}}\nabla_{W_2}\mathcal{V}(\mu_k)\right]_{\#}\mu_k\right) \\ &= \arg\min_{\mu\in\mathcal{P}}\left\{\mathcal{H}(\mu) + \frac{1}{2h}\,W_2^2\left(\left(\mathrm{id} - h\,\nabla_{\mathcal{P}}\mathcal{V}(\mu_k)\right)_{\#}\mu_k, \mu\right)\right\}.\end{aligned}$$

Proximal gradient for Gaussian VI

The set of Gaussians $\mathcal{P}$ is a *geodesically convex submanifold* of Wasserstein space. We consider the iteration

$$\begin{aligned}\mu_{k+1} &= \mathcal{P}\text{-prox}_{h\mathcal{H}}\left(\left[\mathrm{id} - h\,\mathrm{proj}_{T_{\mu_k}\mathcal{P}}\nabla_{W_2}\mathcal{V}(\mu_k)\right]_{\#}\mu_k\right)\\ &= \arg\min_{\mu\in\mathcal{P}}\left\{\mathcal{H}(\mu) + \frac{1}{2h}\,W_2^2\left(\left(\mathrm{id} - h\,\nabla_{\mathcal{P}}\mathcal{V}(\mu_k)\right)_{\#}\mu_k, \mu\right)\right\}.\end{aligned}$$

Michael Z. Diao, Krishnakumar Balasubramanian, **S.C.**, Adil Salim '23, *Forward–backward Gaussian variational inference via JKO in the Bures–Wasserstein space.*

Implementation

Proposition

1. The gradient $\nabla_{\mathcal{P}} \mathcal{V}(\mu)$ can be computed as

$$\nabla_{\mathcal{P}} \mathcal{V}(\mu) = (\mathbb{E}_\mu \nabla^2 V)(\cdot - m_\mu) + \mathbb{E}_\mu \nabla V,$$

   where m_μ is the mean of μ.

The gradient $\nabla_{\mathcal{P}} \mathcal{V}(\mu)$ can be approximated stochastically.

Implementation

Proposition

1. The gradient $\nabla_{\mathcal{P}} \mathcal{V}(\mu)$ can be computed as

$$\nabla_{\mathcal{P}} \mathcal{V}(\mu) = \left(\mathbb{E}_\mu \nabla^2 V\right)(\cdot - m_\mu) + \mathbb{E}_\mu \nabla V\,,$$

   where m_μ is the mean of μ.

2. [Wibisono '18] The $\mathcal{P}$-proximal operator for entropy is given by $\mathcal{P}\text{-prox}_{h\mathcal{H}}(\mathcal{N}(m,\Sigma)) = \mathcal{N}(m, f(\Sigma))$, where

$$f(x) = \left(x + 2h + \sqrt{x\,(x+4h)}\right)/2\,.$$

The gradient $\nabla_{\mathcal{P}} \mathcal{V}(\mu)$ can be approximated stochastically.

Implementation

Concretely, if $\mu_k = \mathcal{N}(m_k, \Sigma_k)$, we have the recursion:

FB–GVI (*forward–backward Gaussian variational inference*)

$$m_{k+1} = m_k - h\widehat{\mathbb{E}}_{\mu_k}\nabla V\,,$$
$$\Sigma_{k+1} = f\left((I - h\widehat{\mathbb{E}}_{\mu_k}\nabla^2 V)\,\Sigma_k\,(I - h\widehat{\mathbb{E}}_{\mu_k}\nabla^2 V)\right).$$

It's easy to implement and converges quickly in practice!

Convergence guarantees

Theorem: If $\alpha I \preceq \nabla^2 V \preceq \beta I$ and $\kappa := \beta/\alpha$, then FB–GVI outputs a measure ε-close to the best Gaussian approximation in $O(\kappa \log(d/\varepsilon^2))$ iterations.

If we use stochastic gradients, the iteration complexity instead becomes $\widetilde{O}(\kappa d/\varepsilon^2)$.

See the paper for other settings.

Back to sampling

Although the naïve application of proximal gradient to sampling is intractable, there is another approach, called the *proximal sampler* [Titsias, Papaspiliopoulos '18; Lee, Shen, Tian '21].

We first define a new sampling analogue of the proximal operator.

Restricted Gaussian oracle

Recall the analogy between sampling and optimization:

$$\text{minimize } f \qquad \rightsquigarrow \qquad \text{sample from } \pi \propto \exp(-V)$$

Restricted Gaussian oracle

Recall the analogy between sampling and optimization:

$$\text{minimize } f \qquad \rightsquigarrow \qquad \text{sample from } \pi \propto \exp(-V)$$

We therefore introduce the *restricted Gaussian oracle (RGO)*:

$$\begin{array}{c} x^+ = \operatorname{prox}_{hf}(x) \\ x^+ \text{ minimizes} \\ f(\cdot) + \frac{1}{2h}\|\cdot - x\|^2 \end{array} \qquad \rightsquigarrow \qquad \boxed{\begin{array}{c} x^+ \sim \mathrm{RGO}_{hV}(x) \\ x^+ \text{ is a sample from} \\ \propto \exp\left(-V(\cdot) - \frac{1}{2h}\|\cdot - x\|^2\right) \end{array}}$$

Derivation of the proximal sampler

If $X \sim \pi^X \propto \exp(-V)$, and $Y \mid X \sim \mathcal{N}(X, hI)$, let $\boldsymbol{\pi}$ denote the joint distribution of (X, Y):

$$\boldsymbol{\pi}(x, y) \propto \exp\left(-V(x) - \frac{1}{2h}\|y - x\|^2\right).$$

Derivation of the proximal sampler

If $X \sim \pi^X \propto \exp(-V)$, and $Y \mid X \sim \mathcal{N}(X, hI)$, let $\boldsymbol{\pi}$ denote the joint distribution of (X, Y):

$$\boldsymbol{\pi}(x, y) \propto \exp\left(-V(x) - \frac{1}{2h}\|y - x\|^2\right).$$

Observation: $\boldsymbol{\pi}^{X|Y=y} = \mathrm{RGO}_{hV}(y)$.

The proximal sampler

Algorithm (Gibbs sampling for π):

- Draw $Y_k \sim \pi^{Y|X=X_k} = \mathcal{N}(X_k, hI)$.
- Draw $X_{k+1} \sim \pi^{X|Y=Y_k} = \mathrm{RGO}_{hV}(Y_k)$.

The proximal sampler

Algorithm (Gibbs sampling for π):

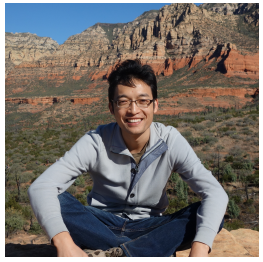
- Draw $Y_k \sim \pi^{Y|X=X_k} = \mathcal{N}(X_k, hI)$.
- Draw $X_{k+1} \sim \pi^{X|Y=Y_k} = \mathrm{RGO}_{hV}(Y_k)$.

Note: Gibbs sampling is automatically *unbiased*, unlike the forward–flow discretization. As $h \searrow 0$, one can indeed show this recovers the Langevin diffusion.

Convergence of the proximal sampler

Theorem [Lee, Shen, Tian '21; Chen, **C.**, Salim, Wibisono '22]
Suppose that V is α-convex. Then, the law $\mu_k^X := \mathrm{law}(X_k)$ of the proximal sampler converges to π^X at rate

$$W_2^2(\mu_k^X, \pi^X) \le \frac{1}{(1+\alpha h)^{2k}} W_2^2(\mu_0^X, \pi^X) .$$



Yongxin Chen, **S.C.**, Adil Salim, Andre Wibisono '22, *Improved analysis for a proximal algorithm for sampling.*

Implementation of the RGO

. . . will not be discussed today.

Recently, the proximal sampler has led to the first *high-accuracy samplers* with $\sqrt{d}$ *dimension dependence* in two concurrent works [Altschuler, **C.** '23; Fan, Yuan, Chen '23].

Jason M. Altschuler, **S.C.** '23, *Faster high-accuracy log-concave sampling via algorithmic warm starts.*

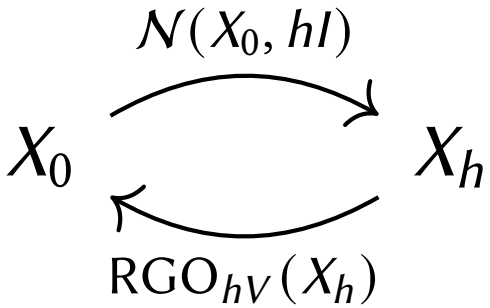
One last vignette: the SDE perspective

Instead of thinking of $Y \mid X \sim \mathcal{N}(X, hI)$, we can think of Y as the output of a *Brownian motion* $(X_t)_{t \geq 0}$ at time h, started from $X_0 = X$.

One last vignette: the SDE perspective

Instead of thinking of $Y \mid X \sim \mathcal{N}(X, hI)$, we can think of Y as the output of a *Brownian motion* $(X_t)_{t \geq 0}$ at time h, started from $X_0 = X$.

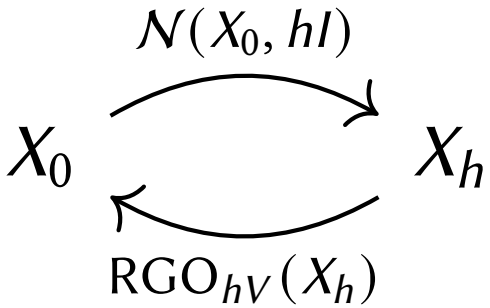
The two steps of the proximal sampler then correspond to running an SDE *forward and backward* in time.



One last vignette: the SDE perspective

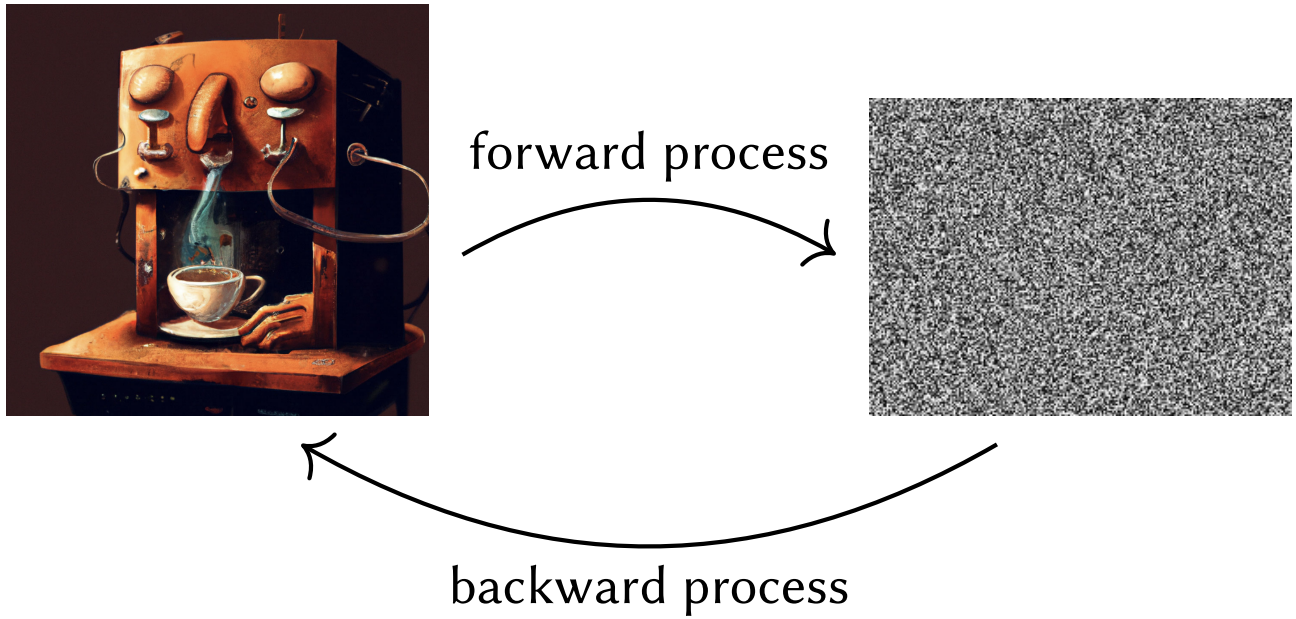
Instead of thinking of $Y \mid X \sim \mathcal{N}(X, hI)$, we can think of Y as the output of a *Brownian motion* $(X_t)_{t \geq 0}$ at time h, started from $X_0 = X$.

The two steps of the proximal sampler then correspond to running an SDE *forward and backward* in time.



Where have we seen the use of forward and backward SDEs elsewhere?

Diffusion models



Diffusion models are the "large step size" regime of the PPM. They converge rapidly (see growing literature) but implementation of the reverse process is difficult, requiring *deep learning*.

Outlook

Probabilistic problems involving the *non-smooth* entropy functional benefit from the use of *proximal methods*.

- ⊵ We saw this through forward–backward Gaussian variational inference and the proximal sampler.

Is there a deeper sense in which the proximal sampler is the true PPM analogue for sampling? How far can we push the analogy?

Thank you for your attention!