Discretization Issues in Sampling

Wasserstein Gradient Flow : Applications

September 17, 2025

Outline

1. Discretization in Optimization
2. Handling Non-Smoothness: Proximal Methods
3. Discretization in Sampling: Langevin Dynamics
4. Correcting Discretization Bias: MALA
5. Analysis of Metropolized Methods
6. Improving Conditioning: Proximal Sampling

Optimization Example: Finite Dimensions

Problem Setup

Minimize a smooth convex loss $\mathcal{L}(\theta)$ over $\theta \in \mathbb{R}^d$:

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta)$$

Example: Linear regression loss $\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|y_i - \theta^T x_i\|^2$.

Continuous Ideal: Gradient Flow

The path of steepest descent is given by the differential equation:

$$\dot{\theta}_t = -\nabla \mathcal{L}(\theta_t)$$

This continuous flow converges to the minimum under convexity.

Optimization Example: Finite Dimensions

Discrete Approximation: Gradient Descent

We approximate the flow using discrete steps (Euler discretization):

$$\theta_{k+1} = \theta_k - h\nabla\mathcal{L}(\theta_k)$$

Optimization Example: Finite Dimensions

Discrete Approximation: Gradient Descent

We approximate the flow using discrete steps (Euler discretization):

$$\theta_{k+1} = \theta_k - h\nabla\mathcal{L}(\theta_k)$$

- Convexity of $\mathcal{L}$ ensures convergence of the continuous time flow to the minimum

Optimization Example: Finite Dimensions

Discrete Approximation: Gradient Descent

We approximate the flow using discrete steps (Euler discretization):

$$\theta_{k+1} = \theta_k - h\nabla\mathcal{L}(\theta_k)$$

- Convexity of $\mathcal{L}$ ensures convergence of the continuous time flow to the minimum
- Smoothness of $\mathcal{L}$ ensures that the discretization is asymptotically unbiased (i.e, converges to the minimum)

Optimization over Measures

Problem Setup: Minimizing KL Divergence

Minimize KL divergence over probability measures $\mu \in \mathcal{P}(\mathbb{R}^d)$:

$$\min_{\mu \in \mathcal{P}(\mathbb{R}^d)} \mathrm{KL}(\mu \| \pi)$$

where $\pi(x) \propto e^{-V(x)}$ is the target measure. If μ has density ρ w.r.t. Lebesgue measure, this is equivalent to minimizing:

$$F(\mu) = \int V d\mu + \int \rho \log \rho \, dx + \mathrm{const}$$

This involves the negative entropy $-H(\mu) = \int \rho \log \rho \, dx$.

Optimization over Measures

Wasserstein Gradient Flow

We have seen the analogous gradient flow on the space of measures. We have also seen that $\mu \mapsto KL(\mu \mid \pi)$ is convex.

Optimization over Measures

Wasserstein Gradient Flow

We have seen the analogous gradient flow on the space of measures. We have also seen that $\mu \mapsto KL(\mu \mid \pi)$ is convex.

Discretization Challenge: Non-Smoothness?

The entropy term $H(\mu)$ behaves poorly. The function $x \mapsto x \log x$ has derivative $1 + \log x$, which blows up as $x \to 0^+$.

$$-H(\mu) = \int \mu(x) \log \mu(x) dx$$

This inherent non-smoothness (at the boundary of the space of measures) makes naive discretization of the corresponding gradient flow challenging. Smoothness conditions on V (e.g., $\nabla^2 V \leq \beta I$) are needed, but the entropy itself poses problems.

Example: L1 Regularization

Problem Setup

Consider minimizing a smooth loss plus an L1 penalty (LASSO):

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) + \lambda \|\theta\|_1$$

The L1 norm promotes sparsity but is non-differentiable at $\theta_i = 0$.

Example: L1 Regularization

Problem Setup

Consider minimizing a smooth loss plus an L1 penalty (LASSO):

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) + \lambda \|\theta\|_1$$

The L1 norm promotes sparsity but is non-differentiable at $\theta_i = 0$.

Problem with Naive Gradient Descent

A simple gradient step is ill-defined due to the non-differentiability:

$$\theta_{k+1} = \theta_k - h\nabla\mathcal{L}(\theta_k) - h\nabla(\lambda\|\theta_k\|_1) \quad \text{(Problematic! } \nabla\|\cdot\|_1 \text{ undefined)}$$

Proximal Gradient Descent

Algorithm: Proximal Gradient Descent

Split the objective into smooth ($\mathcal{L}$) and non-smooth ($\lambda\|\cdot\|_1$) parts.

Alternate between:

1. Gradient step on the smooth part:

$$\theta_{k+1/2} = \theta_k - h\nabla\mathcal{L}(\theta_k)$$

2. Proximal step on the non-smooth part:

$$\theta_{k+1} = \operatorname*{prox}_{h\lambda\|\cdot\|_1}\left(\theta_{k+1/2}\right)$$

Proximal Operator Definition

The proximal operator is defined as the solution to a small optimization problem:

$$\operatorname{prox}_g(z) = \arg\min_{\theta} \left\{ g(\theta) + \frac{1}{2}\|\theta - z\|^2 \right\}$$

For $g(\theta) = \gamma\|\theta\|_1$, this becomes:

$$\operatorname{prox}_{\gamma\|\cdot\|_1}(z) = \arg\min_{\theta\in\mathbb{R}^d} \left\{ \gamma\|\theta\|_1 + \frac{1}{2}\|\theta - z\|^2 \right\}$$

This has a closed-form solution: the *soft-thresholding* operator.

$$(\operatorname{prox}_{\gamma\|\cdot\|_1}(z))_i = \operatorname{sign}(z_i)\max(|z_i| - \gamma, 0)$$

Proximal Operator Definition

The proximal operator is defined as the solution to a small optimization problem:

$$\operatorname{prox}_{g}(z) = \arg\min_{\theta} \left\{ g(\theta) + \frac{1}{2}\|\theta - z\|^2 \right\}$$

For $g(\theta) = \gamma\|\theta\|_1$, this becomes:

$$\operatorname{prox}_{\gamma\|\cdot\|_1}(z) = \arg\min_{\theta\in\mathbb{R}^d} \left\{ \gamma\|\theta\|_1 + \frac{1}{2}\|\theta - z\|^2 \right\}$$

This has a closed-form solution: the *soft-thresholding* operator.

$$(\operatorname{prox}_{\gamma\|\cdot\|_1}(z))_i = \operatorname{sign}(z_i)\max(|z_i| - \gamma, 0)$$

This allows handling non-smooth terms rigorously and often efficiently.

Extension to Measures: Proximal Entropy

Goal

Apply the proximal idea to minimize functionals involving entropy $H(\mu)$, like the KL divergence.

Extension to Measures: Proximal Entropy

Goal

Apply the proximal idea to minimize functionals involving entropy $H(\mu)$, like the KL divergence.

Proximal Operator for Entropy (JKO Scheme Idea)

Define a proximal operator for the entropy functional $H(\mu)$:

$$\underset{hH}{\operatorname{prox}}(\mu_k) = \arg\min_{\nu \in \mathcal{P}(\mathbb{R}^d)} \left\{ H(\nu) + \frac{1}{2h} W_2^2(\nu, \mu_k) \right\}$$

where W_2 is the 2-Wasserstein distance. This finds a measure ν that balances minimizing entropy (spreading out) and staying close to μ_k in the Wasserstein sense.

Extension to Measures: Proximal Entropy

Challenge

This step defines the celebrated Jordan-Kinderlehrer-Otto (JKO) scheme, a variational approach to gradient flows in Wasserstein space. However, computing this proximal step is generally very difficult. "Not very nice". Alternatives like convolution methods can also be complex.

Extension to Measures: Proximal Entropy

Challenge

This step defines the celebrated Jordan-Kinderlehrer-Otto (JKO) scheme, a variational approach to gradient flows in Wasserstein space. However, computing this proximal step is generally very difficult. "Not very nice". Alternatives like convolution methods can also be complex.

So, how do we sample ??

Langevin Diffusion SDE

Goal

Sample from a target distribution $\pi(x) \propto e^{-V(x)}$.

Continuous Time Process: Langevin SDE

Consider the Stochastic Differential Equation (SDE):

$$dX_t = \underbrace{-\nabla V(X_t)dt}_{\text{Drift towards minimum}} + \underbrace{\sqrt{2}dB_t}_{\text{Random diffusion}}$$

where B_t is standard Brownian motion.

Langevin Diffusion SDE

Goal

Sample from a target distribution $\pi(x) \propto e^{-V(x)}$.

Continuous Time Process: Langevin SDE

Consider the Stochastic Differential Equation (SDE):

$$dX_t = \underbrace{-\nabla V(X_t)dt}_{\text{Drift towards minimum}} + \underbrace{\sqrt{2}dB_t}_{\text{Random diffusion}}$$

where B_t is standard Brownian motion.

Stationary Distribution

Under suitable conditions on V, the unique stationary (equilibrium) distribution of the process X_t is exactly the target distribution $\pi(x) \propto e^{-V(x)}$.

Problem: Euler-Maruyama Discretization (ULA)

Discrete Approximation: Unadjusted Langevin Algorithm (ULA)

Apply the simplest time discretization (Euler-Maruyama) to the Langevin SDE:

$$X_{k+1} = X_k - h\nabla V(X_k) + \sqrt{2h}Z_k, \quad Z_k \sim \mathcal{N}(0, I)$$

This is computationally simple: take a gradient step and add Gaussian noise.

Problem: Euler-Maruyama Discretization (ULA)

Discrete Approximation: Unadjusted Langevin Algorithm (ULA)

Apply the simplest time discretization (Euler-Maruyama) to the Langevin SDE:

$$X_{k+1} = X_k - h\nabla V(X_k) + \sqrt{2h} Z_k, \quad Z_k \sim \mathcal{N}(0, I)$$

This is computationally simple: take a gradient step and add Gaussian noise.

Issue: Discretization Bias

For any step size $h > 0$, the Markov chain generated by ULA does **not** have π as its exact stationary distribution.

$$\pi_{ULA}^{(h)} \neq \pi$$

The error $||\pi_{ULA}^{(h)} - \pi||_{W_2}$ depends on h. Achieving high accuracy requires very small h, which is inefficient.

Solution: Metropolis-Adjusted Langevin Algorithm (MALA)

Idea

Use the ULA step as a proposal mechanism within a Metropolis-Hastings framework to correct the discretization bias.

Solution: Metropolis-Adjusted Langevin Algorithm (MALA)

Theorem 1 (MALA Algorithm)

Given current state X_k:

1. **Propose:** *Generate a candidate Y using one ULA step:*

$$Y = X_k - h\nabla V(X_k) + \sqrt{2h}\,Z_k, \quad Z_k \sim \mathcal{N}(0, I).$$

2. **Accept/Reject:** *Accept Y with probability $\alpha(X_k, Y)$:*

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

where $q(x, y) \propto \exp\left(-\frac{\|y-(x-h\nabla V(x))\|^2}{4h}\right)$ is the proposal transition density.

3. **Update:** *If accepted, $X_{k+1} = Y$; otherwise, $X_{k+1} = X_k$.*

Solution: Metropolis-Adjusted Langevin Algorithm (MALA)

Key Property: The MALA Markov chain has **exactly** $\pi(x) \propto e^{-V(x)}$ as its invariant distribution for **any** step size $h > 0$. (Though acceptance rate decreases for large h).

Connection to Metropolized HMC

Literature

Appendix A of *[Logsmooth Gradient Concentration and Tighter Runtimes for Metropolized Hamiltonian Monte Carlo, Lee et. al. '20]* establishes an equivalence.

Connection to Metropolized HMC

Literature

Appendix A of *[Logsmooth Gradient Concentration and Tighter Runtimes for Metropolized Hamiltonian Monte Carlo, Lee et. al. '20]* establishes an equivalence.

Equivalence Result

A specific variant of Metropolized Hamiltonian Monte Carlo (MHMC) using exactly **one** leapfrog integration step is **algorithmically equivalent** to MALA.

- Both generate proposals of the form:

  $$Y = X_k - c\nabla V(X_k) + \mathcal{N}(0, \sigma^2 I)$$

  (with constants matched appropriately).
- Both use a Metropolis-Hastings correction based on the target π.

Connection to Metropolized HMC

Implication

Rigorous analysis and theoretical guarantees (like mixing time bounds) derived for 1-step MHMC directly apply to MALA.

Theorem: Mixing Time of MALA

Theorem 2 (Mixing of Metropolized Langevin Dynamics (via MHMC))

Let $V : \mathbb{R}^d \to \mathbb{R}$ be L-smooth ($\nabla^2 V \preceq LI$) and μ-strongly convex ($\nabla^2 V \succeq \mu I$). Let $\kappa = L/\mu \geq 1$ be the condition number. Target distribution is $\pi(x) \propto e^{-V(x)}$. Assume initialization $X_0 \sim \mathcal{N}(x^*, L^{-1}I)$, where x^* minimizes V.
Then there exists $C > 0$ such that for k iterations where

$$k = O\left(\kappa\, d \log\left(\frac{\kappa}{\epsilon}\right) \log\left(d \log \frac{\kappa}{\epsilon}\right) \log\left(\log \frac{\kappa}{\epsilon}\right)\right) = \tilde{O}(\kappa d)$$

the distribution ρ_k of X_k satisfies Total Variation distance:

$$\|\rho_k - \pi\|_{TV} \leq \epsilon.$$

($\tilde{O}$ hides logarithmic factors in $\kappa, d, 1/\epsilon$).

Theorem: Mixing Time of MALA (Discussion)

Remark

The previous theorem shows that MALA achieves ϵ-accuracy in sampling π after $\tilde{O}(\kappa d)$ steps. This is efficient, but the dependence on the condition number κ can be problematic if κ is large.

The Problem of Large κ

Recap

The mixing time of MALA scales as $\tilde{O}(\kappa d)$.

Challenge: Ill-Conditioned Problems

If the potential V is poorly conditioned (e.g., features have vastly different scales), $\kappa = L/\mu$ can be very large. This makes the $\tilde{O}(\kappa d)$ mixing time prohibitive.

The Problem of Large κ

Recap

The mixing time of MALA scales as $\tilde{O}(\kappa d)$.

Challenge: Ill-Conditioned Problems

If the potential V is poorly conditioned (e.g., features have vastly different scales), $\kappa = L/\mu$ can be very large. This makes the $\tilde{O}(\kappa d)$ mixing time prohibitive.

Goal

Can we modify the sampling approach to mitigate the dependence on the original κ?

Sampling :: Optimization Analogies

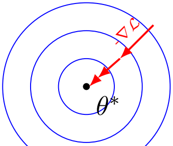
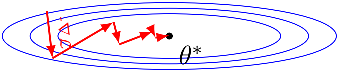
To solve this problem, we would like to draw some analogies with optimization.

Sampling :: Optimization Analogies

To solve this problem, we would like to draw some analogies with optimization.

Note that sampling from $\pi \propto e^{-V(x)}$ is equivalent to optimization of a certain functional (specifically, $\int V d\mu + \int \frac{d\mu}{d\lambda} \log \frac{d\mu}{d\lambda}$) over the space of measures.

Sampling :: Optimization Analogies

Sampling	**Optimization**
$\pi(x) \propto e^{-V(x)}$	$\min_\theta \mathcal{L}(\theta)$
Ill-Conditioned Density $\kappa_V = L/\mu \gg 1$ Elongated contours $\implies$ Slow Mixing (ULA/MALA)	**Ill-Conditioned Objective** $\kappa_{\mathcal{L}} = L/\mu \gg 1$ Elongated level sets $\implies$ Slow Convergence (GD)
Well-conditioned ($\kappa \approx 1$)  	Ill-conditioned ($\kappa \gg 1$) 

Sampling :: Optimization Analogies

- We have already dealt with a rather extreme case of ill-conditioned objective earlier: $\int V d\mu + \int \frac{d\mu}{d\lambda} \log \frac{d\mu}{d\lambda}$.

- Specifically, the entropy term $\int \frac{d\mu}{d\lambda} \log \frac{d\mu}{d\lambda}$ is non-smooth, making its condition number infinity!

Sampling :: Optimization Analogies

- We have already dealt with a rather extreme case of ill-conditioned objective earlier: $\int V d\mu + \int \frac{d\mu}{d\lambda} \log \frac{d\mu}{d\lambda}$.

- Specifically, the entropy term $\int \frac{d\mu}{d\lambda} \log \frac{d\mu}{d\lambda}$ is non-smooth, making its condition number infinity!

- The way we dealt with this was to use the **proximal operator**.

Proximal Methods Comparison

Sampling	**Optimization**
Proximal Sampler (Gibbs) Step: $X \sim p(x \mid y)$ where $p(x \mid y) \propto e^{-V(x)-\frac{1}{2h}\Vert x-y\Vert^2}$ Mode is $\text{prox}_{hV}(y)$ Benefit: Uses well-conditioned $V_y(x) = V(x) + \frac{1}{2h}\Vert x-y\Vert^2$	**Proximal Gradient Desc.** Update: $\theta_{k+1} = \text{prox}_{hg}(\theta_k - h\nabla f(\theta_k))$ for min $f(\theta) + g(\theta)$ Benefit: Handles non-smooth $g(\theta)$
(Leveraging proximal structure $\min_x\{Func(x) + \frac{1}{2\lambda}\Vert x-z\Vert^2\}$ *)*	

Proximal Sampling Algorithm

Augmented Target Distribution

Define a joint distribution $\boldsymbol{\pi}$ on $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$:

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

Note: The marginal distribution of x under $\boldsymbol{\pi}$ is the original target $\pi(x) \propto e^{-V(x)}$. $\int \boldsymbol{\pi}(x, y)dy \propto e^{-V(x)}$.

Proximal Sampling Algorithm

Proximal Sampler (Gibbs Sampling on π)

Iterate the following steps:

1. Given $X_k = x$, sample $Y_{k+1} \sim \boldsymbol{\pi}^{Y|X=x}$:

$$Y_{k+1} \sim \mathcal{N}(x, hI) \quad \text{(Simple Gaussian step)}$$

2. Given $Y_{k+1} = y$, sample $X_{k+1} \sim \boldsymbol{\pi}^{X|Y=y}$:

$$\boldsymbol{\pi}^{X|Y=y}(x) \propto \exp\left(-V(x) - \frac{1}{2h}\|y - x\|^2\right) \quad \text{(RGO Sampling)}$$

This Gibbs sampler targets $\boldsymbol{\pi}$, so the X samples marginally target π.

Theorem: Improved Conditioning

Implementing Step 2: Sampling from RGO

Step 2 requires sampling from $\exp(-V_y(x))$, where the potential is modified:

$$V_y(x) = V(x) + \frac{1}{2h}\|y - x\|^2$$

We can use MALA (or another sampler) for this sub-problem.

Theorem: Improved Conditioning

Condition Number Improvement

Let $\mu I \preceq \nabla^2 V(x) \preceq LI$. The Hessian of the modified potential $V_y(x)$ is:

$$\nabla^2 V_y(x) = \nabla^2 V(x) + \frac{1}{h} I$$

The condition number κ_y for the sub-problem is: $\kappa_y = \frac{L+1/h}{\mu+1/h}$

Theorem: Improved Conditioning

Condition Number Improvement

Let $\mu I \preceq \nabla^2 V(x) \preceq LI$. The Hessian of the modified potential $V_y(x)$ is:

$$\nabla^2 V_y(x) = \nabla^2 V(x) + \frac{1}{h} I$$

The condition number κ_y for the sub-problem is: $\kappa_y = \frac{L+1/h}{\mu+1/h}$

Limiting Behavior

As the proximal parameter $h \to 0$:

$$\lim_{h \to 0} \kappa_y = \lim_{h \to 0} \frac{hL + 1}{h\mu + 1} = \frac{1}{1} = 1$$

For small h, the condition number of the MALA sub-problem becomes $\kappa_y \approx 1$, regardless of the original $\kappa = L/\mu$.

Conclusion: Benefits of Proximal Sampling

Key Idea

Proximal sampling alternates between:

1. A simple Gaussian step $Y \sim \mathcal{N}(X, hI)$.
2. Sampling X from a modified potential $V_y(x) = V(x) + \frac{1}{2h}\|y - x\|^2$.

Conclusion: Benefits of Proximal Sampling

Key Idea

Proximal sampling alternates between:

1. A simple Gaussian step $Y \sim \mathcal{N}(X, hI)$.
2. Sampling X from a modified potential $V_y(x) = V(x) + \frac{1}{2h}\|y - x\|^2$.

Advantage

The second step (sampling from $\exp(-V_y)$) can be done using MALA. Crucially, the potential $V_y(x)$ has a much better condition number $\kappa_y \approx 1$ for small h.

Conclusion: Benefits of Proximal Sampling

Result

Applying Theorem 2 to the MALA sub-problem suggests its mixing time is $\tilde{O}(\kappa_y d) \approx \tilde{O}(d)$. This removes the potentially large factor κ from the MALA iterations within the proximal sampler. Proximal sampling offers a way to efficiently sample from ill-conditioned distributions $\pi \propto e^{-V}$ by solving a sequence of well-conditioned sub-problems.