CPSC 661: Sampling Algorithms in ML

Andre Wibisono

April 19, 2021

Yale University

Last time

- Wasserstein W_2 metric, Otto calculus
- Langevin dynamics in continuous time
  Exponential convergence rate under SLC $\Rightarrow$ LSI $\Rightarrow$ PI
- Unadjusted Langevin Algorithm in discrete time

Today: Convergence rate of ULA under SLC $\Rightarrow$ LSI

References

- Dalalyan, *Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent*, COLT 2017
- Durmus, Majewski, & Miasojedow, *Analysis of Langevin Monte Carlo via Convex Optimization*, JMLR, 2019
- Wibisono, *Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem*, COLT 2018
- Vempala & Wibisono, *Rapid Convergence of the Unadjusted Langevin Algorithm: Isoperimetry Suffices*, NeurIPS 2019

Recap: ULA

Unadjusted Langevin Algorithm

The **Unadjusted Langevin Algorithm (ULA)** in discrete time to sample from $\nu \propto e^{-f}$ on $\mathbb{R}^n$ is

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, Z_k$$

where $\eta > 0$ is step size and $Z_k \sim \mathcal{N}(0, I)$ is independent of x_k

- Discretization of the Langevin dynamics:

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

- Biased: Converges to $\nu_\eta \neq \nu$

ULA in the space of distributions

ULA:

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, Z_k$$

If $X_k \sim \rho_k$, then $X_{k+1} \sim \rho_{k+1}$

$$\rho_{k+1} = \underbrace{\big((I - \eta \nabla f)_{\#} \rho_k\big)}_{\substack{\text{gradient descent} \\ \text{to min } F(\rho) = \mathbb{E}_\rho[f]}} \underbrace{* \mathcal{N}(0, 2\eta I)}_{\substack{\text{gradient flow for min} \\ -H(\rho) = \mathbb{E}_\rho[\log \rho]}}$$

ULA as Forward-Flow

Relative entropy with respect to $\nu = e^{-f}$ is a composite objective

$$H_\nu(\rho) = F(\rho) - H(\rho)$$

$$\mathbb{E}_\rho\left[\log \frac{\rho}{\nu}\right] = \mathbb{E}_\rho[f] + \mathbb{E}_\rho[\log \rho]$$

- **ULA** is Forward-Flow algorithm for minimizing relative entropy
- Forward-Flow is in general biased for composite optimization
- Should use e.g. Forward-Backward algorithm

Composite optimization

Composite optimization

To optimize a composite objective function:

$$\min_{x \in \mathbb{R}^n} f(x) + g(x)$$

For f (and similarly g) have basic optimization methods:

- Forward method (gradient descent):

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

- Flow method (gradient flow):

$$x_{k+1} = X_\eta \text{ (solution to } \dot{X}_t = -\nabla f(X_t) \text{ from } X_0 = x_k)$$

- Backward method (proximal method):

$$x_{k+1} = x_k - \eta \nabla f(x_{k+1}) = \arg\min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2\eta} \|x - x_k\|^2 \right\}$$

Forward-Backward algorithm

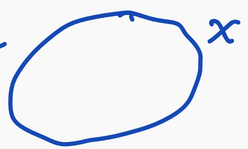
$$\min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Forward-Backward algorithm: $= \text{Backward}_g \circ \text{Forward}_f$

$$x_{k+\frac{1}{2}} = x_k - \eta \nabla f(x_k)$$

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{1}{2\eta} \|x - x_{k+\frac{1}{2}}\|^2 \right\}$$

$$g(x) = \begin{cases} 0 & \text{if } x \in \mathcal{X} \\ \infty & \text{else} \end{cases}$$



- E.g. constrained optimization $\min_{x \in \mathcal{X}} f(x)$: $g(x) = 1_{\mathcal{X}}(x)$
  Forward-Backward algorithm = projected gradient descent

Forward-Backward algorithm is consistent

Lemma: Forward-Backward algorithm preserves the minimizer

$$x^* = \arg\min_{x \in \mathbb{R}^n} f(x) + g(x)$$

if $x_n = x^*$,

then $x_{n+1} = x^*$

Forward-Backward algorithm is consistent

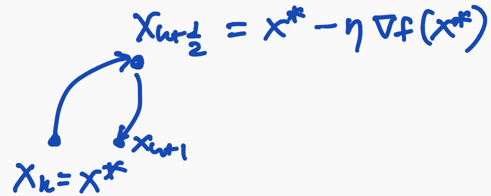
Lemma: Forward-Backward algorithm preserves the minimizer

$$x^* = \arg\min_{x \in \mathbb{R}^n} f(x) + g(x)$$

$\nabla f(x^*) + \nabla g(x^*) = 0 \iff \nabla f(x^*) = -\nabla g(x^*) \neq 0$

Proof: Suppose $x_k = x^*$. In the first half-step:

$$x_{k+\frac{1}{2}} = x^* - \eta \nabla f(x^*)$$



In the second half-step:

$$x_{k+1} = x_{k+\frac{1}{2}} - \eta \nabla g(x_{k+1})$$

or equivalently

$$x_{k+1} + \eta \nabla g(x_{k+1}) = x^* - \eta \nabla f(x^*).$$

Since $\nabla f(x^*) + \nabla g(x^*) = 0$, we have $\nabla g(x^*) = -\nabla f(x^*)$, so a solution is $x_{k+1} = x^*$. □

Forward-Backward algorithm is consistent

Forward-Backward algorithm for $\min_{x \in \mathbb{R}^n} f(x) + g(x)$:

$$x_{k+\frac{1}{2}} = \text{Forward}_f(x_k) = x_k - \eta \nabla f(x_k)$$

$$x_{k+1} = \text{Backward}_g(x_{k+\frac{1}{2}}) = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{1}{2\eta} \|x - x_{k+\frac{1}{2}}\|^2 \right\}$$

- Consistent because Backward is the *adjoint* of the Forward method (The *adjoint* of an integrator A_f is $A_f^* = (A_{-f})^{-1}$)

$$\text{Forward}_f = (I - \eta \nabla f)$$

$$\text{Backward}_f = (I + \eta \nabla f)^{-1}$$

Forward-Backward algorithm is consistent

Forward-Backward algorithm for $\min_{x \in \mathbb{R}^n} f(x) + g(x)$:

$$x_{k+\frac{1}{2}} = \text{Forward}_f(x_k) = x_k - \eta \nabla f(x_k)$$

$$x_{k+1} = \text{Backward}_g(x_{k+\frac{1}{2}}) = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{1}{2\eta} \|x - x_{k+\frac{1}{2}}\|^2 \right\}$$

- Consistent because Backward is the *adjoint* of the Forward method

  (The *adjoint* of an integrator A_f is $A_f^* = (A_{-f})^{-1}$)

- Can also do e.g. Backward-Forward or Flow-Flow algorithm:

$$x_{k+1} = (\text{Flow}_g \circ \text{Flow}_f)(x_k)$$

- But Forward-Flow is inconsistent, even for f, g quadratic

Convergence rate of Forward-Backward algorithm

$$\min_{x \in \mathbb{R}^n} f(x) + g(x)$$

Lemma: Assume:

1. $F = f + g$ is α-gradient dominated.
2. $-LI \preceq \nabla^2 f(x) \preceq LI$ for some $L > 0$, and g is convex.

Then Forward-Backward algorithm with step size $\eta \leq \frac{1}{L}$ satisfies

$$F(x_k) - F(x^*) \leq (1 - \alpha\eta)^k \left(F(x_0) - F(x^*)\right)$$

[Wibisono, *Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem*, COLT 2018, Lemma 8]

Forward-Backward algorithm for relative entropy

Relative entropy

Relative entropy with respect to $\nu = e^{-f}$:

$$H_\nu(\rho) = F(\rho) - H(\rho)$$

where

1. $F(\rho) = \mathbb{E}_\rho[f]$ is potential energy
   - F satisfies $-LI \preceq \mathrm{Hess}\, F(\rho) \preceq LI \Leftrightarrow -LI \preceq \nabla^2 f(x) \preceq LI$
   - Forward method implemented by gradient descent of f:

$$\text{Forward}_F(\rho) = (I - \eta \nabla f)_{\#}\rho$$

2. $-H(\rho) = \mathbb{E}_\rho[\log \rho]$ is negative entropy
   - Convex in W_2 metric
   - Backward method is not (?) implementable in general
   - Flow method implemented by heat equation/Brownian motion

also, $H_\nu = F - H$ is α-gradient dominated $\Leftrightarrow$ ν satisfies α-LSI

Forward-Backward for relative entropy

$$\min_{\rho \in \mathcal{P}(\mathbb{R}^n)} H_\nu(\rho) = F(\rho) - H(\rho)$$

Forward-Backward algorithm:

$$\rho_{k+\frac{1}{2}} = (I - \eta \nabla f)_{\#}\rho_k \iff x_{k+\frac{1}{2}} = x_k - \eta \nabla f(x_k)$$

$$\rho_{k+1} = \arg \min_{\rho \in \mathcal{P}(\mathbb{R}^n)} \left\{ -H(\rho) + \frac{1}{2\eta} W_2(\rho, \rho_{k+\frac{1}{2}})^2 \right\}$$

- Not (?) implementable in general, but can do in Gaussian case
- Should be consistent, and converge to ν

Convergence of Forward-Backward for relative entropy

Theorem: Assume f is α-strongly convex and L-smooth:

$$\alpha I \preceq \nabla^2 f(x) \preceq LI$$

Then the Forward-Backward algorithm with $\eta < \frac{1}{L}$ has

$$W_2(\rho_k, \nu)^2 \leq (1 - \alpha\eta)^k \, W_2(\rho_0, \nu)^2.$$

- Matches exponential rate of gradient flow in continuous time
- Uses convexity of $-H$ along generalized geodesics

[Salim, Korba, & Luise, *The Wasserstein Proximal Gradient Algorithm*, NeurIPS 2020, Corollary 11]

Forward-Backward for relative entropy: Gaussian case

Let $\nu = \mathcal{N}(0, \Sigma)$. Start from $\rho_0 = \mathcal{N}(0, \Sigma_0)$, so $\rho_k = \mathcal{N}(0, \Sigma_k)$.

To minimize $H_\nu(\rho) = F(\rho) - H(\rho)$, can use: (all consistent: $\Sigma_k \to \Sigma$)

1. Gradient descent (Forward$_{F-H}$):

$$\Sigma_{k+1} = \Sigma_k \left(I + \eta(\Sigma_k^{-1} - \Sigma^{-1})\right)^2$$

2. Proximal method (Backward$_{F-H}$):

$$\Sigma_{k+1} \left(I - \eta(\Sigma_{k+1}^{-1} - \Sigma^{-1})\right)^2 = \Sigma_k$$

Forward-Backward for relative entropy: Gaussian case

Let $\nu = \mathcal{N}(0, \Sigma)$. Start from $\rho_0 = \mathcal{N}(0, \Sigma_0)$, so $\rho_k = \mathcal{N}(0, \Sigma_k)$.

To minimize $H_\nu(\rho) = F(\rho) - H(\rho)$, can use: (all consistent: $\Sigma_k \to \Sigma$)

1. Gradient descent (Forward$_{F-H}$):

$$\Sigma_{k+1} = \Sigma_k \left(I + \eta(\Sigma_k^{-1} - \Sigma^{-1})\right)^2$$

2. Proximal method (Backward$_{F-H}$):

$$\Sigma_{k+1} \left(I - \eta(\Sigma_{k+1}^{-1} - \Sigma^{-1})\right)^2 = \Sigma_k$$

3. Forward-Backward (Backward$_{-H} \circ$ Forward$_F$):

$$\Sigma_{k+1} \left(I - \eta\Sigma_{k+1}^{-1}\right)^2 = \Sigma_k \left(I - \eta\Sigma^{-1}\right)^2$$

Forward-Backward for relative entropy: Gaussian case

Let $\nu = \mathcal{N}(0, \Sigma)$. Start from $\rho_0 = \mathcal{N}(0, \Sigma_0)$, so $\rho_k = \mathcal{N}(0, \Sigma_k)$.

To minimize $H_\nu(\rho) = F(\rho) - H(\rho)$, can use: (all consistent: $\Sigma_k \to \Sigma$)

1. Gradient descent (Forward$_{F-H}$):

$$\Sigma_{k+1} = \Sigma_k \left(I + \eta(\Sigma_k^{-1} - \Sigma^{-1})\right)^2$$

2. Proximal method (Backward$_{F-H}$):

$$\Sigma_{k+1} \left(I - \eta(\Sigma_{k+1}^{-1} - \Sigma^{-1})\right)^2 = \Sigma_k$$

3. Forward-Backward (Backward$_{-H}$ $\circ$ Forward$_F$):

$$\Sigma_{k+1} \left(I - \eta\Sigma_{k+1}^{-1}\right)^2 = \Sigma_k \left(I - \eta\Sigma^{-1}\right)^2$$

4. Backward-Forward (Forward$_{-H}$ $\circ$ Backward$_F$):

$$\Sigma_{k+1} \left(I + \eta\Sigma^{-1}\right)^2 = \Sigma_k \left(I + \eta\Sigma_k^{-1}\right)^2$$

Convergence analysis of ULA

ULA

Approaches to handle bias of **ULA**:

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, Z_k$$

1. Remove bias of **ULA** via Metropolis-Hastings
    - **MALA**: Metropolis-Adjusted Langevin Algorithm
2. Analyze convergence rate to biased limit
    - Choose small step size to make bias small

1. MALA

MALA

Recall **MALA** = **ULA** + Metropolis-Hastings

To sample from $\nu \propto e^{-f}$ on $\mathbb{R}^n$:

1. From x_k, let
$$y_k = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, z_k$$
where $z_k \sim \mathcal{N}(0, I)$ is independent, $\eta > 0$ is step size

2. Set
$$x_{k+1} = \begin{cases} y_k & \text{with prob } \; a_{x_k}(y_k) = \min\left\{1, \dfrac{\nu(y_k) P_{y_k}(x_k)}{\nu(x_k) P_{x_k}(y_k)}\right\} \\ x_k & \text{with prob } \; 1 - a_{x_k}(y_k) \end{cases}$$
where $P_x(y) = \frac{1}{(4\pi\eta)^{n/2}} \exp\left(-\frac{\|y - x + \eta \nabla f(x)\|^2}{4\eta}\right)$

Convergence rate of MALA

Theorem 1: Assume f is α-strongly convex and L-smooth:

$$\alpha I \preceq \nabla^2 f(x) \preceq LI$$

Let $\rho_0 = \mathcal{N}(x^*, \frac{1}{L} I)$, which is warm with $M_\nu(\rho_0) = \kappa^{n/2}$ where $\kappa = \frac{L}{\alpha}$ is condition number. Assume $\kappa \ll n$. Then **MALA** with $\eta = \Theta\left(\frac{1}{nL}\right)$ has mixing time in TV distance:

$$\tau(\epsilon) = O\left(n^2 \kappa \log\left(\frac{\kappa}{\epsilon^{1/n}}\right)\right) = \tilde{O}(n^2\kappa).$$

- Conductance analysis using *isoperimetry* ($\Leftrightarrow$ Poincaré inequality)

[Dwivedi, Chen, Wainwright, and Yu, *Log-Concave Sampling: Metropolis-Hastings Algorithms are Fast*, Journal of Machine Learning Research, 2019]

Improved convergence rate of MALA

Theorem 2: Assume f is α-strongly convex and L-smooth:

$$\alpha I \preceq \nabla^2 f(x) \preceq LI$$

Let $\rho_0 = \mathcal{N}(x^*, \frac{1}{L} I)$, which is warm with $M_\nu(\rho_0) = \kappa^{n/2}$ where $\kappa = \frac{L}{\alpha}$ is condition number. Assume $\kappa \ll n$. Then **MALA** with $\eta = \Theta\left(\frac{1}{nL}\right)$ has mixing time in TV distance:

$$\tau(\epsilon) = O\left(n\kappa \log\left(\frac{\kappa}{\epsilon}\right)\right) = \tilde{O}(n\kappa).$$

- Analysis using *log-isoperimetry* ($\Leftrightarrow$ log-Sobolev inequality)
- Conductance profile

[Chen, Dwivedi, Wainwright, Yu, *Fast mixing of Metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients*, JMLR, 2020]

2. Biased convergence of ULA

Biased convergence of ULA

Analyze convergence of **ULA** to biased limit

Choose step size small to make bias small, and derive mixing time

- Dalalyan, *Theoretical guarantees for approximate sampling from a smooth and log-concave density*, Journal of the Royal Statistical Society: Series B, 2017
- Dalalyan, *Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent*, COLT 2017
- Durmus & Moulines, *Non-asymptotic convergence analysis for the Unadjusted Langevin Algorithm*, Annals of Applied Probability, 2017
- Durmus, Majewski, & Miasojedow, *Analysis of Langevin Monte Carlo via Convex Optimization*, JMLR, 2019
- Cheng & Bartlett, *Convergence of Langevin MCMC in KL-divergence*, ALT 2018
- Vempala & Wibisono, *Rapid Convergence of the Unadjusted Langevin Algorithm: Isoperimetry Suffices*, NeurIPS 2019

ULA analysis 1: Coupling

Convergence of ULA under SLC

Theorem: Assume f is α-strongly convex and L-smooth:

$$\alpha I \preceq \nabla^2 f(x) \preceq LI$$

Then **ULA** for $\nu \propto e^{-f}$ on $\mathbb{R}^n$ with step size $\eta \le \frac{2}{\alpha+L}$ satisfies:

$$W_2(\rho_k, \nu) \le \underbrace{(1-\alpha\eta)^k W_2(\rho_0, \nu)}_{\le \frac{\sqrt{\varepsilon}}{2}} + \underbrace{\sqrt{2\eta\kappa^2 n}}_{\le \frac{\sqrt{\varepsilon}}{2}}$$

[Dalalyan, *Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent*, COLT 2017, Theorem 1]

Mixing time of ULA under SLC

Corollary: To reach $W_2(\rho_k, \nu)^2 \le \epsilon$, can set

$$\eta = \frac{\epsilon}{8\kappa^2 n}$$

and suffices to run **ULA** for the number of iterations:

$$k \ge \frac{1}{\alpha\eta} \log \frac{2W_2(\rho_0, \nu)}{\sqrt{\epsilon}} = \frac{8\kappa^2 n}{\alpha\epsilon} \log \frac{2W_2(\rho_0, \nu)}{\sqrt{\epsilon}}$$

Therefore, mixing time of **ULA** in W_2 distance is

$$\tau(\epsilon) = \tilde{O}\left(\frac{\kappa^2 n}{\alpha\epsilon}\right)$$

Proof of Theorem

<u>Proof of Theorem:</u> Let $x_0 \sim \rho_0$ and $y_0 \sim \nu$ with the optimal coupling, so

$$W_2(\rho_0, \nu)^2 = \mathbb{E}[\|x_0 - y_0\|^2]$$

Evolve $x_k \sim \rho_k$ along **ULA** and $y_k \sim \nu$ along the Langevin dynamics, coupled with the same Brownian motion dW_t:

1. **ULA**:

   $$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta} Z_k$$

   is solution $x_{k+1} = X_\eta$ of the SDE

   $$dX_t = -\nabla f(X_0)\, dt + \sqrt{2} dW_t$$

   starting from $X_0 = x_k$.

   $$X_t = X_0 - t \nabla f(X_0) + \sqrt{2} W_t$$

Proof of Theorem

Proof of Theorem: Let $x_0 \sim \rho_0$ and $y_0 \sim \nu$ with the optimal coupling, so

$$W_2(\rho_0, \nu)^2 = \mathbb{E}[\|x_0 - y_0\|^2]$$

Evolve $x_k \sim \rho_k$ along **ULA** and $y_k \sim \nu$ along the Langevin dynamics, coupled with the same Brownian motion dW_t:

1. **ULA**:

   $$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta} Z_k$$

   is solution $x_{k+1} = X_\eta$ of the SDE

   $$dX_t = -\nabla f(X_0)\, dt + \sqrt{2} dW_t$$

   starting from $X_0 = x_k$.

2. Langevin dynamics: $y_{k+1} = Y_\eta$ is solution to the SDE

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

   starting from $Y_0 = y_k$. Note $Y_t \sim \nu$ since ν is stationary.

Using strong convexity and smoothness of f, can show

$$\sqrt{\mathbb{E}[\|x_{k+1} - y_{k+1}\|^2]} \leq (1 - \alpha\eta)\sqrt{\mathbb{E}[\|x_k - y_k\|^2]} + \sqrt{2\eta^3 L^2 n}$$

Unrolling the recursion gives

$$\begin{aligned} W_2(\rho_k, \nu) &\leq \sqrt{\mathbb{E}[\|x_k - y_k\|^2]} \\ &\leq (1 - \alpha\eta)^k \sqrt{\mathbb{E}[\|x_0 - y_0\|^2]} + \frac{\sqrt{2\eta^3 L^2 n}}{\alpha\eta} \\ &= (1 - \alpha\eta)^k W_2(\rho_0, \nu) + \sqrt{2\eta\kappa^2 n} \end{aligned}$$

$\square$

ULA analysis 2: Convex optimization

Improved convergence of ULA under SLC

Theorem: Assume f is α-strongly convex and L-smooth:

$$\alpha I \preceq \nabla^2 f(x) \preceq LI$$

Then **ULA** for $\nu \propto e^{-f}$ on $\mathbb{R}^n$ with step size $\eta \leq \frac{1}{L}$ satisfies:

$$W_2(\rho_k, \nu)^2 \leq (1 - \alpha\eta)^k W_2(\rho_0, \nu)^2 + 2\eta\kappa n$$

[Durmus, Majewski, & Miasojedow, *Analysis of Langevin Monte Carlo via Convex Optimization*, JMLR, 2019, Theorem 9]

Improved mixing time of ULA under SLC

Corollary: To reach $W_2(\rho_k, \nu)^2 \le \epsilon$, can set

$$\eta = \frac{\epsilon}{4\kappa n}$$

and suffices to run **ULA** for the number of iterations:

$$k \ge \frac{1}{\alpha\eta} \log \frac{2W_2(\rho_0, \nu)^2}{\epsilon} = \frac{4\kappa n}{\alpha\epsilon} \log \frac{2W_2(\rho_0, \nu)^2}{\epsilon}$$

Therefore, mixing time of **ULA** in W_2 distance is

$$\tau(\epsilon) = \tilde{O}\left(\frac{\kappa n}{\alpha\epsilon}\right)$$

- Note better dependence on κ compared to previous bound

Proof of Theorem

Proof of Theorem: Suffices to prove recursion:

$$W_2(\rho_{k+1},\nu)^2 \le (1-\alpha\eta)W_2(\rho_k,\nu)^2 + 2\eta^2 Ln$$

In fact will show:

$$0 \le 2\eta H_\nu(\rho_{k+1}) \le (1-\alpha\eta)W_2(\rho_k,\nu)^2 - W_2(\rho_{k+1},\nu)^2 + 2\eta^2 Ln$$

Proof of Theorem

Proof of Theorem: Suffices to prove recursion:

$$W_2(\rho_{k+1}, \nu)^2 \le (1 - \alpha\eta) W_2(\rho_k, \nu)^2 + 2\eta^2 L n$$

In fact will show:

$$2\eta H_\nu(\rho_{k+1}) \le (1 - \alpha\eta) W_2(\rho_k, \nu)^2 - W_2(\rho_{k+1}, \nu)^2 + 2\eta^2 L n$$

Note: Analogous to recursion for inexact gradient algorithm for optimization $\min_{x \in \mathbb{R}^n} f(x)$:

$$2\eta(f(x_{k+1}) - f(x^*)) \le \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + C\eta^2$$

[Beck & Teboulle, *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM Journal on Imaging Sciences, 2009]

Recall decomposition for $\nu \propto e^{-f}$:

$$H_\nu(\rho) = F(\rho) - H(\rho)$$

where $F(\rho) = \mathbb{E}_\rho[f] + \log \int_{\mathbb{R}^n} e^{-f(x)}\, dx$ and $-H(\rho) = \mathbb{E}_\rho[\log \rho]$.

Note $F(\nu) = H(\nu)$ since $H_\nu(\nu) = F(\nu) - H(\nu) = 0$.

Then can write:

$$\begin{aligned} H_\nu(\rho_{k+1}) &= F(\rho_{k+1}) - H(\rho_{k+1}) \\ &= \underbrace{F(\rho_{k+1}) - F(\rho_{k+\frac{1}{2}})}_{(a)} + \underbrace{F(\rho_{k+\frac{1}{2}}) - F(\nu)}_{(b)} + \underbrace{H(\nu) - H(\rho_{k+1})}_{(c)} \end{aligned}$$

where

$$\begin{aligned} \rho_{k+\frac{1}{2}} &= (I - \eta \nabla f)_{\#}\rho_k = \mathrm{Forward}_F(\rho_k) \\ \rho_{k+1} &= \rho_{k+\frac{1}{2}} * \mathcal{N}(0, 2\eta I) = \mathrm{Flow}_{-H}(\rho_{k+\frac{1}{2}}) \end{aligned}$$

Show:

(a) f L-smooth $\Rightarrow$

$$F(\rho_{k+1}) - F(\rho_{k+\frac{1}{2}}) \le \eta L n$$

(b) f α-strongly convex, L-smooth $\Rightarrow$

$$F(\rho_{k+\frac{1}{2}}) - F(\nu) \ \le \ \frac{(1-\alpha\eta)}{2\eta} W_2(\rho_k, \nu)^2 - \frac{1}{2\eta} W_2(\rho_{k+\frac{1}{2}}, \nu)^2$$

(c) $-H$ convex in W_2 $\Rightarrow$

$$H(\nu) - H(\rho_{k+1}) \le \frac{1}{2\eta} W_2(\rho_{k+\frac{1}{2}}, \nu)^2 - \frac{1}{2\eta} W_2(\rho_{k+1}, \nu)^2$$

Summing gives the desired relation. □

[Durmus, Majewski, & Miasojedow, *Analysis of Langevin Monte Carlo via Convex Optimization*, JMLR, 2019, Lemma 3, 4, 5]

ULA analysis 3: LSI

Convergence of ULA under LSI

Theorem: Assume $\nu \propto e^{-f}$ satisfies α-LSI and f is L-smooth ($-LI \preceq \nabla^2 f(x) \preceq LI$). Then ULA with step size $\eta \leq \frac{\alpha}{4L^2}$ satisfies

$$H_\nu(\rho_k) \leq e^{-\alpha\eta k} H_\nu(\rho_0) + \eta\kappa L n$$

[Vempala & Wibisono, *Rapid Convergence of the Unadjusted Langevin Algorithm: Isoperimetry Suffices*, NeurIPS 2019, Theorem 1]

Mixing time of ULA under LSI

Corollary: Let $\rho_0 = \mathcal{N}(x^*, \frac{1}{L}I)$ for some $\nabla f(x^*) = 0$, so $H_\nu(\rho_0) \le O(n)$. To reach $H_\nu(\rho_0) \le \epsilon$, can set

$$\eta = \frac{\epsilon}{2\kappa L n}$$

and suffices to run **ULA** for the number of iterations

$$k \ge \frac{1}{\alpha\eta} \log \frac{2H_\nu(\rho_0)}{\epsilon} = \frac{2\kappa^2 n}{\epsilon} \log \frac{2H_\nu(\rho_0)}{\epsilon}$$

Thus, the mixing time of **ULA** in relative entropy under LSI is

$$\tau(\epsilon) = \tilde{O}\left(\frac{\kappa^2 n}{\epsilon}\right)$$

Mixing time of ULA under LSI

Corollary: Let $\rho_0 = \mathcal{N}(x^*, \frac{1}{L} I)$ for some $\nabla f(x^*) = 0$, so $H_\nu(\rho_0) \leq O(n)$. To reach $H_\nu(\rho_0) \leq \epsilon$, can set

$$\eta = \frac{\epsilon}{2\kappa L n}$$

and suffices to run **ULA** for the number of iterations

$$k \geq \frac{1}{\alpha\eta} \log \frac{2H_\nu(\rho_0)}{\epsilon} = \frac{2\kappa^2 n}{\epsilon} \log \frac{2H_\nu(\rho_0)}{\epsilon}$$

Thus, the mixing time of **ULA** in relative entropy under LSI is

$$\tau(\epsilon) = \tilde{O}\left(\frac{\kappa^2 n}{\epsilon}\right)$$

- Since $H_\nu(\rho) \geq \frac{\alpha}{2} W_2(\rho, \nu)^2$, implies mixing time to $W_2(\rho, \nu)^2 \leq \epsilon'$ is

$$\tau_{W_2^2}(\epsilon') = \tilde{O}\left(\frac{\kappa^2 n}{\alpha\epsilon'}\right) = \tilde{O}\left(\frac{\kappa^2 n}{\alpha\epsilon'}\right)$$

$\varepsilon = \varepsilon' \alpha$

Proof via PDE interpolation

Proof of Theorem: Suffices to prove recursion in each iteration:

$$H_\nu(\rho_{k+1}) \le e^{-\alpha\eta} H_\nu(\rho_k) + 6\eta^2 n L^2$$

Write **ULA**:

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta} Z_k$$

as solution $x_{k+1} = X_\eta$ of the SDE

$$dX_t = -\nabla f(X_0)\, dt + \sqrt{2} dW_t$$

starting from $X_0 = x_k$.

The density $X_t \sim \rho_t$ satisfies the modified Fokker-Planck equation:

$$\frac{\partial \rho_t(x)}{\partial t} = \nabla \cdot \big(\rho_t(x)\, \mathbb{E}_{\rho_{0|t}}[\nabla f(X_0) \mid X_t = x]\big) + \Delta \rho_t(x)$$

where $\rho_{0|t}(\cdot \mid x)$ is the conditional density $X_0 \mid \{X_t = x\}$.

Write as original Fokker-Planck equation plus error:

$$\frac{\partial \rho_t(x)}{\partial t} = \nabla \cdot \left(\rho_t(x) \nabla \log \frac{\rho_t}{\nu}(x)\right) \quad \leftarrow \nabla \cdot (\rho_t \nabla f(x)) + \Delta \rho_t(x)$$

$$+ \nabla \cdot \big(\rho_t(x)\, \mathbb{E}_{\rho_{0|t}}[\nabla f(X_0) - \nabla f(x) \mid X_t = x]\big)$$

The change of relative entropy is

$$\frac{d}{dt} H_\nu(\rho_t) = -J_\nu(\rho_t) + \mathbb{E}_{\rho_{0t}} \left[\left\langle \nabla f(X_t) - \nabla f(X_0), \nabla \log \frac{\rho_t}{\nu}(X_t) \right\rangle \right]$$

Using ∇f is L-Lipschitz, LSI, and Talagrand inequality, can bound:

$$\frac{d}{dt} H_\nu(\rho_t) \leq -\frac{3}{2} \alpha H_\nu(\rho_t) + \frac{4\eta^2 L^4}{\alpha} H_\nu(\rho_0) + 3\eta n L^2$$

Integrating gives desired recursion. $\square$

Rényi divergence along ULA

Rényi divergence along Langevin dynamics

Rényi divergence: $R_{q,\nu}(\rho) = \frac{1}{q-1} \log \mathbb{E}_\nu \left[\left(\frac{\rho}{\nu} \right)^q \right]$

Theorem: Assume $\nu \propto e^{-f}$ satisfies α-LSI. Then along the Langevin dynamics

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

Rényi divergence of order $q \geq 1$ converges exponentially fast:

$$R_{q,\nu}(\rho_t) \leq e^{-\frac{2\alpha t}{q}} R_{q,\nu}(\rho_0)$$

Rényi divergence along ULA

$\nu \propto e^{-f}$

Let ν_η denote the biased limiting distribution of **ULA**

Theorem: Assume ν_η satisfies β-LSI, and f is L-smooth $(-LI \preceq \nabla^2 f(x) \preceq LI)$. Along **ULA** with $\eta \le \frac{1}{L}$, for $q > 1$:

$$R_{q,\nu}(\rho_k) \le \left(\frac{q - \frac{1}{2}}{q-1} \right) R_{2q,\nu_\eta}(\rho_0) e^{-\frac{\beta \eta k}{2q}} + R_{2q-1,\nu}(\nu_\eta)$$

[Vempala & Wibisono, *Rapid Convergence of the Unadjusted Langevin Algorithm: Isoperimetry Suffices*, NeurIPS 2019, Theorem 4]

- E.g. $\nu = \mathcal{N}(0, \frac{1}{\alpha} I) \Rightarrow \nu_\eta = \mathcal{N}(0, \frac{1}{\alpha(1-\frac{\alpha\eta}{2})} I)$ satisfies β-LSI with $\beta = \alpha(1 - \frac{\alpha\eta}{2}) \gtrsim \frac{\alpha}{2}$
Bias is $R_{q,\nu}(\nu_\eta) = O(\eta^2)$ for $1 < q < \frac{2}{\alpha\eta}$
- Can also prove convergence of Rényi divergence along **ULA** under Poincaré inequality

Variants of ULA

Variants of ULA

Many discretization of the Langevin dynamics

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

- Stochastic gradient
- Proximal Langevin algorithm
- Splitting method
- Ozaki discretization
- Projection
- ...

1. ULA with stochastic gradient

ULA with stochastic gradient: $\nu \propto e^{-f}$

$$x_{k+1} = x_k - \eta g(x_k) + \sqrt{2\eta}\, Z_k$$

where $Z_k \sim \mathcal{N}(0, I)$, and $g(x_k)$ is an estimator of $\nabla f(x_k)$

- E.g. $f(x) = \mathbb{E}_\theta[F(x;\theta)]$ and $g(x) = \nabla F(x;\theta)$ for some random θ

  $f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$, $g(x) = \nabla f_I(x)$, $I \sim \text{Uniform}\{1, \dots, m\}$

  - Welling & Teh, *Bayesian Learning via Stochastic Gradient Langevin Dynamics*, ICML 2011
  - Dalalyan & Karagulyan, *User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient*, Stochastic Processes and their Applications, 2017

2. Proximal Langevin Algorithm

Use proximal method for f instead of gradient descent:

$$
\begin{aligned}
& x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2\eta} \|x - (x_k + \sqrt{2\eta}\, Z_k)\|^2 \right\} \\
\Leftrightarrow \quad & x_{k+1} = x_k - \eta \nabla f(x_{k+1}) + \sqrt{2\eta}\, Z_k
\end{aligned}
$$

- Pereyra, *Proximal Markov chain Monte Carlo algorithms*, Statistics and Computing, 2016
- Bernton, *Langevin Monte Carlo and JKO splitting*, COLT 2018
- Wibisono, *Proximal Langevin Algorithm: Rapid Convergence Under Isoperimetry*, arXiv 2019

3. Splitting method

$\nu \propto \nu_1 \cdot \nu_2$

For sampling from composite distribution $\nu \propto e^{-(f+g)}$

- e.g. Bayesian posterior $\propto$ prior $\times$ likelihood
- **ULA** (gradient descent) for f and proximal method for g:

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{1}{2\eta} \| x - (x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, Z_k) \|^2 \right\}$$

- Durmus, Moulines, & Pereyra, *Efficient Bayesian computation by proximal Markov Chain Monte Carlo: When Langevin meets Moreau*, SIAM Journal on Imaging Sciences, 2018
- Salim, Kovalev, & Richtárik, *Stochastic Proximal Langevin Algorithm: Potential Splitting and Nonasymptotic Rates*, NeurIPS 2019

4. Ozaki discretization

Use Hessian information to help discretize the Langevin dynamics

$$x_{k+1} = x_k - (I - e^{-\eta H_k}) H_k^{-1} \nabla f(x_k) + \sqrt{(I - e^{-2\eta H_k}) H_k^{-1}}\, Z_k$$

where $H_k = \nabla^2 f(x_k)$ and $Z_k \sim \mathcal{N}(0, I)$ is independent

- Ozaki, *A bridge between nonlinear time series models and nonlinear stochastic dynamical systems: A local linearization approach*, Statistica Sinica, 1992
- Dalalyan, *Theoretical guarantees for approximate sampling from a smooth and log-concave density*, Journal of the Royal Statistical Society: Series B, 2017

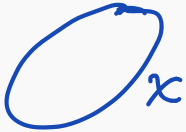
$$I - e^{-\eta H_k} \approx I - (I - \eta H_k) = \eta H_k$$

$$(I - e^{-\eta H_k}) H_k^{-1} \approx \eta$$

5. ULA with projection

For sampling from a distribution $\nu \propto e^{-f}$ with compact support $\mathcal{X} \subseteq \mathbb{R}^n$

$$x_{k+1} = \Pi_{\mathcal{X}}(x_k - \eta \nabla f(x_k) + \sqrt{2\eta} Z_k)$$



where $\Pi_{\mathcal{X}}$ is projection to $\mathcal{X}$, and $Z_k \sim \mathcal{N}(0, I)$ is independent

- Bubeck, Eldan, & Lehec, *Sampling from a log-concave distribution with Projected Langevin Monte Carlo*, NeurIPS 2015
- Brosse, Durmus, Moulines, & Pereyra, *Sampling from a log-concave distribution with compact support with proximal Langevin Monte Carlo*, COLT 2017