On Gradient-Based Optimization: Accelerated, Nonconvex and Stochastic

Michael I. Jordan
University of California, Berkeley

April 16, 2018

Statistics and Computation

- A Grand Challenge of our era: tradeoffs between statistical inference and computation
  - most data analysis problems have a time budget
  - and they're often embedded in a control problem

Statistics and Computation

- A Grand Challenge of our era: tradeoffs between statistical inference and computation
  - most data analysis problems have a time budget
  - and they're often embedded in a control problem
- Optimization has provided the computational model for this effort (computer science, not so much)
  - it's provided the algorithms and the insights

Statistics and Computation

- A Grand Challenge of our era: tradeoffs between statistical inference and computation
  - most data analysis problems have a time budget
  - and they’re often embedded in a control problem
- Optimization has provided the computational model for this effort (computer science, not so much)
  - it’s provided the algorithms and the insights
- Statistics has quite a few good lower bounds
  - which have delivered fundamental understanding
  - placing them in contact with computational lower bounds will deliver further fundamental understanding

Statistics and Computation (cont)

- Modern large-scale statistics has posed new challenges for optimization
  - millions of variables, millions of terms, sampling issues, nonconvexity, need for confidence intervals, parallel—distributed platforms, etc

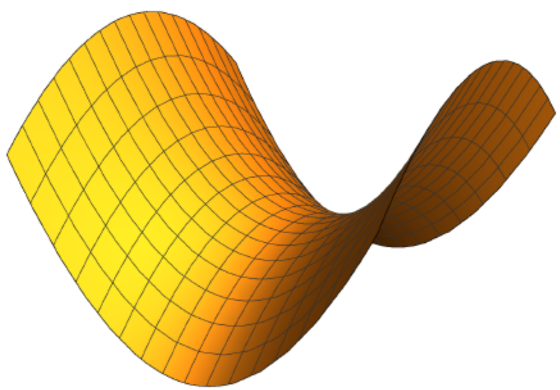
Statistics and Computation (cont)

- Modern large-scale statistics has posed new challenges for optimization
  - millions of variables, millions of terms, sampling issues, nonconvexity, need for confidence intervals, parallel—distributed platforms, etc
- Current focus: what can we do with the following ingredients?
  - gradients
  - stochastics
  - acceleration

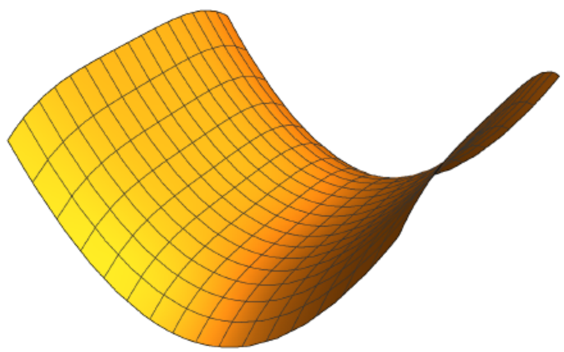
Nonconvex Optimization in Machine Learning

- Bad local minima used to be thought of as the main problem on the optimization side of machine learning
- But many machine learning architectures either have no local minima (see list later), or stochastic gradient seems to have no trouble (eventually) finding global optima
- But saddle points abound in these architectures, and they cause the learning curve to flatten out, perhaps (nearly) indefinitely

The Importance of Saddle Points



Strict saddle point



Non-strict saddle point

- How to escape?
  - need to have a negative eigenvalue that's strictly negative
- How to escape efficiently?
  - in high dimensions how do we find the direction of escape?
  - should we expect exponential complexity in dimension?

Part I: How to Escape Saddle Points Efficiently

with Chi Jin, Rong Ge, Sham Kakade, and Praneeth Netrapalli

A Few Facts

- Gradient descent will asymptotically avoid saddle points (Lee, Simchowitz, Jordan & Recht, 2017)
- Gradient descent can take exponential time to escape saddle points (Du, Jin, Lee, Jordan, & Singh, 2017)
- Stochastic gradient descent can escape saddle points in polynomial time (Ge, Huang, Jin & Yuan, 2015)
  - but that’s still not an explanation for its practical success
- Can we prove a stronger theorem?

Optimization

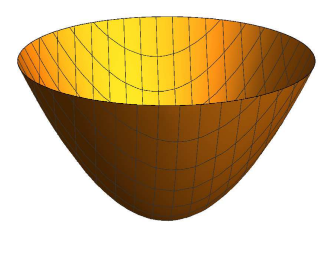
Consider problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \; f(\mathbf{x})$$

Gradient Descent (GD):

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t).$$

Convex: converges to global minimum; **dimension-free** iterations.



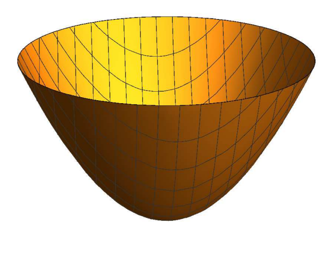
Consider problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \ f(\mathbf{x})$$

Gradient Descent (GD):

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t).$$

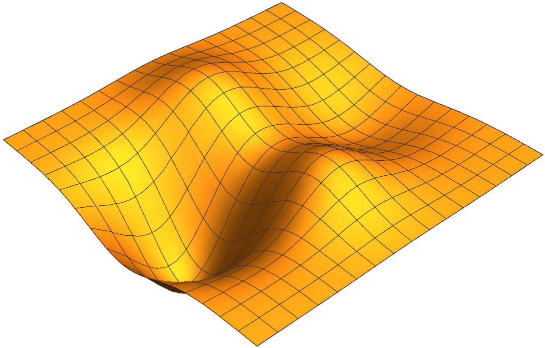
Convex: converges to global minimum; **dimension-free** iterations.



Nonconvex Optimization

Non-convex: converges to Stationary Point (SP) $\nabla f(\mathbf{x}) = 0$.

SP : local min / local max / saddle points



Many applications: no spurious local min (see full list later).

Some Well-Behaved Nonconvex Problems

- PCA, CCA, Matrix Factorization
- Orthogonal Tensor Decomposition (Ge, Huang, Jin, Yang, 2015)
- Complete Dictionary Learning (Sun et al, 2015)
- Phase Retrieval (Sun et al, 2015)
- Matrix Sensing (Bhojanapalli et al, 2016; Park et al, 2016)
- Symmetric Matrix Completion (Ge et al, 2016)
- Matrix Sensing/Completion, Robust PCA (Ge, Jin, Zheng, 2017)

- The problems have no spurious local minima and all saddle points are strict

Convergence to FOSP

Function $f(\cdot)$ is ℓ-**smooth (or gradient Lipschitz)**

$$\forall \mathbf{x}_1, \mathbf{x}_2, \ \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le \ell \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point $\mathbf{x}$ is an ϵ-**first-order stationary point** (ϵ-FOSP) if

$$\|\nabla f(\mathbf{x})\| \le \epsilon$$

GD Converges to FOSP (Nesterov, 1998)

For ℓ-smooth function, GD with $\eta = 1/\ell$ finds ϵ-FOSP in iterations:

$$\frac{2\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2}$$

*Number of iterations is dimension free.

Convergence to FOSP

Function $f(\cdot)$ is ℓ-**smooth (or gradient Lipschitz)**

$$\forall \mathbf{x}_1, \mathbf{x}_2,\ \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le \ell \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point $\mathbf{x}$ is an ϵ-**first-order stationary point** (ϵ-FOSP) if

$$\|\nabla f(\mathbf{x})\| \le \epsilon$$

GD Converges to FOSP (Nesterov, 1998)

For ℓ-smooth function, GD with $\eta = 1/\ell$ finds ϵ-FOSP in iterations:

$$\frac{2\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2}$$

*Number of iterations is dimension free.

Definitions and Algorithm

Function $f(\cdot)$ is ρ-**Hessian Lipschitz** if

$$\forall \mathbf{x}_1, \mathbf{x}_2, \ \|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \le \rho \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point $\mathbf{x}$ is an ϵ-**second-order stationary point** (ϵ-SOSP) if

$$\|\nabla f(\mathbf{x})\| \le \epsilon, \qquad \text{and} \qquad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \ge -\sqrt{\rho\epsilon}$$

Perturbed Gradient Descent (PGD)

1. **for** $t = 0, 1, \ldots$ **do**
2. **if** perturbation condition holds **then**
3. $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
4. $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$

Only adds perturbation when $\|\nabla f(\mathbf{x}_t)\| \le \epsilon$; no more than once per T steps.

Definitions and Algorithm

Function $f(\cdot)$ is ρ-**Hessian Lipschitz** if

$$\forall \mathbf{x}_1, \mathbf{x}_2,\ \|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \leq \rho \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

Point $\mathbf{x}$ is an ϵ-**second-order stationary point** (ϵ-SOSP) if

$$\|\nabla f(\mathbf{x})\| \leq \epsilon, \qquad \text{and} \qquad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}$$

Perturbed Gradient Descent (PGD)

1. **for** $t = 0, 1, \ldots$ **do**
2. **if** perturbation condition holds **then**
3. $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
4. $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$

Only adds perturbation when $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$; no more than once per T steps.

Main Result

PGD Converges to SOSP (This Work)

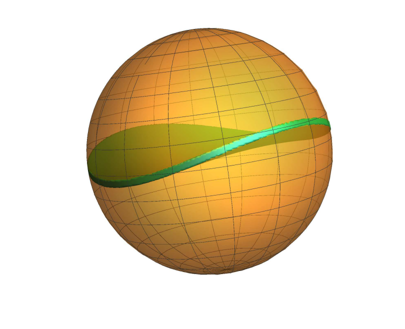
For ℓ-smooth and ρ-Hessian Lipschitz function f, PGD with $\eta = O(1/\ell)$ and proper choice of r, T w.h.p. finds ϵ-SOSP in iterations:

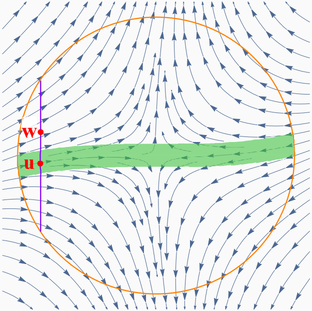
$$\tilde{O}\left(\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2}\right)$$

*Dimension dependence in iteration is $\log^4(d)$ (almost dimension free).

	GD(Nesterov 1998)	**PGD**(This Work)
Assumptions	ℓ-grad-Lip	ℓ-grad-Lip + ρ-Hessian-Lip
Guarantees	ϵ-FOSP	ϵ-SOSP
Iterations	$2\ell(f(\mathbf{x}_0) - f^\star)/\epsilon^2$	$\tilde{O}(\ell(f(\mathbf{x}_0) - f^\star)/\epsilon^2)$

Challenge: non-constant Hessian + large step size $\eta = O(1/\ell)$.
Around saddle point, **stuck region** forms a non-flat "pancake" shape.

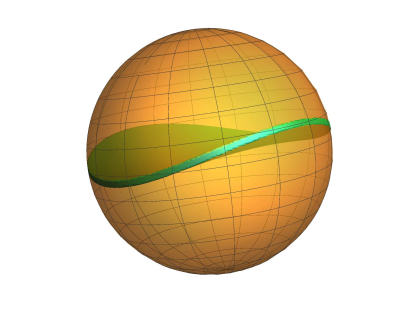


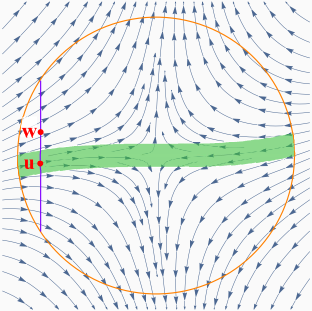


Key Observation: although we don't know its shape, we know it's thin!
(Based on an analysis of two nearly coupled sequences)

Geometry and Dynamics around Saddle Points

Challenge: non-constant Hessian + large step size $\eta = O(1/\ell)$.
Around saddle point, **stuck region** forms a non-flat "pancake" shape.





Key Observation: although we don't know its shape, we know it's thin!
(Based on an analysis of two nearly coupled sequences)

Next Questions

- Does acceleration help in escaping saddle points?
- What other kind of stochastic models can we use to escape saddle points?
- How do acceleration and stochastics interact?

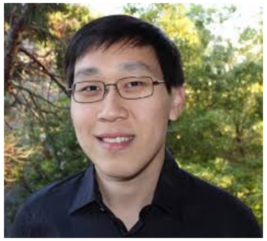
Next Questions

- Does acceleration help in escaping saddle points?
- What other kind of stochastic models can we use to escape saddle points?
- How do acceleration and stochastics interact?

- To address these questions we need to understand develop a deeper understanding of acceleration than has been available in the literature to date

Part I: Variational, Hamiltonian and Symplectic Perspectives on Acceleration

with Andre Wibisono, Ashia Wilson and Michael Betancourt



Interplay between Differentiation and Integration

- The 300-yr-old fields: Physics, Statistics
  - cf. Lagrange/Hamilton, Laplace expansions, saddlepoint expansions
- The numerical disciplines
  - e.g.,. finite elements, Monte Carlo

Interplay between Differentiation and Integration

- The 300-yr-old fields: Physics, Statistics
  - cf. Lagrange/Hamilton, Laplace expansions, saddlepoint expansions
- The numerical disciplines
  - e.g.,. finite elements, Monte Carlo
- Optimization?

Interplay between Differentiation and Integration

- The 300-yr-old fields: Physics, Statistics
  - cf. Lagrange/Hamilton, Laplace expansions, saddlepoint expansions
- The numerical disciplines
  - e.g.,. finite elements, Monte Carlo
- Optimization?
  - to date, almost entirely focused on differentiation

Accelerated gradient descent

Setting: Unconstrained convex optimization

$$\min_{x \in \mathbb{R}^d} f(x)$$

- Classical gradient descent:

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

  obtains a convergence rate of $O(1/k)$

- Accelerated gradient descent:

$$\begin{aligned} y_{k+1} &= x_k - \beta \nabla f(x_k) \\ x_{k+1} &= (1 - \lambda_k) y_{k+1} + \lambda_k y_k \end{aligned}$$

  obtains the (optimal) convergence rate of $O(1/k^2)$

Accelerated methods: Continuous time perspective

- Gradient descent is discretization of gradient flow

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

  (and mirror descent is discretization of natural gradient flow)

- Su, Boyd, Candes '14: Continuous time limit of accelerated gradient descent is a second-order ODE

$$\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0$$

- These ODEs are obtained by taking continuous time limits. Is there a deeper generative mechanism?

Our work: A general variational approach to acceleration

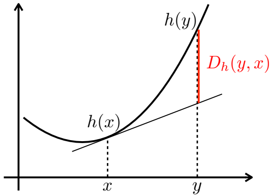
A systematic discretization methodology

Bregman Lagrangian

Define the **Bregman Lagrangian**:

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

- Function of position x, velocity $\dot{x}$, and time t
- $D_h(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$ is the Bregman divergence
- h is the convex distance-generating function
- f is the convex objective function

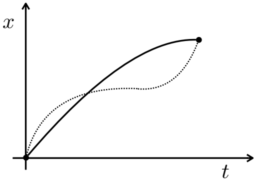


Bregman Lagrangian

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

Variational problem over curves:

$$\min_X \int \mathcal{L}(X_t, \dot{X}_t, t)\, dt$$



Optimal curve is characterized by **Euler-Lagrange** equation:

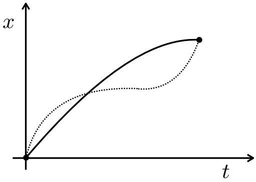
$$\frac{d}{dt}\left\{ \frac{\partial \mathcal{L}}{\partial \dot{x}}(X_t, \dot{X}_t, t) \right\} = \frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t)$$

Bregman Lagrangian

$$\mathcal{L}(x, \dot{x}, t) = e^{\gamma_t + \alpha_t} \left(D_h(x + e^{-\alpha_t}\dot{x}, x) - e^{\beta_t} f(x) \right)$$

Variational problem over curves:

$$\min_X \int \mathcal{L}(X_t, \dot{X}_t, t)\, dt$$



Optimal curve is characterized by **Euler-Lagrange** equation:

$$\frac{d}{dt}\left\{ \frac{\partial \mathcal{L}}{\partial \dot{x}}(X_t, \dot{X}_t, t) \right\} = \frac{\partial \mathcal{L}}{\partial x}(X_t, \dot{X}_t, t)$$

E-L equation for Bregman Lagrangian under ideal scaling:

$$\ddot{X}_t + (e^{\alpha_t} - \dot{\alpha}_t)\dot{X}_t + e^{2\alpha_t + \beta_t} \left[\nabla^2 h(X_t + e^{-\alpha_t}\dot{X}_t) \right]^{-1} \nabla f(X_t) = 0$$

General convergence rate

Theorem

Theorem Under ideal scaling, the E-L equation has convergence rate

$$f(X_t) - f(x^*) \le O(e^{-\beta_t})$$

Proof. Exhibit a Lyapunov function for the dynamics:

$$\mathcal{E}_t = D_h\left(x^*,\, X_t + e^{-\alpha_t}\dot{X}_t\right) + e^{\beta_t}(f(X_t) - f(x^*))$$

$$\dot{\mathcal{E}}_t = -e^{\alpha_t+\beta_t}D_f(x^*, X_t) + (\dot{\beta}_t - e^{\alpha_t})e^{\beta_t}(f(X_t) - f(x^*)) \;\le\; 0$$

$\square$

Note: Only requires convexity and differentiability of f, h

Mysteries

- Why can't we discretize the dynamics when we are using exponentially fast clocks?
- What happens when we arrive at a clock speed that we can discretize?
- How do we discretize once it's possible?

Mysteries

- Why can’t we discretize the dynamics when we are using exponentially fast clocks?
- What happens when we arrive at a clock speed that we can discretize?
- How do we discretize once it’s possible?

- The answers are to be found in symplectic integration

Symplectic Integration

- Consider discretizing a system of differential equations obtained from physical principles
- Solutions of the differential equations generally conserve various quantities (energy, momentum, volumes in phase space)
- Is it possible to find discretizations whose solutions exactly conserve these same quantities?
- Yes!
  - from a long line of research initiated by Jacobi, Hamilton, Poincare’ and others

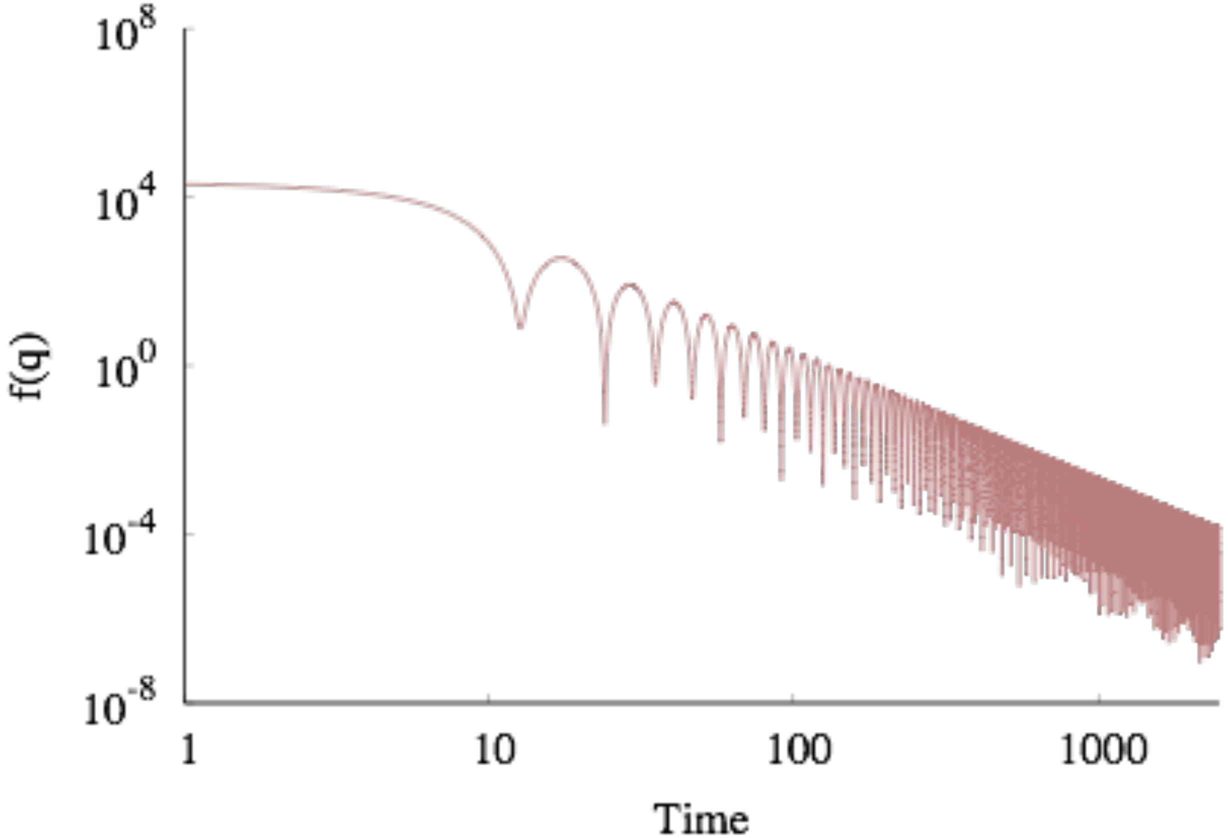
Towards A Symplectic Perspective

- We've discussed discretization of Lagrangian-based dynamics
- Discretization of Lagrangian dynamics is often fragile and requires small step sizes
- We can build more robust solutions by taking a Legendre transform and considering a *Hamiltonian* formalism:

$$L(q, v, t) \rightarrow H(q, p, t, \mathcal{E})$$

$$\left(\frac{\mathrm{d}q}{\mathrm{d}t}, \frac{\mathrm{d}v}{\mathrm{d}t}\right) \rightarrow \left(\frac{\mathrm{d}q}{\mathrm{d}\tau}, \frac{\mathrm{d}p}{\mathrm{d}\tau}, \frac{\mathrm{d}t}{\mathrm{d}\tau}, \frac{\mathrm{d}\mathcal{E}}{\mathrm{d}\tau}\right)$$

Symplectic Integration of Bregman Hamiltonian



Part II: Acceleration and Saddle Points

with Chi Jin and Praneeth Netrapalli

Acceleration in the Nonconvex Setting

Existing literature:

- **AGD** finds $\epsilon-$SP in $O(1/\epsilon^2)$ iterations [Ghadimi and Lan, 2016]
- Nested-loop **gradient** algorithm finds $\epsilon-$SP in $\tilde{O}(1/\epsilon^{1.75})$ iterations [Carmon et al, 2017]
- Nested-loop **Hessian-vector** algorithms finds $\epsilon-$SOSP in $\tilde{O}(1/\epsilon^{1.75})$ iters [Agarwal et al. 2016; Carmon et al 2016]

Acceleration in the Nonconvex Setting

Existing literature:

- **AGD** finds $\epsilon-$SP in $O(1/\epsilon^2)$ iterations [Ghadimi and Lan, 2016]
- Nested-loop **gradient** algorithm finds $\epsilon-$SP in $\tilde{O}(1/\epsilon^{1.75})$ iterations [Carmon et al, 2017]
- Nested-loop **Hessian-vector** algorithms finds $\epsilon-$SOSP in $\tilde{O}(1/\epsilon^{1.75})$ iters [Agarwal et al. 2016; Carmon et al 2016]

Question: Can AGD find $\epsilon-$SOSP efficiently? Faster than GD?

Problem Setup

Smooth Assumption: $f(\cdot)$ is smooth:

- ℓ-gradient Lipschitz, i.e. $\forall \mathbf{x}_1, \mathbf{x}_2,\ \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le \ell \|\mathbf{x}_1 - \mathbf{x}_2\|$.
- ρ-Hessian Lipschitz, i.e. $\forall \mathbf{x}_1, \mathbf{x}_2,\ \|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \le \rho \|\mathbf{x}_1 - \mathbf{x}_2\|$.

Problem Setup

Smooth Assumption: $f(\cdot)$ is smooth:

- ℓ-gradient Lipschitz, i.e. $\forall \mathbf{x}_1, \mathbf{x}_2,\ \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le \ell \|\mathbf{x}_1 - \mathbf{x}_2\|$.
- ρ-Hessian Lipschitz, i.e. $\forall \mathbf{x}_1, \mathbf{x}_2,\ \|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \le \rho \|\mathbf{x}_1 - \mathbf{x}_2\|$.

Goal: find **second-order stationary point** (SOSP):

$$\nabla f(\mathbf{x}) = 0, \quad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \ge 0.$$

Relaxed version: ϵ-**second-order stationary point** (ϵ-SOSP):

$$\|\nabla f(\mathbf{x})\| \le \epsilon, \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(\mathbf{x})) \ge -\sqrt{\rho\epsilon}$$

Algorithm

Perturbed Accelerated Gradient Descent (PAGD)

1. **for** $t = 0, 1, \ldots$ **do**
2. **if** $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$ *and* no perturbation in last T steps **then**
3. $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
4. $\mathbf{y}_t \leftarrow \mathbf{x}_t + (1 - \theta)\mathbf{v}_t$
5. $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta \nabla f(\mathbf{y}_t); \quad \mathbf{v}_{t+1} \leftarrow \mathbf{x}_{t+1} - \mathbf{x}_t$
6. **if** $f(\mathbf{x}_t) \leq f(\mathbf{y}_t) + \langle \nabla f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t \rangle - \frac{\gamma}{2}\|\mathbf{x}_t - \mathbf{y}_t\|^2$ **then**
7. $\mathbf{x}_{t+1} \leftarrow \mathrm{NCE}(\mathbf{x}_t, \mathbf{v}_t, s); \quad \mathbf{v}_{t+1} \leftarrow 0$

Algorithm

Perturbed Accelerated Gradient Descent (PAGD)

1. **for** $t = 0, 1, \ldots$ **do**
2. **if** $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$ *and* no perturbation in last T steps **then**
3. $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t, \qquad \xi_t$ uniformly $\sim \mathbb{B}_0(r)$
4. $\mathbf{y}_t \leftarrow \mathbf{x}_t + (1-\theta)\mathbf{v}_t$
5. $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta \nabla f(\mathbf{y}_t); \quad \mathbf{v}_{t+1} \leftarrow \mathbf{x}_{t+1} - \mathbf{x}_t$
6. **if** $f(\mathbf{x}_t) \leq f(\mathbf{y}_t) + \langle \nabla f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t \rangle - \frac{\gamma}{2}\|\mathbf{x}_t - \mathbf{y}_t\|^2$ **then**
7. $\mathbf{x}_{t+1} \leftarrow \mathrm{NCE}(\mathbf{x}_t, \mathbf{v}_t, s); \quad \mathbf{v}_{t+1} \leftarrow 0$

- Perturbation (line 2-3);
- Standard AGD (line 4-5);
- Negative Curvature Exploitation (NCE, line 6-7)
  - 1) simple (two steps), 2) auxiliary. [inspired by Carmon et al. 2017]

Convergence Result

PAGD Converges to SOSP Faster (Jin, Netrapalli and Jordan, 2017)

For ℓ-gradient Lipschitz and ρ-Hessian Lipschitz function f, PAGD with proper choice of $\eta, \theta, r, T, \gamma, s$ w.h.p. finds ϵ-SOSP in iterations:

$$\tilde{O}\left(\frac{\ell^{1/2}\rho^{1/4}(f(\mathbf{x}_0) - f^\star)}{\epsilon^{7/4}}\right)$$

Convergence Result

PAGD Converges to SOSP Faster (Jin, Netrapalli and Jordan, 2017)

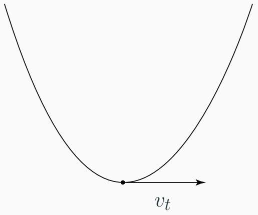
For ℓ-gradient Lipschitz and ρ-Hessian Lipschitz function f, PAGD with proper choice of $\eta, \theta, r, T, \gamma, s$ w.h.p. finds ϵ-SOSP in iterations:

$$\tilde{O}\left(\frac{\ell^{1/2}\rho^{1/4}(f(\mathbf{x}_0) - f^\star)}{\epsilon^{7/4}}\right)$$

	Strongly Convex	**Nonconvex (SOSP)**
Assumptions	ℓ-grad-Lip & α-str-convex	ℓ-grad-Lip & ρ-Hessian-Lip
(Perturbed) GD	$\tilde{O}(\ell/\alpha)$	$\tilde{O}(\Delta_f \cdot \ell/\epsilon^2)$
(Perturbed) AGD	$\tilde{O}(\sqrt{\ell/\alpha})$	$\tilde{O}(\Delta_f \cdot \ell^{\frac{1}{2}}\rho^{\frac{1}{4}}/\epsilon^{\frac{7}{4}})$
Condition κ	ℓ/α	$\ell/\sqrt{\rho\epsilon}$
Improvement	$\sqrt{\kappa}$	$\sqrt{\kappa}$

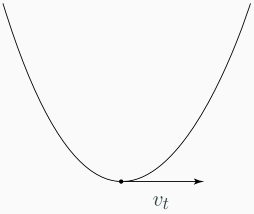
The Hamiltonian

GD: Function value $f(\mathbf{x}_t)$ decreases monotonically. **Not true for AGD.**



The Hamiltonian

GD: Function value $f(\mathbf{x}_t)$ decreases monotonically. **Not true for AGD.**



For **AGD**, in the convex case, **the Hamiltonian decreases monotonically**:

$$E_t = f(\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{v}_t\|^2$$

In the nonconvex case, this isn't true, but it is "nearly true"; i.e., the non-monotonicity is small enough such that NCE suffices to ensure progress

Part III: Acceleration and Stochastics

with Xiang Cheng, Niladri Chatterji and Peter Bartlett

Acceleration and Stochastics

- Can we accelerate diffusions?
- There have been negative results...

Acceleration and Stochastics

- Can we accelerate diffusions?
- There have been negative results…
- …but they've focused on classical overdamped diffusions

Acceleration and Stochastics

- Can we accelerate diffusions?
- There have been negative results…
- …but they've focused on classical overdamped diffusions
- Inspired by our work on acceleration, can we accelerate underdamped diffusions?

Overdamped Langevin MCMC

Described by the Stochastic Differential Equation (SDE):

$$dx_t = -\nabla U(x_t)dt + \sqrt{2}dB_t$$

where $U(x): R^d \to R$ and B_t is standard Brownian motion.

The stationary distribution is $p^*(x) \propto \exp\big(U(x)\big)$

Corresponding Markov Chain Monte Carlo Algorithm (MCMC):

$$\tilde{x}_{(k+1)\delta} = \tilde{x}_{k\delta} - \nabla U(\tilde{x}_{k\delta}) + \sqrt{2\delta}\xi_k$$

where δ is the *step-size* and $\xi_k \sim N(0, I_{d\times d})$

Guarantees under Convexity

Assuming $U(x)$ is L-smooth and m-strongly convex:

Dalalyan'14: Guarantees in Total Variation

If $n \geq O\left(\frac{d}{\epsilon^2}\right)$ then, $TV(p^{(n)}, p^*) \leq \epsilon$

Durmus & Moulines'16: Guarantees in 2-Wasserstein

If $n \geq O\left(\frac{d}{\epsilon^2}\right)$ then, $W_2(p^{(n)}, p^*) \leq \epsilon$

Cheng and Bartlett'17: Guarantees in KL divergence

If $n \geq O\left(\frac{d}{\epsilon^2}\right)$ then, $\mathrm{KL}(p^{(n)}, p^*) \leq \epsilon$

Underdamped Langevin Diffusion

Described by the *second-order* equation:

$$dx_t = v_t dt$$
$$dv_t = -\gamma v_t dt + \lambda \nabla U(x_t) dt + \sqrt{2\gamma\lambda}\, dB_t$$

The stationary distribution is $p^*(x, v) \propto \exp\left(-U(x) - \frac{|v|_2^2}{2\lambda}\right)$

Intuitively, x_t is the position and v_t is the velocity

$\nabla U(x_t)$ is the force and γ is the drag coefficient

Discretization

We can discretize; and at each step evolve according to

$$d\tilde{x}_t = \tilde{v}_t dt$$
$$d\tilde{v}_t = -\gamma \tilde{v}_t dt \ - \lambda \nabla U\left(\tilde{x}_{\lfloor t/\delta \rfloor \delta}\right) dt + \sqrt{2\gamma\lambda}\, dB_t$$

we evolve this for time δ to get an MCMC algorithm

Notice this is a *second-order* method. Can we get faster rates?

Quadratic Improvement

Let $p^{(n)}$ denote the distribution of $(\tilde{x}_{n\delta}, \tilde{v}_{n\delta})$. Assume $U(x)$ is strongly convex

Cheng, Chatterji, Bartlett, Jordan '17:

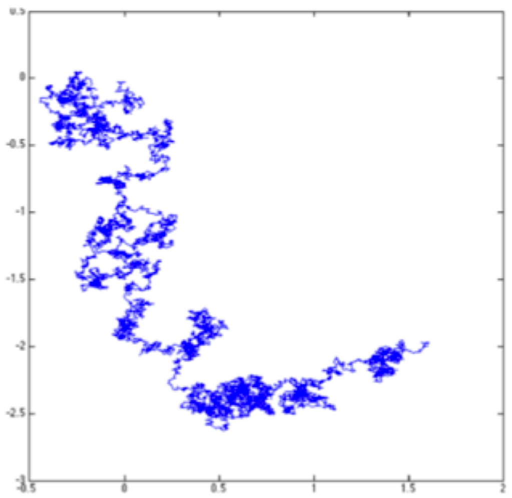
If $n \geq O\left(\frac{\sqrt{d}}{\epsilon}\right)$ then $W_2\left(p^{(n)}, p^*\right) \leq \epsilon$

Compare with Durmus & Moulines '16 (Overdamped)

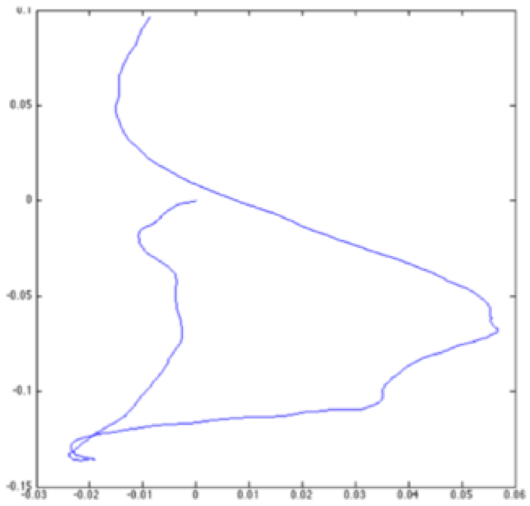
If $n \geq O\left(\frac{d}{\epsilon^2}\right)$ then $W_2\left(p^{(n)}, p^*\right) \leq \epsilon$

Intuition: Smoother Sample Paths

x_t is much smoother for Underdamped Langevin Diffusion, so easier to discretize



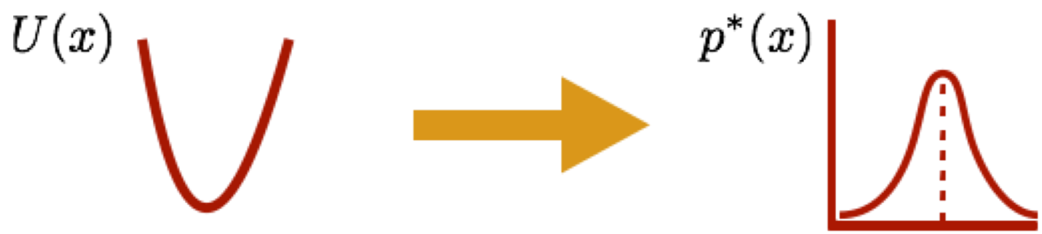
Overdamped Langevin Diffusion



Underdamped Langevin Diffusion

Beyond Convexity?

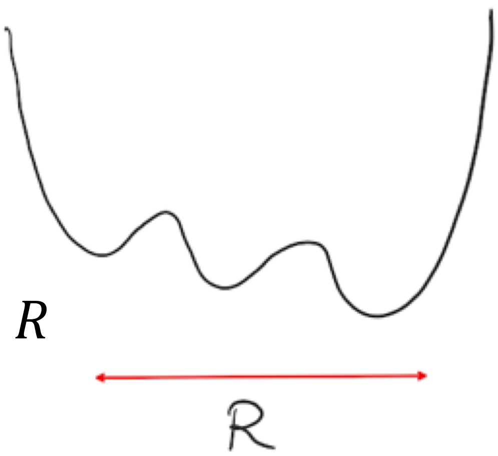
So far we assume $U(x)$ is m- strongly convex



Goal: Establish rates when $U(x)$ is non-convex

Multiple modes

Strongly Convex Outside a Ball



1. Smooth everywhere

   $\forall x, y \; |\nabla \mathrm{U(x)} - \nabla U(y)|_2 \leq L|x \; - y \;|_2$

2. Strongly convex outside a ball $\forall x, y$: $|x - y|_2 \geq R$

   $\langle \nabla U(x) - \nabla U(y), x - y \, \rangle \geq m|x - y \;|_2$

Cheng, Chatterji, Abbasi-Yakdori, Bartlett, & Jordan ’18:

To get $W_1\left(p^{(n)}, p^*\right) \leq \epsilon$:

Overdamped MCMC : $n \geq O\left(\frac{e^{cLR^2} d}{\epsilon^2}\right)$

Underdamped MCMC needs: $n \geq O\left(\frac{e^{cLR^2} \sqrt{d}}{\epsilon}\right)$

Proof Idea: Reflection Coupling

Tricky to prove continuous-time process contracts. Consider two processes,

$$dx_t = -\nabla U(x_t)dt + \sqrt{2}\, dB_t^x$$
$$dy_t = -\nabla U(y_t)dt + \sqrt{2}\, dB_t^y$$

where $x_0 \sim p_0$ and $y_0 \sim p^*$. Couple these through Brownian motion

$$dB_t^y = \left[I_{d\times d} - \frac{2 \cdot (x_t - y_t)(x_t - y_t)^\top}{|x_t - y_t|_2^2} \right] dB_t^x$$

"reflection along line separating the two processes"

Reduction to One Dimension

By Itô's Lemma we can monitor the evolution of the separation distance

$$d|x_t - y_t|_2 = -\left\langle \frac{x_t - y_t}{|x_t - y_t|_2}, \nabla U(x_t) - \nabla U(y_t) \right\rangle dt + 2\sqrt{2} dB_t^1$$

'Drift'

'1-d random walk'

Two cases are possible

1. If $|x_t - y_t|_2 \leq R$ then we have strong convexity; the drift helps.
2. If $|x_t - y_t|_2 \geq R$ then the drift hurts us, but Brownian motion helps stick*

Rates not exponential in d as we have a 1-d random walk

*Under a clever choice of Lyapunov function.