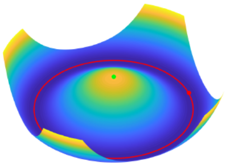
Computational Principles for High-dim Data Analysis

(Lecture Seventeen)

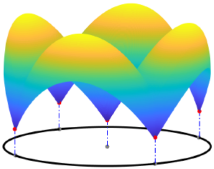
Yi Ma

EECS Department, UC Berkeley

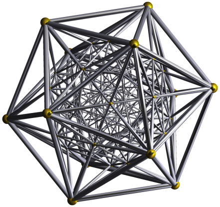
October 28, 2021



Rotational symmetry



Discrete symmetry



Nonconvex Optimization for High-Dim Problems
First Order Methods

"Premature optimization is the root of all evil."
– Donald Knuth, *The Art of Computer Programming*

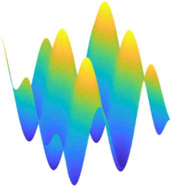
Nonconvex Optimization

Consider the problem of
minimizing a general nonlinear function:

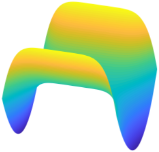
$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathsf{C}. \tag{1}$$

In the worst case, even finding
a *local* minimizer can be NP-hard[1].

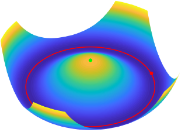
Nonconvex problems that
arise from natural physical, geometrical,
or statistical origins typically have
nice structures, in terms of symmetries!



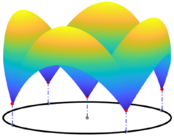
Spurious local minimizers



Flat saddle points



Rotational symmetry



Discrete symmetry

[1] Some NP-complete problems in quadratic and nonlinear programming, K.G Murty and S. N. Kabadi, 1987

Objectives

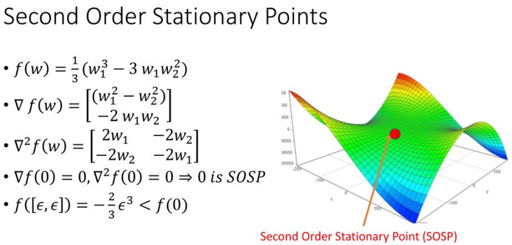
Hence typically people seek to work with relatively benign (gradient/Hessian Lipschitz continuous) functions:

$$\forall \boldsymbol{x}, \boldsymbol{y} \quad \|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\|_2 \leq L_1 \|\boldsymbol{y} - \boldsymbol{x}\|_2 \tag{2}$$

with benign objectives:

1. convergence to some critical point $\boldsymbol{x}_\star$ such that: $\nabla f(\boldsymbol{x}_\star) = \mathbf{0}$;
2. the critical point $\boldsymbol{x}_\star$ is second-order stationary: $\nabla^2 f(\boldsymbol{x}_\star) \succeq \mathbf{0}$.

Example: in general f could have irregular second-order stationary points:

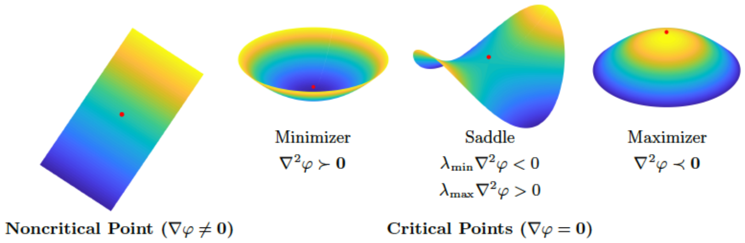


Objectives

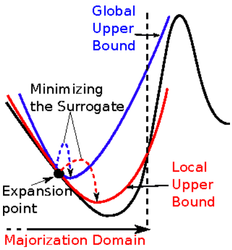
Hence typically people seek to work with relatively benign (gradient/Hessian Lipschitz continuous) functions with benign objectives:

1. convergence to some critical point $\boldsymbol{x}_\star$ such that: $\nabla f(\boldsymbol{x}_\star) = \mathbf{0}$;
2. the critical point $\boldsymbol{x}_\star$ is second-order stationary: $\nabla^2 f(\boldsymbol{x}_\star) \succeq \mathbf{0}$.

Example: a function φ with symmetry only has **regular** critical points:



Gradient Descent (GD)



Function class:
∇f Lipschitz continuous with constant L_1.

First-order oracle:
the gradient $\nabla f(\boldsymbol{x})$ of the function $f(\boldsymbol{x})$.

The gradient descent iteration:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - \gamma_k \nabla f(\boldsymbol{x}_{k-1}). \tag{3}$$

$$\boldsymbol{x}_k := \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}_{k-1}) + \langle \nabla f(\boldsymbol{x}_{k-1}), \boldsymbol{x} - \boldsymbol{x}_{k-1} \rangle + \frac{L_1}{2} \|\boldsymbol{x} - \boldsymbol{x}_{k-1}\|_2^2 \right\}.$$

Proposition (Convergence Rate of GD for Nonconvex Functions)

The gradient descent scheme with the step size $\gamma_k = 1/L_1$ *converges to a critical point* $\boldsymbol{x}_\star$. *Furthermore, for the gradient norm at the best iterate* $\min_{0 \le i \le k-1} \|\nabla f(\boldsymbol{x}_i)\|_2 \le \epsilon_g$, *the number of iterations* $k = O(\epsilon_g^{-2})$.

Newton's Method (strong convex)

Function class: f strongly convex and $\nabla^2 f$ Lipschitz continuous with L_2.

The second-order oracle: the gradient $\nabla f(\boldsymbol{x})$ and the Hessian $\nabla^2 f(\boldsymbol{x})$.

The Newton iteration:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k). \tag{4}$$

Proposition (Convergence Rate of Newton's Method)

Let $f(\boldsymbol{x})$ be strongly convex, with $\lambda_{\min}(\nabla^2 f(\boldsymbol{x})) \geq \lambda > 0$ for all $\boldsymbol{x}$, and assume that $\nabla^2 f$ is Lipschitz continuous with constant L_2, and let $\boldsymbol{x}_\star$ be the (unique) minimizer of f over $\mathbb{R}^n$. Assuming $\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2 < \frac{2\lambda}{L_2}$, the iterates $\boldsymbol{x}_k$ converge to $\boldsymbol{x}_\star$, with quadratic rate.

Unfortunately, for high-dim problems, impossible to compute $\nabla^2 f$.

Cubic Regularized Newton's Method (nonconvex)

Function class: f nonconvex and $\nabla f / \nabla^2 f$ Lips. continuous with L_1/L_2.

The second-order oracle: the gradient $\nabla f(\boldsymbol{x})$ and the Hessian $\nabla^2 f(\boldsymbol{x})$. Consider the local cubic surrogate:

$$\hat{f}(\boldsymbol{y}, \boldsymbol{x}) \doteq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{x})^* \nabla^2 f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) + \frac{L_2}{6} \|\boldsymbol{y} - \boldsymbol{x}\|_2^3 . \quad (5)$$

The cubic Newton iteration:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{y}} \hat{f}(\boldsymbol{y}, \boldsymbol{x}_k). \quad (6)$$

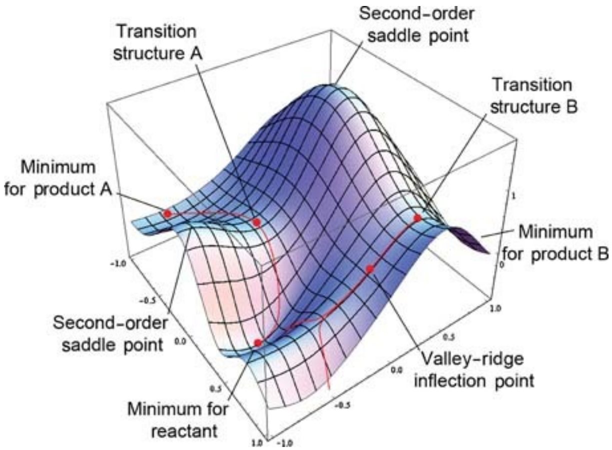
Theorem (Convergence Rate of Cubic Newton's Method)

Suppose $f(\boldsymbol{x})$ is bounded from below. Then the sequence $\{\boldsymbol{x}_k\}$ generated by the cubic regularized Newton step (6) converges to a non-empty set of limit points $\boldsymbol{X}_\star$ of SOS points. For $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon_g$, the number of iterations $k = O(\epsilon_g^{-3/2})$.

Unfortunately, for high-dim problems, impossible to compute $\nabla^2 f$.

Gradient and Negative Curvature Descent

An Intuitive Example: potential energy surface in Chemistry.



Gradient and Negative Curvature Descent

Function class: f nonconvex and $\nabla f / \nabla^2 f$ Lips. continuous with L_1 / L_2.

The oracle: gradient $\nabla f(\boldsymbol{x})$ and a negative eigenvector $\boldsymbol{e}$ of $\nabla^2 f(\boldsymbol{x})$.

Hybrid gradient and negative curvature descent:

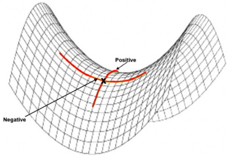
- **if** $\|\nabla f(\boldsymbol{x}_k)\|_2 \geq \epsilon_g = (2L_1\epsilon)^{1/2}$, **then** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L_1}\nabla f(\boldsymbol{x}_k)$;
- **else if** $-\lambda_k(\nabla^2 f(\boldsymbol{x})) \geq \epsilon_H = \left(1.5 L_2^2 \epsilon\right)^{1/3}$, **then** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{2\lambda_k}{L_2}\boldsymbol{e}_k$.

Theorem (Convergence of Gradient and Negative Curvature Descent)

The above hybrid gradient and negative curvature descent scheme converges to a second-order stationary point $\boldsymbol{x}_\star$ with the desired precision in function value ϵ in no more than $k = (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_\star))/\epsilon$ iterations. Or in terms of $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon_g$, $k = O(\epsilon_g^{-2})$.

The same convergence rate as GD, but converges to an SOS point!

Compute Negative Curvature: the Power Iteration



Want to compute negative curvature direction $\boldsymbol{e}$ without Hessian $\boldsymbol{H} \doteq \nabla^2 f(\boldsymbol{x})$:

$$\boldsymbol{H}\boldsymbol{e} = \lambda_{\min}(\boldsymbol{H})\boldsymbol{e} \quad \text{or} \quad \boldsymbol{A}\boldsymbol{e} = \lambda_{\max}(\boldsymbol{A})\boldsymbol{e},$$

with $\boldsymbol{A} \doteq \boldsymbol{I} - L_1^{-1}\boldsymbol{H} \succ \mathbf{0}$.

Power iteration:

$$\hat{\lambda}_{k+1} = \frac{\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle}{\langle \boldsymbol{x}, \boldsymbol{x}\rangle}, \quad \boldsymbol{x} = \boldsymbol{A}^k\boldsymbol{b},$$

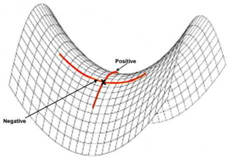
where $\boldsymbol{A}^i\boldsymbol{b}$ can be approximated for a small $t > 0$ with:

$$\boldsymbol{A}\boldsymbol{b} = \left[\boldsymbol{I} - L_1^{-1}\boldsymbol{H}\right]\boldsymbol{b} \;\approx\; \boldsymbol{b} - (tL_1)^{-1}\big(\nabla f(\boldsymbol{x} + t\boldsymbol{b}) - \nabla f(\boldsymbol{x})\big).$$

Two gradient evaluations per iteration.

Compute Negative Curvature: the Lanczos Method

The Krylov information:



$$\boldsymbol{K} \doteq \left[\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^k\boldsymbol{b}\right].$$

The Lanczos method:

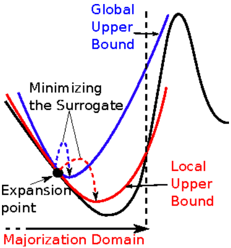
$$\hat{\lambda}_{k+1} = \max_{\boldsymbol{x}} \frac{\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}\rangle}{\langle \boldsymbol{x}, \boldsymbol{x}\rangle}, \quad \boldsymbol{x} \in \text{span}(\boldsymbol{K}).$$

Proposition (Convergence Rate of Lanczos)

Use the Lanczos procedure to find the largest eigenvalue of $\boldsymbol{I} - L_1^{-1}\boldsymbol{H}$ *starting from a random unit vector. Then, for any* $\epsilon_\lambda > 0$ *and* $\delta \in (0,1)$, *with a probability at least* $1-\delta$ *the procedure outputs a unit vector* $\boldsymbol{e}'$ *such that* $(\boldsymbol{e}')^*\boldsymbol{H}\boldsymbol{e}' \leq \lambda_{\min}(\boldsymbol{H}) + \epsilon_\lambda$ *in at most number of iterations:*
$\min\left\{n, \frac{\log(n/\delta^2)}{2\sqrt{2}}\sqrt{\frac{L_1}{\epsilon_\lambda}}\right\}.$

In terms of the first-order oracle, complexity of the inexact gradient and negative curvature descent is $k \leq O(\epsilon_g^{-2})$.

Negative Curvature and Newton Descent



Consider a nonconvex program:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}).$$

Quadratic regularized Newton:

$$\begin{aligned} \boldsymbol{s}_k &= \arg\min_{\boldsymbol{s}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{s} \rangle + \frac{1}{2}\boldsymbol{s}^* \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{s} + \frac{\lambda}{2}\|\boldsymbol{s}\|_2^2 \quad (7) \\ &= -[\nabla^2 f(\boldsymbol{x}_k) + \lambda \boldsymbol{I}]^{-1} \nabla f(\boldsymbol{x}_k). \quad (8) \end{aligned}$$

The Levenberg-Marquardt iteration:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla^2 f(\boldsymbol{x}_k) + \lambda \boldsymbol{I}\right]^{-1} \nabla f(\boldsymbol{x}_k). \quad (9)$$

LM is very popular for solving nonlinear least squares problems.

Negative Curvature and Newton Descent

Function class: f nonconvex and $\nabla f / \nabla^2 f$ Lips. continuous with L_1 / L_2.

The oracle: gradient $\nabla f(\boldsymbol{x})$ and $\nabla^2 f(\boldsymbol{x})$ (to be approximated).

Hybrid curvature and Newton descent (why flip order?):

- **if** $-\lambda_k(\nabla^2 f(\boldsymbol{x})) \geq \epsilon_H = \left(3L_2^2\epsilon\right)^{1/3}$, **then** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{2\lambda_k}{L_2}\boldsymbol{e}_k$;
- **else if** $\|\nabla f(\boldsymbol{x}_k)\|_2 \geq \epsilon_g = 3^{8/3} L_2^{1/3} \epsilon^{2/3}/2$, **then** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma_k \boldsymbol{s}_k$.

Theorem

Assume $\{\boldsymbol{x}_k\}$ are generated by the hybrid negative curvature and Newton descent. Then in at most

$$k \leq \frac{f(\boldsymbol{x}_0) - f(\boldsymbol{x}_\star)}{\epsilon} \tag{10}$$

iterations, $\boldsymbol{x}_k$ will be an an approximate second-order stationary point such that $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon_g, \lambda_{\min}(\nabla^2 f(\boldsymbol{x}_k)) \geq -\epsilon_H$.

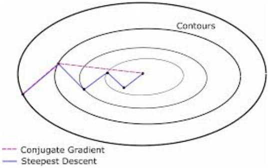
Conjugate Gradient Descent

Need to compute $\boldsymbol{e}_k$ and $\boldsymbol{s}_k$ without knowing $\nabla^2 f(\boldsymbol{x})$. Notice that, similar to $\boldsymbol{e}_k$, to find $\boldsymbol{s}_k$ we need solve: $\underbrace{\left[\nabla^2 f(\boldsymbol{x}_k) + \lambda \boldsymbol{I}\right]}_{\boldsymbol{A}} \boldsymbol{s}_k = \underbrace{-\nabla f(\boldsymbol{x}_k)}_{\boldsymbol{y}}$.

A special case of the quadratic minimization problem: $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$.

Conjugate gradient descent:[2] Initialize the residual $\boldsymbol{r}_i$ and descent direction $\boldsymbol{d}_i$ as: $\boldsymbol{d}_0 = \boldsymbol{r}_0 = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_0$. Then for $i = 0, 1, 2, \ldots$:

Conjugate Gradient: $$\begin{cases} \alpha_i = \dfrac{\boldsymbol{r}_i^* \boldsymbol{r}_i}{\boldsymbol{d}_i^* \boldsymbol{A} \boldsymbol{d}_i}, \\ \boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{d}_i, \\ \boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A}\boldsymbol{d}_i, \\ \beta_{i+1} = \dfrac{\boldsymbol{r}_{i+1}^* \boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^* \boldsymbol{r}_i}, \\ \boldsymbol{d}_{i+1} = \boldsymbol{r}_{i+1} + \beta_{i+1} \boldsymbol{d}_i. \end{cases}$$



[2] An introduction to the conjugate gradient method without the agonizing pain, Jonathan Shewchuk, Technical report, Carnegie Mellon University, 1994.

Negative Curvature and Newton Descent: Complexity

Theorem (Complexity of Approximate Conjugate Gradient)

To solve $\boldsymbol{A}\boldsymbol{s} = \boldsymbol{y}$ with $\epsilon_H \boldsymbol{I} \preceq \boldsymbol{A} \preceq (L_1 + 2\epsilon_H)\boldsymbol{I}$, the conjugate gradient method computes an $\boldsymbol{s}'$ that satisfies

$$\left\|(\nabla^2 f(\boldsymbol{x}_k) + 2\epsilon_H)\boldsymbol{s}_k + \nabla f(\boldsymbol{x}_k)\right\|_2 \leq \frac{1}{2}\epsilon_H \|\boldsymbol{s}_k\|_2$$

in at most $O\big(\epsilon_H^{-1/2} \log(\frac{1}{\epsilon_H})\big)$ iterations.

With the first-order oracle, complexity of the inexact negative curvature and newton descent achieves the best known rate: $k \leq O(\epsilon_g^{-7/4})$.

Gradient Descent with Small Random Noise

Function class: f nonconvex and $\nabla f/\nabla^2 f$ Lips. continuous with L_1/L_2.

The oracle: gradient $\nabla f(\boldsymbol{x})$ and small random noise.

The **Langevin dynamics** with noisy gradient flow:

$$\dot{\boldsymbol{x}}(t) = -\frac{1}{2}\nabla f(\boldsymbol{x}(t)) + \sqrt{\lambda}\boldsymbol{n}(t), \quad (11)$$

Probability density of $\boldsymbol{x}$ converges to the **Gibbs measure**:

$$p^{\lambda}(\boldsymbol{x}) = C^{\lambda}\exp\Big(-\frac{1}{\lambda}f(\boldsymbol{x})\Big). \quad (12)$$

Lemma (Laplace's Method: Scalar Case)

Suppose $f(x)$ is a twice continuously differentiable function with a unique maximizer x_0 and $f''(x_0) < 0$. Then we have

$$\lim_{\lambda\to 0}\int e^{\frac{1}{\lambda}f(x)}dx \;=\; e^{\frac{1}{\lambda}f(x_0)}\sqrt{\frac{2\pi\lambda}{-f''(x_0)}} \;\propto\; \int e^{\frac{1}{\lambda}f(x)}\delta(x-x_0)dx. \quad (13)$$

The Laplace Method

Theorem (Laplace Method: Multivariate and Multiple Global Minimizers)

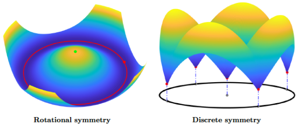
Let $f(\boldsymbol{x})$ be a function with at least quadratic growth as $\boldsymbol{x} \to \infty$. Suppose $f(\boldsymbol{x})$ has multiple global non-degenerate minimizers at $\boldsymbol{x}_\star^1, \ldots, \boldsymbol{x}_\star^N$ and they are all non-degenerate. Then in the limit $\lambda \downarrow 0$, the density $p^\lambda(\boldsymbol{x})$ converges to

$$p^0(\boldsymbol{x}) = \frac{\sum_{i=1}^N a_i \delta(\boldsymbol{x} - \boldsymbol{x}_\star^i)}{\sum_{i=1}^N a_i}, \quad \textit{with} \quad a_i = \det[\boldsymbol{H}(\boldsymbol{x}_\star^i)]^{-1/2}, \tag{14}$$

where $\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x})$ is the Hessian of the function $f(\boldsymbol{x})$.

When all global minimizers make a continuous submanifold $\mathcal{M}$, $p^\lambda(\boldsymbol{x})$ converges to a density on $\mathcal{M}$ given by:

$$p^0(\boldsymbol{x}) = \frac{\det[\boldsymbol{H}(\boldsymbol{x})]^{-1/2}}{\int_{\mathcal{M}} \det[\boldsymbol{H}(\boldsymbol{y})]^{-1/2} d\boldsymbol{y}}, \quad \boldsymbol{x} \in \mathcal{M}.$$



Noisy Gradient with Langevin Monte Carlo

Function class: ∇f Lipschitz continuous with constant L_1.

First-order oracle: the gradient $\nabla f(\boldsymbol{x})$ and small noise $\boldsymbol{n}$.

Langevin Monte Carlo:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L_1}\nabla f(\boldsymbol{x}_k) + \sqrt{2\lambda/L_1}\,\boldsymbol{n}_k. \tag{15}$$

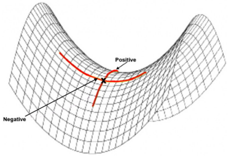
Proposition (Noisy Gradient Descent)

Considering the above noisy gradient descent scheme (15), *if* $\|\nabla f(\boldsymbol{x}_k)\|_2 \geq (2L_1\epsilon)^{1/2}$, *then we have*

$$\mathbb{E}[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k] \leq f(\boldsymbol{x}_k) - \epsilon + \lambda. \tag{16}$$

Descent when $\|\nabla f(\boldsymbol{x}_k)\|_2 > (2L_1\lambda)^{1/2}$; explore stability otherwise.

Effect of Noisy Gradient around a Saddle Point



Consider a standard quadratic function:
$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^*\boldsymbol{H}\boldsymbol{x}$ for a constant $\boldsymbol{H} \in \mathbb{R}^{n\times n}$,
with the smallest eigenvalue $\lambda_{\min} < 0$,
and the Lipschitz constant $L_1 = \max_i |\lambda_i(\boldsymbol{H})|$.

The Langevin dynamics becomes:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \frac{1}{L_1}\nabla f(\boldsymbol{x}_k) + \sqrt{2\lambda/L_1}\,\boldsymbol{n}_k \\ &= \underbrace{(\boldsymbol{I} - L_1^{-1}\boldsymbol{H})}_{\boldsymbol{A}}\boldsymbol{x}_k + \underbrace{\sqrt{2\lambda/L_1}}_{b}\,\boldsymbol{n}_k. \end{aligned} \tag{17}$$

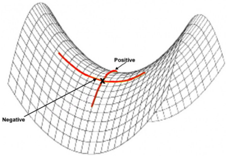
Since $\lambda_{\max}(\boldsymbol{A}) = 1 - \lambda_{\min}(\boldsymbol{H})/L_1 > 1$, this is **an unstable linear dynamic system** with random noise as the input:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + b\,\boldsymbol{n}_k. \tag{18}$$

Escaping Saddle Point

Therefore, the accumulated dynamics:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}^{k+1}\boldsymbol{x}_0 + b\sum_{i=0}^{k}\boldsymbol{A}^{k-i}\boldsymbol{n}_i. \tag{19}$$



$\boldsymbol{A}^{k+1}\boldsymbol{x}_0$ and $\boldsymbol{A}^{k-i}\boldsymbol{n}_i$ are **powers** of the matrix $\boldsymbol{A}$ applied to random vectors (assuming $\boldsymbol{x}_0$ random too).

Question: which direction survives in power iteration?

Proposition (Escaping Saddle Point via Noisy Gradient Descent)

Consider the noisy gradient descent via the Langevin dynamics (17) *for the function* $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^*\boldsymbol{H}\boldsymbol{x}$, *starting from* $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. *Then after* $k \geq \frac{\log n - \log(|\lambda_{\min}|/L_1)}{2\log(1+|\lambda_{\min}|/L_1)}$ *steps, we have*

$$\mathbb{E}[f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_0)] \leq -\lambda. \tag{20}$$

Hybrid Noisy Gradient Descent

Function class: f nonconvex and $\nabla f / \nabla^2 f$ Lips. continuous with L_1/L_2.

The oracle: gradient $\nabla f(\boldsymbol{x})$ and small noise $\boldsymbol{n}$.

Hybrid noisy gradient descent:

- **if** $\|\nabla f(\boldsymbol{x}_k)\|_2 \geq \epsilon_g$, **then** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L_1}\nabla f(\boldsymbol{x}_k)$;
- **else** $\boldsymbol{x}_k^0 = \boldsymbol{x}_k$, and negative curvature descent with noisy gradients: **for** $i = 0, 1, 2, \ldots, k_{\max} = O(\log n)$

$$\boldsymbol{x}_k^{i+1} = \boldsymbol{x}_k^i - \frac{1}{L_1}\nabla f(\boldsymbol{x}_k^i) + \sqrt{2\epsilon / L_1}\boldsymbol{n}^i,$$

  where $\boldsymbol{n}^i \sim \mathcal{N}(0, \boldsymbol{I})$.

Complexity: To guarantee $\|\nabla f(\boldsymbol{x})\| \leq \epsilon_g$, the number of total gradient evaluations needed is $O(\epsilon_g^{-2})$, up to a $\log(n)$ factor.[3]

[3] Perturbed accelerated gradient descent reduces to $O(\epsilon_g^{-7/4})$.

Summary

Table: Oracles and complexities (up to log factors) of different optimization methods. Complexity is measured in terms of the number of oracles accessed before attaining a prescribed accuracy $\|\nabla f(\boldsymbol{x}_\star)\| \leq \epsilon_g$.

Methods	Oracles	Stat. Points	Complexity
Vanilla gradient descent	first-order	first-order	$O(\epsilon_g^{-2})$
Cubic Regularized Newton	second-order	second-order	$O(\epsilon_g^{-1.5})$
Gradient/negative curvature	first-order	second-order	$O(\epsilon_g^{-2})$
Negative curvature/Newton	first-order	second-order	$O(\epsilon_g^{-1.75})$
Hybrid noisy gradient	first-order	second-order	$O(\epsilon_g^{-2})$
Perturbed accelerated gradient	first-order	second-order	$O(\epsilon_g^{-1.75})$

The (probably only) two fundamental ideas for first-order optimization:

Gradient Descent and Acceleration.

Power Iteration for Inexact Newton Descent.

Assignments

- Reading: Section 9.1 - 9.5 of Chapter 9.
- Programming Homework #3.