Nonconvex first-order optimization: When can gradient descent escape saddle points in linear time?

Rishabh Dixit and **Waheed U. Bajwa**

Department of Electrical and Computer Engineering
Rutgers University–New Brunswick, NJ USA
www.inspirelab.us

Bellairs Research Institute Workshop
December 13, 2021

Lagrange Program

CCF-1453073
CCF-1910110
CCF-1907658

W911NF-17-1-0546
W911NF-21-1-0301

Numerical optimization: A workhorse of the digital age

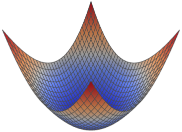
$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$$

Applications: Machine learning, signal processing, statistics, robotics, computer vision, wireless communications, ...

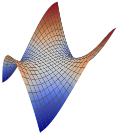
$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$$

Applications: Machine learning, signal processing, statistics, robotics, computer vision, wireless communications, . . .

Convex functions



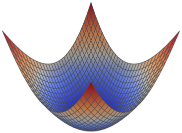
Nonconvex functions



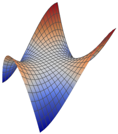
$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$$

Applications: Machine learning, signal processing, statistics, robotics, computer vision, wireless communications, . . .

Convex functions



Nonconvex functions

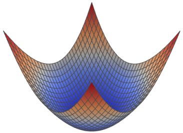


- Plethora of work, going back decades
- Known oracle complexity of problems
- Many classes of near-optimal methods

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

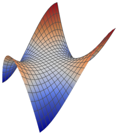
Applications: Machine learning, signal processing, statistics, robotics, computer vision, wireless communications, . . .

Convex functions



- Plethora of work, going back decades
- Known oracle complexity of problems
- Many classes of near-optimal methods

Nonconvex functions



- Traditional focus on convexification
- Recent focus on certain geometries
- Still much remains unknown ...

Outline

Outline

The nonconvex optimization problem

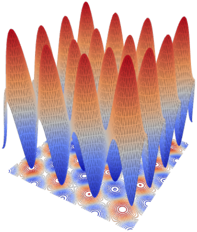
Objective: $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ using the first-order (gradient) information $\nabla f(\mathbf{x})$

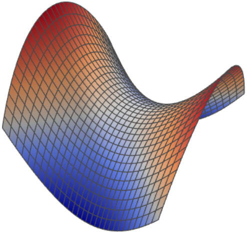
The nonconvex optimization problem

Objective: $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ using the first-order (gradient) information $\nabla f(\mathbf{x})$

Challenges for nonconvex functions

- A nonconvex landscape can have three types of attractive stationary points ($\nabla f(\mathbf{x}) = \mathbf{0}$): global minima, local minima, and saddle points





The nonconvex optimization problem

Objective: $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ using the first-order (gradient) information $\nabla f(\mathbf{x})$

Challenges for nonconvex functions

- A nonconvex landscape can have three types of attractive stationary points ($\nabla f(\mathbf{x}) = \mathbf{0}$): global minima, local minima, and saddle points
- Any first-order method will likely encounter many saddle neighborhoods in its trajectory, which will eventually determine its convergence behavior

The nonconvex optimization problem

Objective: $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ using the first-order (gradient) information $\nabla f(\mathbf{x})$

Challenges for nonconvex functions

- A nonconvex landscape can have three types of attractive stationary points ($\nabla f(\mathbf{x}) = \mathbf{0}$): global minima, local minima, and saddle points
- Any first-order method will likely encounter many saddle neighborhoods in its trajectory, which will eventually determine its convergence behavior
- **How long does a first-order method spend in a saddle neighborhood** is not that straightforward due to the local regions of attraction and repulsion

The nonconvex optimization problem

Objective: $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ using the first-order (gradient) information $\nabla f(\mathbf{x})$

Challenges for nonconvex functions

- A nonconvex landscape can have three types of attractive stationary points ($\nabla f(\mathbf{x}) = \mathbf{0}$): global minima, local minima, and saddle points
- Any first-order method will likely encounter many saddle neighborhoods in its trajectory, which will eventually determine its convergence behavior
- **How long does a first-order method spend in a saddle neighborhood** is not that straightforward due to the local regions of attraction and repulsion

An approach: Assume specialized geometry for $f(\mathbf{x})$ such as *essential strong convexity*, *weak strong convexity*, *restricted strong convexity*, *Polyak–Łojasiewicz condition*, and *quadratic growth condition*

- All but the quadratic growth condition imply all local minimizers are global minimizers and there are no saddle points in the function landscape

Nonconvex optimization: State-of-the-art on saddle escape

Continuous-time analysis

- Stochastic differential equation approach: Kifer, 1981; Shi, Su, and Jordan, 2020; J. Yang, Hu, and C. J. Li, 2021
- Normalized gradient flow curves: Murray, Swenson, and Kar, 2019

Geometric landscape analysis

- Statistical estimation problems: X. Li et al., 2019; Ma et al., 2020

Asymptotic analysis

- Stochastic gradient (Langevin) dynamics: Gelfand and Mitter, 1991; Mertikopoulos et al., 2020
- Measure theoretic results: Lee et al., 2017; O'Neill and Wright, 2019

Nonconvex optimization: State-of-the-art on saddle escape

Noise injection / stochasticity for saddle escape

- Perturbed gradient descent: Du et al., 2017; Jin, Ge, et al., 2017
- Curvature-based perturbation: Daneshmand et al., 2018
- Langevin dynamics: Raginsky, Rakhlin, and Telgarsky, 2017; Erdogdu, Mackey, and Shamir, 2018
- Accelerated methods: Jin, Netrapalli, and Jordan, 2018; Reddi et al., 2018; Xu, Rong, and T. Yang, 2018

Higher-order methods

- Anandkumar and Ge, 2016; Mokhtari, Ozdaglar, and Jadbabaie, 2018; Paternain, Mokhtari, and Ribeiro, 2019

Nonconvex optimization: State-of-the-art on saddle escape

Noise injection / stochasticity for saddle escape

- Perturbed gradient descent: Du et al., 2017; Jin, Ge, et al., 2017
- Curvature-based perturbation: Daneshmand et al., 2018
- Langevin dynamics: Raginsky, Rakhlin, and Telgarsky, 2017; Erdogdu, Mackey, and Shamir, 2018
- Accelerated methods: Jin, Netrapalli, and Jordan, 2018; Reddi et al., 2018; Xu, Rong, and T. Yang, 2018

Higher-order methods

- Anandkumar and Ge, 2016; Mokhtari, Ozdaglar, and Jadbabaie, 2018; Paternain, Mokhtari, and Ribeiro, 2019

But how does the 'vanilla' gradient descent behave around saddle neighborhoods?

Understanding gradient descent through its trajectories

Gradient descent (GD) iteration: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$

Overarching Goal: Study the GD trajectories $\{\mathbf{x}_k\}$, as a function of the initialization $\mathbf{x}_0$, for general nonconvex functions $f(\cdot)$

Understanding gradient descent through its trajectories

Gradient descent (GD) iteration: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$

Overarching Goal: Study the GD trajectories $\{\mathbf{x}_k\}$, as a function of the initialization $\mathbf{x}_0$, for general nonconvex functions $f(\cdot)$

The study of trajectories helps address the following questions:

- What trajectories around saddle points can be considered useful in the sense of 'fast' saddle escape?
- Given a trajectory starting around a saddle point, can we understand (and subsequently control) its behavior by knowing its initial conditions?

References

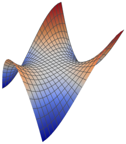
1. R. Dixit and **B.**, "Exit time analysis for approximations of gradient descent trajectories around saddle points," arXiv:2006.01106, Jun. 2020.
2. R. Dixit and **B.**, "Boundary conditions for linear exit time gradient trajectories around saddle points: Analysis and algorithm," arXiv:2101.02625, Jan. 2021.

Outline

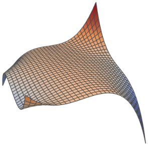
Assumptions

The nonconvex $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable Morse function (i.e., has non-degenerate saddles), along with the following assumptions:

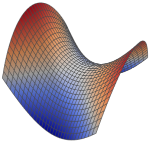
1. It is **locally analytic** around saddle points (i.e., admits Taylor expansion)
2. It has L-**Lipschitz gradients**: $\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|$
3. It has M-**Lipschitz Hessians**: $\|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\|_2 \leq M\|\mathbf{x}_1 - \mathbf{x}_2\|$
4. It has **well-conditioned strict saddles**: $\min_i |\lambda_i(\nabla^2 f(\mathbf{x}^*)| > \beta$
5. The **minimum gap** between any two **degenerate eigenvalue groups** of the Hessian $\nabla^2 f(\mathbf{x}^*)$ at any strict saddle is δ



Non-strict saddle



Degenerate strict saddle

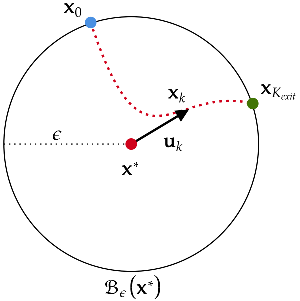


Morse function strict saddle

Setup: Given a strict saddle point $\mathbf{x}^*$ of $f(\cdot)$, suppose the gradient descent trajectory $\{\mathbf{x}_k\}$ starts on the boundary of the ball $\mathcal{B}_\epsilon(\mathbf{x}^*)$ at $k = 0$ and it exits $\mathcal{B}_\epsilon(\mathbf{x}^*)$ at $k = K_{exit}$

The radial vector: $\mathbf{u}_k := \mathbf{x}_k - \mathbf{x}^*$

The exit time: $K_{exit} := \inf_{k \geq 1} \left\{ k \middle| \|\mathbf{u}_k\|^2 > \epsilon^2 \right\}$

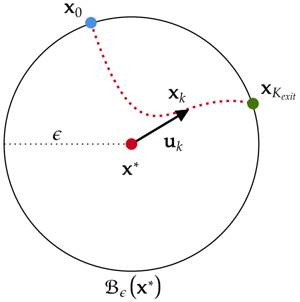


The exit time of a gradient descent trajectory

Setup: Given a strict saddle point $\mathbf{x}^*$ of $f(\cdot)$, suppose the gradient descent trajectory $\{\mathbf{x}_k\}$ starts on the boundary of the ball $\mathcal{B}_\epsilon(\mathbf{x}^*)$ at $k = 0$ and it exits $\mathcal{B}_\epsilon(\mathbf{x}^*)$ at $k = K_{exit}$

The radial vector: $\mathbf{u}_k := \mathbf{x}_k - \mathbf{x}^*$

The exit time: $K_{exit} := \inf\limits_{k \geq 1} \left\{ k \middle| \|\mathbf{u}_k\|^2 > \epsilon^2 \right\}$

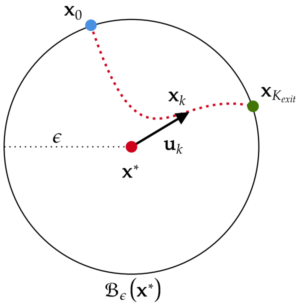


Objective I: Investigate whether there exists K_{exit} for which the sequence $\{\mathbf{x}_k\}_{k > K_{exit}}$ lies outside $\mathcal{B}_\epsilon(\mathbf{x}^*)$ such that $K_{exit} = \mathcal{O}(\log(\epsilon^{-1}))$

Setup: Given a strict saddle point $\mathbf{x}^*$ of $f(\cdot)$, suppose the gradient descent trajectory $\{\mathbf{x}_k\}$ starts on the boundary of the ball $\mathcal{B}_\epsilon(\mathbf{x}^*)$ at $k = 0$ and it exits $\mathcal{B}_\epsilon(\mathbf{x}^*)$ at $k = K_{exit}$

The radial vector: $\mathbf{u}_k := \mathbf{x}_k - \mathbf{x}^*$

The exit time: $K_{exit} := \inf_{k \geq 1} \left\{ k \middle| \|\mathbf{u}_k\|^2 > \epsilon^2 \right\}$



Objective I: Investigate whether there exists K_{exit} for which the sequence $\{\mathbf{x}_k\}_{k > K_{exit}}$ lies outside $\mathcal{B}_\epsilon(\mathbf{x}^*)$ such that $K_{exit} = \mathcal{O}(\log(\epsilon^{-1}))$

Objective II: Derive sufficient conditions on $\mathbf{x}_0$ for guaranteeing the linear exit time and develop a robust gradient descent-based algorithm

What allows a GD trajectory to escape the saddle point?

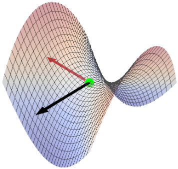
A dynamical system perspective (Shub, 2013; Lee et al., 2017): A GD trajectory can be viewed as a dynamical system, with each strict saddle $\mathbf{x}^*$ imparting both **attractive** (stable) and **repulsive** (unstable) dynamics on the trajectory

What allows a GD trajectory to escape the saddle point?

A dynamical system perspective (Shub, 2013; Lee et al., 2017): A GD trajectory can be viewed as a dynamical system, with each strict saddle $\mathbf{x}^*$ imparting both **attractive** (stable) and **repulsive** (unstable) dynamics on the trajectory

The stable and unstable subspaces of a strict saddle: Let $(\lambda_i, \mathbf{v}_i)$ be the i^{th} eigenvalue–eigenvector pair of the Hessian $\nabla^2 f(\mathbf{x}^*)$, then:

- The stable subspace $\mathcal{E}_S = \text{span}\{\mathbf{v}_i | \lambda_i > 0\}$ is attractive
- The unstable subspace $\mathcal{E}_{US} = \text{span}\{\mathbf{v}_i | \lambda_i < 0\}$ is repulsive

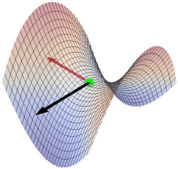


What allows a GD trajectory to escape the saddle point?

A dynamical system perspective (Shub, 2013; Lee et al., 2017): A GD trajectory can be viewed as a dynamical system, with each strict saddle $\mathbf{x}^*$ imparting both **attractive** (stable) and **repulsive** (unstable) dynamics on the trajectory

The stable and unstable subspaces of a strict saddle: Let $(\lambda_i, \mathbf{v}_i)$ be the i^{th} eigenvalue–eigenvector pair of the Hessian $\nabla^2 f(\mathbf{x}^*)$, then:

- The stable subspace $\mathcal{E}_S = \text{span}\{\mathbf{v}_i | \lambda_i > 0\}$ is attractive
- The unstable subspace $\mathcal{E}_{US} = \text{span}\{\mathbf{v}_i | \lambda_i < 0\}$ is repulsive



Challenge: A careful characterization of the exit time for a GD trajectory requires a precise handle on the **stable and unstable projections** of the trajectory

Recipe (Step I): A Hessian-based gradient approximation

Claim: Let $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*)$ be any point in the saddle neighborhood and define $\mathbf{u} := \mathbf{x} - \mathbf{x}^*$ to be the **radial vector**. Then

$$\nabla f(\mathbf{x}) = (\nabla^2 f(\mathbf{x}^*) + \mathcal{O}(\epsilon))\mathbf{u}$$

Recipe (Step I): A Hessian-based gradient approximation

Claim: Let $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*)$ be any point in the saddle neighborhood and define $\mathbf{u} := \mathbf{x} - \mathbf{x}^*$ to be the **radial vector**. Then

$$\nabla f(\mathbf{x}) = (\nabla^2 f(\mathbf{x}^*) + \mathcal{O}(\epsilon))\mathbf{u}$$

Proof

1. We can write $\nabla f(\mathbf{x}) = \left(\int_{p=0}^{p=1} \nabla^2 f(\mathbf{x}^* + p\mathbf{u}) dp \right)\mathbf{u}$

Recipe (Step I): A Hessian-based gradient approximation

Claim: Let $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*)$ be any point in the saddle neighborhood and define $\mathbf{u} := \mathbf{x} - \mathbf{x}^*$ to be the **radial vector**. Then

$$\nabla f(\mathbf{x}) = (\nabla^2 f(\mathbf{x}^*) + \mathcal{O}(\epsilon))\mathbf{u}$$

Proof

1. We can write $\nabla f(\mathbf{x}) = \left(\int_{p=0}^{p=1} \nabla^2 f(\mathbf{x}^* + p\mathbf{u}) dp \right) \mathbf{u}$
2. The Hessian $\nabla^2 f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^* + p\mathbf{u}$, where $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*)$, $p \in [0, 1]$, and $\|\mathbf{u}\| \leq \epsilon$, can be expressed as

$$\nabla^2 f(\mathbf{x}^* + p\mathbf{u}) = \nabla^2 f(\mathbf{x}^*) + \mathbf{D}(\mathbf{x}),$$

with the perturbation matrix $\mathbf{D}(\mathbf{x})$ bounded as

$$\|\mathbf{D}(\mathbf{x})\| \leq Mp\epsilon.$$

Recipe (Step I): A Hessian-based gradient approximation

Claim: Let $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*)$ be any point in the saddle neighborhood and define $\mathbf{u} := \mathbf{x} - \mathbf{x}^*$ to be the **radial vector**. Then

$$\nabla f(\mathbf{x}) = (\nabla^2 f(\mathbf{x}^*) + \mathcal{O}(\epsilon))\mathbf{u}$$

Proof

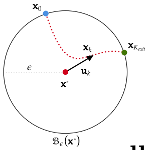
1. We can write $\nabla f(\mathbf{x}) = \left(\int_{p=0}^{p=1} \nabla^2 f(\mathbf{x}^* + p\mathbf{u}) dp \right) \mathbf{u}$
2. The Hessian $\nabla^2 f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^* + p\mathbf{u}$, where $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*)$, $p \in [0, 1]$, and $\|\mathbf{u}\| \leq \epsilon$, can be expressed as

$$\nabla^2 f(\mathbf{x}^* + p\mathbf{u}) = \nabla^2 f(\mathbf{x}^*) + \mathbf{D}(\mathbf{x}),$$

with the perturbation matrix $\mathbf{D}(\mathbf{x})$ bounded as

$$\|\mathbf{D}(\mathbf{x})\| \leq Mp\epsilon.$$

3. Hence, $\nabla f(\mathbf{x}) = \nabla^2 f(\mathbf{x}^*)\mathbf{u} + \left(\int_{p=0}^{p=1} \mathbf{D}(\mathbf{x}) dp \right) \mathbf{u} = (\nabla^2 f(\mathbf{x}^*) + \mathcal{O}(\epsilon))\mathbf{u}$

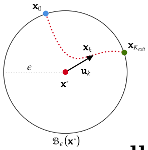


An iterative form of the radial vector

$$\mathbf{u}_{k+1} = \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla f(\mathbf{x}_k) = \left(\mathbf{I} - \alpha \int_0^1 \nabla^2 f(\mathbf{x}^* + p\mathbf{u}_k) dp\right) \mathbf{u}_k$$

$$\implies \mathbf{u}_{k+1} = \left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \underbrace{\alpha \int_0^1 \mathbf{D}(\mathbf{x}^* + p\mathbf{u}_k) dp}_{\mathbf{R}(\mathbf{u}_k) = \mathcal{O}(\epsilon)}\right) \mathbf{u}_k.$$

Recipe (Step II): Trajectory in terms of the radial vector



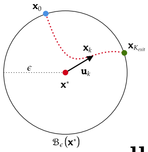
An iterative form of the radial vector

$$\mathbf{u}_{k+1} = \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla f(\mathbf{x}_k) = \left(\mathbf{I} - \alpha \int_0^1 \nabla^2 f(\mathbf{x}^* + p\mathbf{u}_k) dp \right) \mathbf{u}_k$$

$$\implies \mathbf{u}_{k+1} = \Bigg(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \underbrace{\alpha \int_0^1 \mathbf{D}(\mathbf{x}^* + p\mathbf{u}_k) dp}_{\mathbf{R}(\mathbf{u}_k) = \mathcal{O}(\epsilon)} \Bigg) \mathbf{u}_k.$$

- Iteration in terms of initialization: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K} \left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k) \right) \mathbf{u}_0$

Recipe (Step II): Trajectory in terms of the radial vector

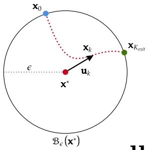


An iterative form of the radial vector

$$\mathbf{u}_{k+1} = \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla f(\mathbf{x}_k) = \left(\mathbf{I} - \alpha \int_0^1 \nabla^2 f(\mathbf{x}^* + p\mathbf{u}_k) dp\right)\mathbf{u}_k$$

$$\implies \mathbf{u}_{k+1} = \left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \underbrace{\alpha \int_0^1 \mathbf{D}(\mathbf{x}^* + p\mathbf{u}_k) dp}_{\mathbf{R}(\mathbf{u}_k) = \mathcal{O}(\epsilon)}\right)\mathbf{u}_k.$$

- Iteration in terms of initialization: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K}\left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\right)\mathbf{u}_0$
- **How to approximate $\mathbf{u}_{K+1}$ from the above relation?**



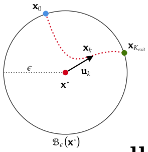
An iterative form of the radial vector

$$\mathbf{u}_{k+1} = \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla f(\mathbf{x}_k) = \left(\mathbf{I} - \alpha \int_0^1 \nabla^2 f(\mathbf{x}^* + p\mathbf{u}_k) dp\right)\mathbf{u}_k$$

$$\implies \mathbf{u}_{k+1} = \Bigg(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \underbrace{\alpha \int_0^1 \mathbf{D}(\mathbf{x}^* + p\mathbf{u}_k) dp}_{\mathbf{R}(\mathbf{u}_k) = \mathcal{O}(\epsilon)}\Bigg)\mathbf{u}_k.$$

- Iteration in terms of initialization: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K}\left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\right)\mathbf{u}_0$
- **How to approximate $\mathbf{u}_{K+1}$ from the above relation?**
  - Zeroth-order: $\mathbf{u}_{K+1} \approx \Pi_{r=0}^{K}\left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*)\right)\mathbf{u}_0$ ✗

Recipe (Step II): Trajectory in terms of the radial vector

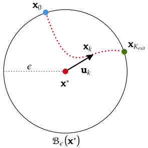


An iterative form of the radial vector

$$\mathbf{u}_{k+1} = \mathbf{x}_k - \mathbf{x}^* - \alpha \nabla f(\mathbf{x}_k) = \left(\mathbf{I} - \alpha \int_0^1 \nabla^2 f(\mathbf{x}^* + p\mathbf{u}_k) dp\right) \mathbf{u}_k$$

$$\implies \mathbf{u}_{k+1} = \left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \underbrace{\alpha \int_0^1 \mathbf{D}(\mathbf{x}^* + p\mathbf{u}_k) dp}_{\mathbf{R}(\mathbf{u}_k) = \mathcal{O}(\epsilon)}\right) \mathbf{u}_k.$$

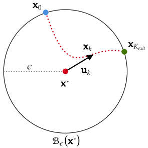
- Iteration in terms of initialization: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K} \left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\right) \mathbf{u}_0$
- **How to approximate $\mathbf{u}_{K+1}$ from the above relation?**
  - Zeroth-order: $\mathbf{u}_{K+1} \approx \Pi_{r=0}^{K} \left(\mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*)\right) \mathbf{u}_0$ ✗
  - First-order: How to handle $\mathbf{R}(\mathbf{u}_k)$? **Answer:** Use local analyticity of $f(\cdot)$



The radial vector: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K}\Big(\mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\Big)\mathbf{u}_0$

How to get a handle on the product of $K+1$ **non-commuting** matrices?

Proof layout for a linear exit time bound

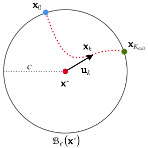


The radial vector: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K}\Big(\mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\Big)\mathbf{u}_0$

How to get a handle on the product of $K+1$ **non-commuting** matrices?

1. Use the **matrix perturbation theory** to express the matrices $\mathbf{R}(\mathbf{u}_k)$

Proof layout for a linear exit time bound



The radial vector: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K}\Big(\mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\Big)\mathbf{u}_0$

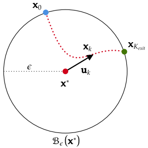
How to get a handle on the product of $K+1$ **non-commuting** matrices?

1. Use the **matrix perturbation theory** to express the matrices $\mathbf{R}(\mathbf{u}_k)$
2. Approximate the product up to **first-order** in order to obtain an "**approximate trajectory**" $\{\tilde{\mathbf{u}}_K\}$ as follows:

$$\tilde{\mathbf{u}}_{K+1} := \Pi_{k=0}^{K}\mathbf{A}_k\mathbf{u}_0 - \sum_{r=0}^{K}(\Pi_{k=r+1}^{K}\mathbf{A}_r\mathbf{R}(\mathbf{u}_r)\Pi_{k=0}^{r-1}\mathbf{A}_r)\mathbf{u}_0,$$

where $\mathbf{A}_k := \mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*)$ for all k.

Proof layout for a linear exit time bound



The radial vector: $\mathbf{u}_{K+1} = \Pi_{k=0}^{K}\Big(\mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*) - \mathbf{R}(\mathbf{u}_k)\Big)\mathbf{u}_0$

How to get a handle on the product of $K+1$ **non-commuting** matrices?

1. Use the **matrix perturbation theory** to express the matrices $\mathbf{R}(\mathbf{u}_k)$
2. Approximate the product up to **first-order** in order to obtain an "**approximate trajectory**" $\{\tilde{\mathbf{u}}_K\}$ as follows:

$$\tilde{\mathbf{u}}_{K+1} := \Pi_{k=0}^{K}\mathbf{A}_k\mathbf{u}_0 - \sum_{r=0}^{K}(\Pi_{k=r+1}^{K}\mathbf{A}_r\mathbf{R}(\mathbf{u}_r)\Pi_{k=0}^{r-1}\mathbf{A}_r)\mathbf{u}_0,$$

where $\mathbf{A}_k := \mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*)$ for all k.

How to confirm whether the approximation is "**tight**"?

- The relative error goes to 0: $\sup_{0\le K\le K_{exit}} \frac{\|\tilde{\mathbf{u}}_K - \mathbf{u}_K\|}{\|\mathbf{u}_K\|} \to 0$ as $\epsilon \to 0$

Proof layout for a linear exit time bound

The final hurdle: The approximate trajectory $\{\tilde{\mathbf{u}}_K\}$ cannot be **uniquely** determined, since it a function of the eigenvalues of the Hessian $\nabla^2 f(\mathbf{x}^*)$

Proof layout for a linear exit time bound

The final hurdle: The approximate trajectory $\{\tilde{\mathbf{u}}_K\}$ cannot be **uniquely** determined, since it a function of the eigenvalues of the Hessian $\nabla^2 f(\mathbf{x}^*)$

Solution

1. Obtain a parametrized family of approximate trajectories for a fixed $\mathbf{u}_0$, denoted by $\{\tilde{\mathbf{u}}_K^\tau\}$, where the parameter $\tau \in \mathbb{R}$

Proof layout for a linear exit time bound

The final hurdle: The approximate trajectory $\{\tilde{\mathbf{u}}_K\}$ cannot be **uniquely** determined, since it a function of the eigenvalues of the Hessian $\nabla^2 f(\mathbf{x}^*)$

Solution

1. Obtain a parametrized family of approximate trajectories for a fixed $\mathbf{u}_0$, denoted by $\{\tilde{\mathbf{u}}_K^\tau\}$, where the parameter $\tau \in \mathbb{R}$
2. Construct the minimal approximate trajectory from this family, defined as one that stays closest to $\mathbf{x}^*$ for each K

Proof layout for a linear exit time bound

The final hurdle: The approximate trajectory $\{\tilde{\mathbf{u}}_K\}$ cannot be **uniquely** determined, since it a function of the eigenvalues of the Hessian $\nabla^2 f(\mathbf{x}^*)$

Solution

1. Obtain a parametrized family of approximate trajectories for a fixed $\mathbf{u}_0$, denoted by $\{\tilde{\mathbf{u}}_K^\tau\}$, where the parameter $\tau \in \mathbb{R}$
2. Construct the minimal approximate trajectory from this family, defined as one that stays closest to $\mathbf{x}^*$ for each K
3. Obtain the smallest upper bound on K of the order $\mathcal{O}(\log(\epsilon^{-1}))$ that satisfies the condition $\inf_\tau \|\tilde{\mathbf{u}}_K^\tau\| > \epsilon$

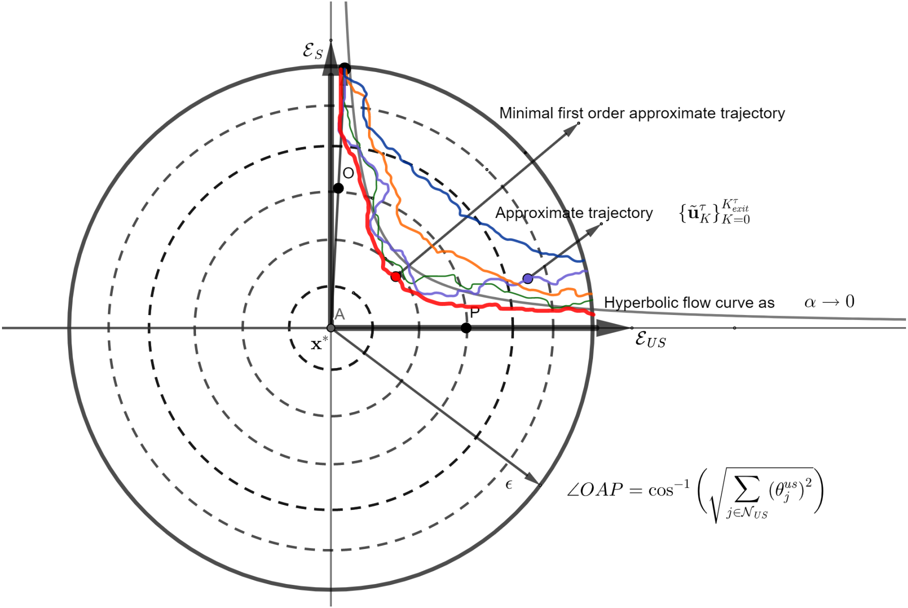
Proof layout for a linear exit time bound

The final hurdle: The approximate trajectory $\{\tilde{\mathbf{u}}_K\}$ cannot be **uniquely** determined, since it a function of the eigenvalues of the Hessian $\nabla^2 f(\mathbf{x}^*)$

Solution

1. Obtain a parametrized family of approximate trajectories for a fixed $\mathbf{u}_0$, denoted by $\{\tilde{\mathbf{u}}_K^\tau\}$, where the parameter $\tau \in \mathbb{R}$
2. Construct the minimal approximate trajectory from this family, defined as one that stays closest to $\mathbf{x}^*$ for each K
3. Obtain the smallest upper bound on K of the order $\mathcal{O}(\log(\epsilon^{-1}))$ that satisfies the condition $\inf_\tau \|\tilde{\mathbf{u}}_K^\tau\| > \epsilon$
4. Derive any necessary and sufficient conditions on $\mathbf{x}_0$ for guaranteeing this linear exit time

A 2-D representation of the approximate trajectories



Outline

Representation of Hessians in the saddle neighborhood

Hessian representation using 'degenerate' matrix perturbation theory

The Hessian $\nabla^2 f(\mathbf{x})$ at any point $\mathbf{x} = \mathbf{x}^* + p\mathbf{u}$, where $p \in [0,1]$ and $\|\mathbf{u}\| \leq \epsilon$, can be represented as

$$\nabla^2 f(\mathbf{x}) = \nabla^2 f(\mathbf{x}^*) + p\,\|\mathbf{u}\|\,\mathbf{H}(\hat{\mathbf{u}}) + \mathcal{O}(\epsilon^2),$$

where $\hat{\mathbf{u}} := \frac{\mathbf{u}}{\|\mathbf{u}\|}$ is the **unit radial vector**, the matrix $\mathbf{H}(\hat{\mathbf{u}})$ is defined as $\mathbf{H}(\hat{\mathbf{u}}) := \frac{d}{dw}(\nabla^2 f(\mathbf{x}^* + w\hat{\mathbf{u}}))|_{w=0}$ and we have that:

$$\mathbf{H}(\hat{\mathbf{u}}) = \sum_{i=1}^{n}\left(\langle \mathbf{v}_i, \mathbf{H}(\hat{\mathbf{u}})\mathbf{v}_i\rangle \mathbf{v}_i\mathbf{v}_i^T + \lambda_i \sum_{l \notin \mathcal{G}_i} \frac{\langle \mathbf{v}_l, \mathbf{H}(\hat{\mathbf{u}})\mathbf{v}_i\rangle}{\lambda_i - \lambda_l}\left(\mathbf{v}_l\mathbf{v}_i^T + \mathbf{v}_i\mathbf{v}_l^T\right)\right)$$

with $\mathcal{G}_i = \{\ j \mid \lambda_j = \lambda_i \pm \mathcal{O}(\epsilon)\}$.

First-order approximation of trajectories

Given an initialization $\mathbf{u}_0$, let $\mathbf{u}_K := \prod_{k=0}^{K-1}\left[\mathbf{A} + \epsilon\mathbf{P}_k\right]\mathbf{u}_0$, where $\{\mathbf{P}_k\}$ are real symmetric matrices and $\mathbf{A}$ is real symmetric and invertible.

Lemma (The 'Approximation Lemma' (Dixit and Bajwa, 2020))

Let $\sup_{0\leq k\leq K-1}\|\mathbf{P}_k\|_2 = \|\mathbf{P}\|_2$ *for some matrix* $\mathbf{P}$, $\epsilon < \left\|\mathbf{A}^{-1}\right\|_2^{-1}\|\mathbf{P}\|_2^{-1}$, *and* $K\epsilon \ll 1$. *We then have the condition:*

$$\left\|\mathbf{A}^{-1}\right\|_2^{-K}\Big(1 - \mathcal{O}(K\epsilon)\Big) \leq |\nu_n| \leq \cdots \leq |\nu_1| \leq \|\mathbf{A}\|_2^K\Big(1 + \mathcal{O}(K\epsilon)\Big),$$

where $\nu_1, \ldots, \nu_n$ *are the eigenvalues of* $\prod_{k=0}^{K-1}\left[\mathbf{A} + \epsilon\mathbf{P}_k\right]$.

In particular, the radial vector trajectory $\mathbf{u}_K$ **can be approximated up to first order in** ϵ **as** $\tilde{\mathbf{u}}_K$ *in this case.*

Lemma (The ϵ-precision trajectory $\{\tilde{\mathbf{u}}_K\}$ (Dixit and Bajwa, 2020))

The dynamical system $\mathbf{u}_K = \prod_{k=0}^{K-1}\Big[\mathbf{A} + \epsilon\mathbf{P}_k\Big]\mathbf{u}_0$ *with the initial condition* $\mathbf{u}_0$ *expressed in terms of the stable and unstable subspaces as* $\mathbf{u}_0 = \epsilon\sum_{i:\mathbf{v}_i\in\mathcal{E}_S}\theta_i^s\mathbf{v}_i + \epsilon\sum_{j:\mathbf{v}_j\in\mathcal{E}_{US}}\theta_j^{us}\mathbf{v}_j$, $\mathbf{A} := \mathbf{I} - \alpha\nabla^2 f(\mathbf{x}^*)$ *and* $\epsilon\mathbf{P}_K := -\frac{\alpha\|\mathbf{u}_K\|}{2}\mathbf{H}(\hat{\mathbf{u}}_K) + \mathcal{O}(\epsilon^2)$ *can be approximated as*

$$\mathbf{u}_K \approx \tilde{\mathbf{u}}_K = \Pi_{k=0}^{K-1}\mathbf{A}\mathbf{u}_0 + \epsilon\sum_{r=0}^{K-1}(\mathbf{A}^{K-1-r}\mathbf{P}_r\mathbf{A}^r)\mathbf{u}_0.$$

First-order approximation of trajectories

Lemma (The ϵ-precision trajectory $\{\tilde{\mathbf{u}}_K\}$ (Dixit and Bajwa, 2020))

The dynamical system $\mathbf{u}_K = \prod_{k=0}^{K-1} \Big[\mathbf{A} + \epsilon \mathbf{P}_k\Big] \mathbf{u}_0$ *with the initial condition* $\mathbf{u}_0$ *expressed in terms of the stable and unstable subspaces as* $\mathbf{u}_0 = \epsilon \sum_{i:\mathbf{v}_i \in \mathcal{E}_S} \theta_i^s \mathbf{v}_i + \epsilon \sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} \theta_j^{us} \mathbf{v}_j$, $\mathbf{A} := \mathbf{I} - \alpha \nabla^2 f(\mathbf{x}^*)$ *and* $\epsilon \mathbf{P}_K := -\frac{\alpha \|\mathbf{u}_K\|}{2} \mathbf{H}(\hat{\mathbf{u}}_K) + \mathcal{O}(\epsilon^2)$ *can be approximated as*

$$\mathbf{u}_K \approx \tilde{\mathbf{u}}_K = \Pi_{k=0}^{K-1} \mathbf{A} \mathbf{u}_0 + \epsilon \sum_{r=0}^{K-1} (\mathbf{A}^{K-1-r} \mathbf{P}_r \mathbf{A}^r) \mathbf{u}_0.$$

Recall: Since the eigenvalues of $\mathbf{A}$ are known only up to an interval, a unique $\tilde{\mathbf{u}}_K$ cannot be obtained. Instead, we get a **family of ϵ- precision trajectories**.

The ‘minimal’ approximate trajectory

Definition (Parametrized approximate trajectories)

We define $S_\epsilon := \left\{ \{\tilde{\mathbf{u}}_K^\tau\}_{K=1}^{K_{exit}^\tau} \middle| \mathbf{u}_0 \right\}$ be the set of τ-parametrized ϵ-**precision trajectories**, with exit times $K_{exit}^\tau := \inf_{K \geq 1} \left\{ K \middle| \left\| \tilde{\mathbf{u}}_K^\tau \right\|^2 > \epsilon^2 \right\}$.

The 'minimal' approximate trajectory

Definition (Parametrized approximate trajectories)

We define $S_\epsilon := \left\{ \{\tilde{\mathbf{u}}_K^\tau\}_{K=1}^{K_{exit}^\tau} \middle| \mathbf{u}_0 \right\}$ be the set of τ-parametrized ϵ-**precision trajectories**, with exit times $K_{exit}^\tau := \inf_{K \geq 1} \left\{ K \middle| \left\| \tilde{\mathbf{u}}_K^\tau \right\|^2 > \epsilon^2 \right\}$.

Definition (The minimal approximate trajectory)

There exists a lower bound on $\left\| \tilde{\mathbf{u}}_K^\tau \right\|^2$ for every K, which we associate with the minimal approximate trajectory. Formally, for $1 \leq K < \sup_\tau \left\{ K_{exit}^\tau \right\}$ we define the bound in terms of a sequence $\Psi(K)$ such that $\epsilon^2 \geq \inf_\tau \left\| \tilde{\mathbf{u}}_K^\tau \right\|^2 > \epsilon^2 \Psi(K)$.

The 'minimal' approximate trajectory

Definition (Parametrized approximate trajectories)

We define $S_\epsilon := \left\{ \{\tilde{\mathbf{u}}_K^\tau\}_{K=1}^{K_{exit}^\tau} \middle| \mathbf{u}_0 \right\}$ be the set of τ-parametrized ϵ-**precision trajectories**, with exit times $K_{exit}^\tau := \inf_{K\geq 1} \left\{ K \middle| \|\tilde{\mathbf{u}}_K^\tau\|^2 > \epsilon^2 \right\}$.

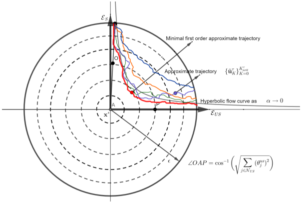
Definition (The minimal approximate trajectory)

There exists a lower bound on $\|\tilde{\mathbf{u}}_K^\tau\|^2$ for every K, which we associate with the minimal approximate trajectory. Formally, for $1 \leq K < \sup_\tau \left\{ K_{exit}^\tau \right\}$ we define the bound in terms of a sequence $\Psi(K)$ such that $\epsilon^2 \geq \inf_\tau \|\tilde{\mathbf{u}}_K^\tau\|^2 > \epsilon^2 \Psi(K)$.

Exit time K^ι for the minimal trajectory

$$K^\iota := \inf_{K\geq 1} \left\{ K \middle| \inf_\tau \left\{ \|\tilde{\mathbf{u}}_K^\tau\|^2 \right\} > \epsilon^2 \right\}$$

Note: $K^\iota \geq \sup_\tau \left\{ K_{exit}^\tau \right\} = \sup_\tau \inf_{K\geq 1} \left\{ K \middle| \|\tilde{\mathbf{u}}_K^\tau\|^2 > \epsilon^2 \right\}$



Characterization of the minimal approximate trajectory

Lemma (The minimal trajectory sequence (Dixit and Bajwa, 2020))

The minimal trajectory sequence $\Psi(K)$*, as a function of the initial radial vector* $\mathbf{u}_0 = \mathbf{x}_0 - \mathbf{x}^*$*, takes the following form:*

$$\Psi(K) = \left(c_1^{2K} - 2Kc_2^{2K-1}b_1 - b_2c_3^K c_2^K - b_2c_3^{2K}\right) \sum_{i:\mathbf{v}_i \in \mathcal{E}_S} (\theta_i^s)^2 +$$

$$\left(c_4^{2K} - 2Kc_3^{2K-1}b_1 - b_2c_3^K c_2^K - b_2c_3^{2K}\right) \sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2,$$

with the constants defined as $c_1 = 1 - \alpha L - \mathcal{O}(\epsilon)$, $c_2 = 1 - \alpha\beta + \mathcal{O}(\epsilon)$, $c_3 = 1 + \alpha L + \mathcal{O}(\epsilon)$, $c_4 = 1 + \alpha\beta - \mathcal{O}(\epsilon)$, $b_1 = \frac{\alpha\epsilon MLn}{2\delta} + \mathcal{O}(\epsilon^2)$*, and* $b_2 = \frac{(\frac{\alpha\epsilon MLn}{2\delta} + \mathcal{O}(\epsilon^2))(1+\mathcal{O}(K\epsilon))}{(\alpha L + \alpha\beta + \mathcal{O}(\epsilon^2))}$.

Existence of GD trajectories with linear exit times

Theorem ('Fast' escape of GD trajectories (Dixit and Bajwa, 2020))

For gradient descent with $\alpha = \frac{1}{L}$ *on a* **well-conditioned function, i.e.,** $\frac{\beta}{L} > \frac{\epsilon M}{2L}$, *and some* **minimum projection** $\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 \geq \Delta$ **of the initial radial vector** u_0 **on the unstable subspace** $\mathcal{E}_{US}$, *there exist* ϵ*-precision trajectories* $\{\tilde{\mathbf{u}}_k\}_{k=1}^{K_{exit}}$ *with* **linear exit time** *such that*

$$K_{exit} < K^{\iota} \lessapprox \frac{\log\left(\left(2 + \frac{\epsilon M}{2L}\right)\log\left(\frac{2+\frac{\epsilon M}{2L}}{1+\frac{\beta}{L}-\frac{\epsilon M}{2L}}\right)\frac{2\delta}{\epsilon M n}\right)}{2\log\left(\frac{2+\frac{\epsilon M}{2L}}{1+\frac{\beta}{L}-\frac{\epsilon M}{2L}}\right)}.$$

Existence of GD trajectories with linear exit times

Theorem ('Fast' escape of GD trajectories (Dixit and Bajwa, 2020))

For gradient descent with $\alpha = \frac{1}{L}$ *on a* **well-conditioned function, i.e.,** $\frac{\beta}{L} > \frac{\epsilon M}{2L}$, *and some* **minimum projection** $\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 \geq \Delta$ **of the initial radial vector** $\mathbf{u}_0$ **on the unstable subspace** $\mathcal{E}_{US}$, *there exist* ϵ*-precision trajectories* $\{\tilde{\mathbf{u}}_k\}_{k=1}^{K_{exit}}$ *with* **linear exit time** *such that*

$$K_{exit} < K^{\iota} \lessapprox \frac{\log\left(\left(2 + \frac{\epsilon M}{2L}\right)\log\left(\frac{2+\frac{\epsilon M}{2L}}{1+\frac{\beta}{L}-\frac{\epsilon M}{2L}}\right)\frac{2\delta}{\epsilon M n}\right)}{2\log\left(\frac{2+\frac{\epsilon M}{2L}}{1+\frac{\beta}{L}-\frac{\epsilon M}{2L}}\right)}.$$

Necessary initial condition for linear exit time

For the above bound to hold, we must have $\Delta > \epsilon \frac{MLn}{\delta(L+\beta)} = \mathcal{O}(\epsilon)$ for some sufficiently small ϵ.

Bound on the neighborhood size ϵ

Step size: $\alpha = \frac{1}{L}$

The linear exit time bound requires that $K\epsilon \ll 1$ and

$$\epsilon < \min\left\{ \inf_{\|\mathbf{u}\|=1} \left(\limsup_{j\to\infty} \sqrt[j]{\frac{r_j(\mathbf{u})}{j!}} \right)^{-1}, \frac{2L\delta}{M(2Ln^2 - \delta)} + \mathcal{O}(\epsilon^2) \right\},$$

where $r_j(\mathbf{u}) := \left\| \left(\frac{d^j}{dw^j} \nabla^2 f(\mathbf{x}^* + w\mathbf{u}) \Big|_{w=0} \right) \right\|_2$.

Bound on the neighborhood size ϵ

Step size: $\alpha = \frac{1}{L}$

The linear exit time bound requires that $K\epsilon \ll 1$ and

$$\epsilon < \min\left\{ \inf_{\|\mathbf{u}\|=1} \left(\limsup_{j\to\infty} \sqrt[j]{\frac{r_j(\mathbf{u})}{j!}} \right)^{-1}, \frac{2L\delta}{M(2Ln^2-\delta)} + \mathcal{O}(\epsilon^2) \right\},$$

where $r_j(\mathbf{u}) := \left\| \left(\frac{d^j}{dw^j} \nabla^2 f(\mathbf{x}^* + w\mathbf{u}) \Big|_{w=0} \right) \right\|_2$.

Remark

The term $\mathcal{O}(\epsilon^2)$ appearing on the R.H.S. of the upper bound of ϵ only implies a bounded uncertainty term that will go to 0 faster than ϵ goes to 0 for sufficiently small ϵ.

Approximate trajectories: Tightness of the approximation

Lemma (Bound on the relative error (Dixit and Bajwa, 2021))

The relative error of the approximate trajectories is upper bounded as

$$\sup_{0\leq K\leq K_{exit}} \frac{\|\mathbf{u}_K - \tilde{\mathbf{u}}_K\|}{\|\mathbf{u}_K\|} \leq \frac{\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\left(\log\left(\frac{1}{\epsilon}\right)\epsilon\right)^2\right)}{\sqrt{\sum_{j:\mathbf{v}_j\in\mathcal{E}_{US}}(\theta_j^{us})^2} - \mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\left(\log\left(\frac{1}{\epsilon}\right)\epsilon\right)\right)},$$

which goes to 0 *as* $\epsilon \to 0$.

Approximate trajectories: Tightness of the approximation

Lemma (Bound on the relative error (Dixit and Bajwa, 2021))

The relative error of the approximate trajectories is upper bounded as

$$\sup_{0 \leq K \leq K_{exit}} \frac{\|\mathbf{u}_K - \tilde{\mathbf{u}}_K\|}{\|\mathbf{u}_K\|} \leq \frac{\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\left(\log\left(\frac{1}{\epsilon}\right)\epsilon\right)^2\right)}{\sqrt{\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2} - \mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\left(\log\left(\frac{1}{\epsilon}\right)\epsilon\right)\right)},$$

which goes to 0 *as* $\epsilon \to 0$*.*

Necessary condition for bounded relative error

The initial projection on the unstable subspace must satisfy

$$\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 > \mathcal{O}\left(\left(\log\left(\frac{1}{\epsilon}\right)\right)^2 \epsilon\right).$$

Sufficient condition for linear exit time trajectories

Theorem (Sufficient unstable projection (Dixit and Bajwa, 2021))

A gradient descent trajectory is guaranteed to have linear exit time whenever the function is well-conditioned with $\frac{\beta}{L} > \frac{\epsilon M}{2L}$ *and the projection of the initial vector* $\mathbf{u}_0$ *on the unstable subspace satisfies*

$$\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 \gtrsim \frac{\left(2 + \frac{\epsilon M}{2L}\right)\left(\frac{2\delta\mu \log\left(1+\frac{\beta}{L}-\frac{\epsilon M}{2L}\right)}{Mn}\right)}{\frac{1}{a}\log\left(\frac{1}{\epsilon \sqrt[a]{\mu}}\right)+1} = \mathcal{O}\left(\frac{1}{\log(\epsilon^{-1})}\right)$$

with $a = \frac{\log\left(2+\frac{\epsilon M}{2L}\right)}{c}$, $c = \log\left(\frac{2+\frac{\epsilon M}{2L}}{1+\frac{\beta}{L}-\frac{\epsilon M}{2L}}\right)$, *and* $\sqrt[a]{\mu} = \frac{Mn\log\left(2+\frac{\epsilon M}{2L}\right)}{2c\delta\left(2+\frac{\epsilon M}{2L}\right)\log\left(1+\frac{\beta}{L}-\frac{\epsilon M}{2L}\right)}$.

Outline

Linear time saddle escape: From theory to practice

It is already known that gradient descent trajectories almost surely escape from strict saddle neighborhoods (Lee et al., 2017). **But how can it be made to follow the trajectory that escapes in linear time?**

Linear time saddle escape: From theory to practice

It is already known that gradient descent trajectories almost surely escape from strict saddle neighborhoods (Lee et al., 2017). **But how can it be made to follow the trajectory that escapes in linear time?**

Theory: A GD trajectory with $\mathbf{u}_0 = \epsilon \sum_{i:\mathbf{v}_i \in \mathcal{E}_S} \theta_i^s \mathbf{v}_i + \epsilon \sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} \theta_j^{us} \mathbf{v}_j$ that satisfies the **sufficient condition**

$$\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 \gtrapprox \mathcal{O}\left(\frac{1}{\log(\epsilon^{-1})}\right)$$

will approximately exit the saddle neighborhood $\mathcal{B}_\epsilon(\mathbf{x}^*)$ in linear time.

Linear time saddle escape: From theory to practice

It is already known that gradient descent trajectories almost surely escape from strict saddle neighborhoods (Lee et al., 2017). **But how can it be made to follow the trajectory that escapes in linear time?**

Theory: A GD trajectory with $\mathbf{u}_0 = \epsilon \sum_{i:\mathbf{v}_i \in \mathcal{E}_S} \theta_i^s \mathbf{v}_i + \epsilon \sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} \theta_j^{us} \mathbf{v}_j$ that satisfies the **sufficient condition**

$$\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 \gtrapprox \mathcal{O}\left(\frac{1}{\log(\epsilon^{-1})}\right)$$

will approximately exit the saddle neighborhood $\mathcal{B}_\epsilon(\mathbf{x}^*)$ in linear time.

How to check if the sufficient condition is satisfied by $\mathbf{u}_0$?

Linear time saddle escape: From theory to practice

It is already known that gradient descent trajectories almost surely escape from strict saddle neighborhoods (Lee et al., 2017). **But how can it be made to follow the trajectory that escapes in linear time?**

Theory: A GD trajectory with $\mathbf{u}_0 = \epsilon \sum_{i:\mathbf{v}_i \in \mathcal{E}_S} \theta_i^s \mathbf{v}_i + \epsilon \sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} \theta_j^{us} \mathbf{v}_j$ that satisfies the **sufficient condition**

$$\sum_{j:\mathbf{v}_j \in \mathcal{E}_{US}} (\theta_j^{us})^2 \gtrapprox \mathcal{O}\left(\frac{1}{\log(\epsilon^{-1})}\right)$$

will approximately exit the saddle neighborhood $\mathcal{B}_\epsilon(\mathbf{x}^*)$ in linear time.

How to check if the sufficient condition is satisfied by $\mathbf{u}_0$?

- Estimate the negative curvature using consecutive gradient difference
- **Intuition:** The gradient difference **approximates** the column space of a Hessian, thereby helping estimate the curvature

A robust check for the sufficient condition

Assume $\mathbf{x}_k$ is in a strict saddle neighborhood of the nonconvex function and fix the gradient descent step size to be $\alpha = \frac{1}{L}$

A robust check for the sufficient condition

Assume $\mathbf{x}_k$ is in a strict saddle neighborhood of the nonconvex function and fix the gradient descent step size to be $\alpha = \frac{1}{L}$

1. **Set** $\mathbf{y}_0 = \mathbf{x}_k$ and $\mathbf{y}_1 = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$

A robust check for the sufficient condition

Assume $\mathbf{x}_k$ is in a strict saddle neighborhood of the nonconvex function and fix the gradient descent step size to be $\alpha = \frac{1}{L}$

1. **Set** $\mathbf{y}_0 = \mathbf{x}_k$ and $\mathbf{y}_1 = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$
2. **Compute** $V_1 = \|\mathbf{y}_1 - \mathbf{y}_0\|^2$ and $V_2 = \frac{1}{L}\langle \mathbf{y}_1 - \mathbf{y}_0, \nabla f(\mathbf{y}_1) - \nabla f(\mathbf{y}_0)\rangle$

A robust check for the sufficient condition

Assume $\mathbf{x}_k$ is in a strict saddle neighborhood of the nonconvex function and fix the gradient descent step size to be $\alpha = \frac{1}{L}$

1. **Set** $\mathbf{y}_0 = \mathbf{x}_k$ and $\mathbf{y}_1 = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$
2. **Compute** $V_1 = \|\mathbf{y}_1 - \mathbf{y}_0\|^2$ and $V_2 = \frac{1}{L}\langle \mathbf{y}_1 - \mathbf{y}_0, \nabla f(\mathbf{y}_1) - \nabla f(\mathbf{y}_0)\rangle$
3. **Set** $P_{min}(\epsilon) = \dfrac{\left(2+\frac{\epsilon M}{2L}\right)\left(\frac{2\delta\mu \log\left(1+\frac{\beta}{L}-\frac{\epsilon M}{2L}\right)}{Mn}\right)}{\frac{1}{a}\log\left(\frac{1}{\epsilon \sqrt[a]{\mu}}\right)+1}$ **(sufficient condition)**

A robust check for the sufficient condition

Assume $\mathbf{x}_k$ is in a strict saddle neighborhood of the nonconvex function and fix the gradient descent step size to be $\alpha = \frac{1}{L}$

1. **Set** $\mathbf{y}_0 = \mathbf{x}_k$ and $\mathbf{y}_1 = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$
2. **Compute** $V_1 = \|\mathbf{y}_1 - \mathbf{y}_0\|^2$ and $V_2 = \frac{1}{L}\langle \mathbf{y}_1 - \mathbf{y}_0, \nabla f(\mathbf{y}_1) - \nabla f(\mathbf{y}_0)\rangle$
3. **Set** $P_{min}(\epsilon) = \dfrac{\left(2+\frac{\epsilon M}{2L}\right)\left(\frac{2\delta\mu \log\left(1+\frac{\beta}{L}-\frac{\epsilon M}{2L}\right)}{Mn}\right)}{\frac{1}{a}\log\left(\frac{1}{\epsilon \sqrt[a]{\mu}}\right)+1}$ (**sufficient condition**)
4. **IF** $V_1 - V_2 > \left(\frac{50P_{min}(\epsilon)+4}{27}\right)\frac{L^2\epsilon^2}{\beta^2}$ then GD will escape in **linear time**

A robust check for the sufficient condition

Assume $\mathbf{x}_k$ is in a strict saddle neighborhood of the nonconvex function and fix the gradient descent step size to be $\alpha = \frac{1}{L}$

1. **Set** $\mathbf{y}_0 = \mathbf{x}_k$ and $\mathbf{y}_1 = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$
2. **Compute** $V_1 = \|\mathbf{y}_1 - \mathbf{y}_0\|^2$ and $V_2 = \frac{1}{L}\langle \mathbf{y}_1 - \mathbf{y}_0, \nabla f(\mathbf{y}_1) - \nabla f(\mathbf{y}_0)\rangle$
3. **Set** $P_{min}(\epsilon) = \dfrac{\left(2+\frac{\epsilon M}{2L}\right)\left(\frac{2\delta\mu \log\left(1+\frac{\beta}{L}-\frac{\epsilon M}{2L}\right)}{Mn}\right)}{\frac{1}{a}\log\left(\frac{1}{\epsilon \sqrt[a]{\mu}}\right)+1}$ (**sufficient condition**)
4. **IF** $V_1 - V_2 > \left(\frac{50P_{min}(\epsilon)+4}{27}\right)\frac{L^2\epsilon^2}{\beta^2}$ then GD will escape in **linear time**

Check fails: *Either* the sufficient condition is not being met *or* the iterate $\mathbf{x}_k$ is already near a **local minimum**

Algorithm CCRGD (Dixit and Bajwa, 2021)

1: **Initialize** $\mathbf{x}_0$ randomly, $\alpha = \frac{1}{L}$, $P_{min}(\epsilon)$, and condition flag $\Xi = 0$
2: **for** $k = 1$ to K_{max} **do**
3:     **If** $\|\nabla f(\mathbf{x}_k)\| > L\epsilon$ **then**
4:         **Update** $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$
        If $\Xi = 1$ **then update condition flag** $\Xi \leftarrow 0$
5:     **Else**
        If $\|\nabla f(\mathbf{x}_k)\| \leq L\epsilon$ **and** $\Xi = 1$ **then**
            Update $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$
        Else If $\|\nabla f(\mathbf{x}_k)\| \leq L\epsilon$ **and** $\Xi = 0$ **then**
            If robust check condition satisfied then
                Update condition flag $\Xi \leftarrow 1$ **and continue**
            Else Call a single-step subroutine
6: **end for**
7: **Return** $\mathbf{x}_k$

Choices of subroutines for CCRGD

Subroutine 1 (Guarantees linear exit time trajectory)

Algorithm Constrained eigenvalue problem (Dixit and Bajwa, 2021)

1: **Get** $\mathbf{x}_{k+1} \in \arg\min_{\|\mathbf{x}-\mathbf{x}_k\| = \frac{\|\nabla f(\mathbf{x}_k)\|}{\beta}} \langle (\mathbf{x} - \mathbf{x}_k), \nabla^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) \rangle$

2: **Update condition flag** $\Xi \leftarrow 1$

3: **IF** $\langle (\mathbf{x}_{k+1} - \mathbf{x}_k), \nabla^2 f(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) \rangle \geq 0$ then **break** from CCRGD

Choices of subroutines for CCRGD

Subroutine 1 (Guarantees linear exit time trajectory)

Algorithm Constrained eigenvalue problem (Dixit and Bajwa, 2021)

1: **Get** $\mathbf{x}_{k+1} \in \arg\min_{\|\mathbf{x}-\mathbf{x}_k\|=\frac{\|\nabla f(\mathbf{x}_k)\|}{\beta}} \langle (\mathbf{x}-\mathbf{x}_k), \nabla^2 f(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)\rangle$

2: **Update condition flag** $\Xi \leftarrow 1$

3: **IF** $\langle (\mathbf{x}_{k+1}-\mathbf{x}_k), \nabla^2 f(\mathbf{x}_k)(\mathbf{x}_{k+1}-\mathbf{x}_k)\rangle \geq 0$ then **break** from CCRGD

Subroutine 2 (Fast probabilistic escape; may not give linear exit time)

Algorithm Perturbed GD (Du et al., 2017; Jin, Ge, et al., 2017)

1: **Update** $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \zeta_k$ **with** ζ_k **uniformly** $\sim \mathbb{B}_{\mathbf{0}}(r)$ *for some* r

2: **Update condition flag** $\Xi \leftarrow 1$

Outline

Characterizing GD trajectories after the fast saddle escape

The CCRGD algorithm can exit sufficiently small saddle neighborhoods at a linear rate. **BUT ...**

- The function outside a saddle neighborhood $\mathcal{B}_\epsilon(\mathbf{x}^*)$ is still nonconvex
- Since CCRGD reverts back to GD after the escape, traditional analytical approaches only yield rates of $\mathcal{O}(\eta^{-2})$ for convergence of CCRGD to the η-neighborhood of a local minimum

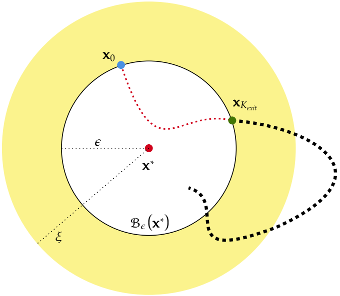
Characterizing GD trajectories after the fast saddle escape

The CCRGD algorithm can exit sufficiently small saddle neighborhoods at a linear rate. **BUT ...**

- The function outside a saddle neighborhood $\mathcal{B}_\epsilon(\mathbf{x}^*)$ is still nonconvex
- Since CCRGD reverts back to GD after the escape, traditional analytical approaches only yield rates of $\mathcal{O}(\eta^{-2})$ for convergence of CCRGD to the η-neighborhood of a local minimum

In order to improve on the $\mathcal{O}(\eta^{-2})$ rate, we need to be able to address the following questions:

1. How do the GD trajectories behave outside the small saddle neighborhood $\mathcal{B}_\epsilon(\mathbf{x}^*)$?
   - **Challenge:** Matrix perturbation theory does not hold outside $\mathcal{B}_\epsilon(\mathbf{x}^*)$
2. What is the guarantee that a trajectory, after escaping $\mathcal{B}_\epsilon(\mathbf{x}^*)$ and/or its augmentation, does not return to the same region?



The 'sequential monotonicity' property of GD trajectories

Lemma (Sequential monotonicity (Dixit and Bajwa, 2021))

Let $\xi < \frac{1}{\varsigma M}\sqrt{\left(\frac{(1+\frac{\beta}{L})^2}{2}\left(1-\left(1-\frac{\beta}{L}\right)^2\right)-1\right)}$ *for some* $\varsigma > 2$, *take* $\alpha = \frac{1}{L}$, *and assume a well-conditioned function. Next, consider the tuple* $(\mathbf{x}, \mathbf{x}^{+}, \mathbf{x}^{++})$ *such that* $\|\mathbf{x}^{+} - \mathbf{x}^{*}\| \geq \|\mathbf{x} - \mathbf{x}^{*}\|$ *and* $\|\mathbf{x} - \mathbf{x}^{*}\| < \xi$. *Then:*

a. $\left\|\mathbf{x}^{++} - \mathbf{x}^{*}\right\| > \left\|\mathbf{x}^{+} - \mathbf{x}^{*}\right\|,$ *and*

b. $\left\|\mathbf{x}^{++} - \mathbf{x}^{*}\right\| \geq \bar{\rho}(\mathbf{x})\left\|\mathbf{x}^{+} - \mathbf{x}^{*}\right\| - \sigma(\mathbf{x}),$

where $\sigma(\mathbf{x}) = \mathcal{O}(\|\mathbf{x} - \mathbf{x}^{*}\|^2)$ *and* $\bar{\rho}(\mathbf{x}) > 1$.

The 'sequential monotonicity' property of GD trajectories

Lemma (Sequential monotonicity (Dixit and Bajwa, 2021))

Let $\xi < \frac{1}{\varsigma M}\sqrt{\left(\frac{(1+\frac{\beta}{L})^2}{2}\left(1-\left(1-\frac{\beta}{L}\right)^2\right)-1\right)}$ *for some* $\varsigma > 2$, *take* $\alpha = \frac{1}{L}$, *and assume a well-conditioned function. Next, consider the tuple* $(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^{++})$ *such that* $\|\mathbf{x}^+ - \mathbf{x}^*\| \geq \|\mathbf{x} - \mathbf{x}^*\|$ *and* $\|\mathbf{x} - \mathbf{x}^*\| < \xi$. *Then:*

a. $\left\|\mathbf{x}^{++} - \mathbf{x}^*\right\| > \left\|\mathbf{x}^+ - \mathbf{x}^*\right\|,$ *and*

b. $\left\|\mathbf{x}^{++} - \mathbf{x}^*\right\| \geq \bar{\rho}(\mathbf{x})\left\|\mathbf{x}^+ - \mathbf{x}^*\right\| - \sigma(\mathbf{x}),$

where $\sigma(\mathbf{x}) = \mathcal{O}(\|\mathbf{x} - \mathbf{x}^*\|^2)$ *and* $\bar{\rho}(\mathbf{x}) > 1$.

The sequential monotonicity property in words

If a gradient descent trajectory with respect to a strict saddle $\mathbf{x}^*$ has non-contractive dynamics at any iteration, then it has expansive dynamics for all subsequent iterations as long as the trajectory stays inside $\mathcal{B}_\xi(\mathbf{x}^*)$.

Implications of the sequential monotonicity property

Note: While the exit time analysis relies on **local** analyticity of $f(\cdot)$ around $\mathbf{x}^*$, the sequential monotonicity property only requires the function to be twice continuously differentiable

Implications

- The property can be utilized to provide rates of convergence to / divergence from $\mathbf{x}^*$ in an **augmented neighborhood** $\mathcal{B}_\xi(\mathbf{x}^*) \supset \mathcal{B}_\epsilon(\mathbf{x}^*)$
- Any rates obtained in this manner would be **exact**, since we no longer rely on matrix perturbation analysis

Implications of the sequential monotonicity property

Note: While the exit time analysis relies on **local** analyticity of $f(\cdot)$ around $\mathbf{x}^*$, the sequential monotonicity property only requires the function to be twice continuously differentiable

Implications

- The property can be utilized to provide rates of convergence to / divergence from $\mathbf{x}^*$ in an **augmented neighborhood** $\mathcal{B}_\xi(\mathbf{x}^*) \supset \mathcal{B}_\epsilon(\mathbf{x}^*)$
- Any rates obtained in this manner would be **exact**, since we no longer rely on matrix perturbation analysis

Roadmap for convergence analysis: In order to develop rates in an augmented neighborhood $\mathcal{B}_\xi(\mathbf{x}^*)$ of $\mathbf{x}^*$, we can utilize/derive:

- Exit time bounds in some **small neighborhood** $\mathcal{B}_\epsilon(\mathbf{x}^*) \subset \mathcal{B}_\xi(\mathbf{x}^*)$ ✓
- Travel time in the **shell** $\bar{\mathcal{B}}_\xi(\mathbf{x}^*) \backslash \mathcal{B}_\epsilon(\mathbf{x}^*)$ using the monotonicity property

Sojourn time inside the shell $\bar{\mathcal{B}}_\xi(\mathbf{x}^*)\backslash\mathcal{B}_\epsilon(\mathbf{x}^*)$

Definitions of different trajectory times

- $\hat{K}_{exit}$: **First** exit time of the gradient descent trajectory from $\mathcal{B}_\xi(\mathbf{x}^*)$.
- K_c: **Last** time when the trajectory is contracting inside the shell
- K_e: **First** time when the trajectory starts expanding inside the shell

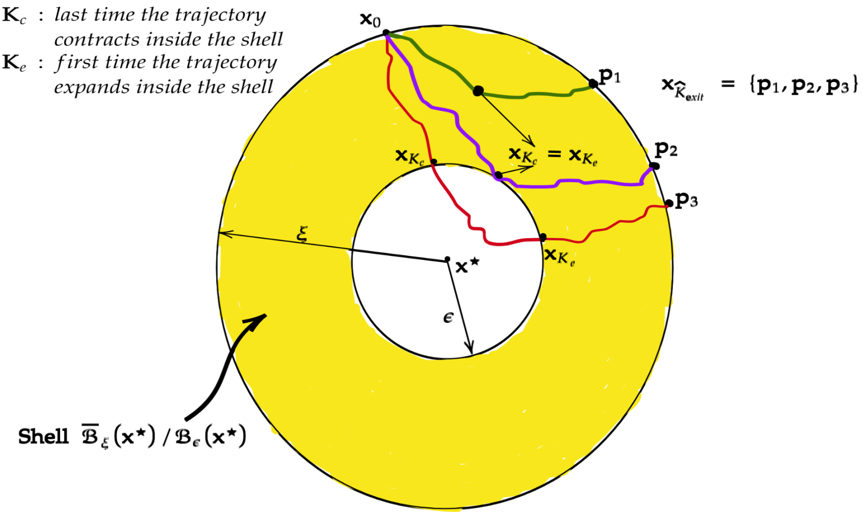
Theorem (Shell travel time (Dixit and Bajwa, 2021))

The **sojourn time** $K_{shell} = \hat{K}_{exit} + K_c - K_e$ *for a gradient descent trajectory inside the compact shell* $\bar{\mathcal{B}}_\xi(\mathbf{x}^*)\backslash\mathcal{B}_\epsilon(\mathbf{x}^*)$ *has the following order:*

$$K_{shell} = \mathcal{O}\left(\log\left(\frac{a}{f(\mathbf{x}_{K_c}) - f(\mathbf{x}^*) - b} \right) \right) + \mathcal{O}\left(\log\left(\frac{\xi}{\epsilon} \right) \right) + \mathcal{O}(1),$$

where a, b *are some positive constants with* $f(\mathbf{x}_{K_c}) - f(\mathbf{x}^*) > b$.

A 2-D representation of 3 possible trajectories



The 'no return' guarantees

So far, the theorems have only provided "first exit time" bounds, but the gradient descent trajectory can possibly re-enter the neighborhood it just escaped!

The 'no return' guarantees

So far, the theorems have only provided "first exit time" bounds, but the gradient descent trajectory can possibly re-enter the neighborhood it just escaped!

Lemma (No return to small neighborhoods (Dixit and Bajwa, 2021))

For well-conditioned problems, i.e., $\mathcal{O}\left(\frac{\sqrt{2}}{\sqrt{\log_2(\frac{1}{\epsilon})}}\right) < \frac{\beta}{L} \leq 1$, *where* ϵ *is upper bounded from the exit time theorem, a gradient descent trajectory having exited the ball* $\mathcal{B}_\epsilon(\mathbf{x}^*)$ *can never re-enter it.*

The 'no return' guarantees

So far, the theorems have only provided "first exit time" bounds, but the gradient descent trajectory can possibly re-enter the neighborhood it just escaped!

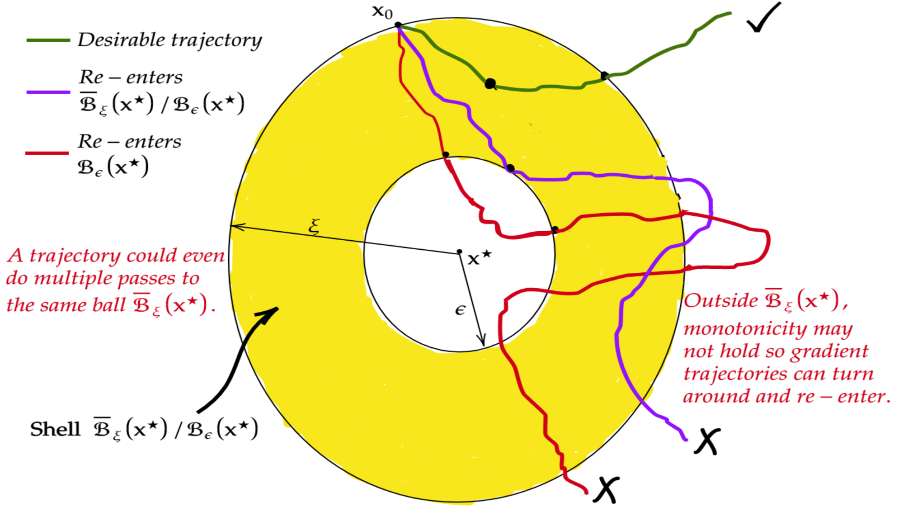
Lemma (No return to small neighborhoods (Dixit and Bajwa, 2021))

For well-conditioned problems, i.e., $\mathcal{O}\left(\frac{\sqrt{2}}{\sqrt{\log_2(\frac{1}{\epsilon})}}\right) < \frac{\beta}{L} \leq 1$*, where* ϵ *is upper bounded from the exit time theorem, a gradient descent trajectory having exited the ball* $\mathcal{B}_\epsilon(\mathbf{x}^*)$ *can never re-enter it.*

Lemma (No return to large neighborhoods (Dixit and Bajwa, 2021))

The gradient descent trajectories exiting the ball $\mathcal{B}_\xi(\mathbf{x}^*)$ *can never re-enter it, provided* *(i)* ξ *is bounded as in the sequential monotonicity lemma with* $\varsigma \geq 47$*, (ii) the function is well conditioned inside* $\mathcal{B}_\xi(\mathbf{x}^*)$*, and (iii) the gradient magnitudes outside* $\mathcal{B}_\xi(\mathbf{x}^*)$ *are sufficiently large with* $\|\nabla f(\mathbf{x})\| \geq \gamma > \frac{1}{\sqrt{2}} L\xi$*.*

Significance of the no-return guarantees



Convergence rate: Assumptions on the global landscape

1. **Minimum separation of stationary points:** Let $\mathcal{S}_*$ be the set of all first-order stationary points of $f(\cdot)$ in some compact domain $\mathcal{U}$. The distance between any two stationary points of $f(\cdot)$ in $\mathcal{U}$ is lower bounded by $R > 0$ and we have that $R > 2\xi$.

Convergence rate: Assumptions on the global landscape

1. **Minimum separation of stationary points:** Let $\mathcal{S}_*$ be the set of all first-order stationary points of $f(\cdot)$ in some compact domain $\mathcal{U}$. The distance between any two stationary points of $f(\cdot)$ in $\mathcal{U}$ is lower bounded by $R > 0$ and we have that $R > 2\xi$.

2. **Initialization and convergence within the compact domain:** Let $\mathbf{x}_0$ be the initialization point and the sequence $\{\mathbf{x}_k\}$ generated by CCRGD converges to the minimum $\mathbf{x}^*_{optimal} \in \mathcal{S}_*$, where $\left\|\mathbf{x}_0 - \mathbf{x}^*_{optimal}\right\| \leq \zeta$ and $R < \zeta < lR$.

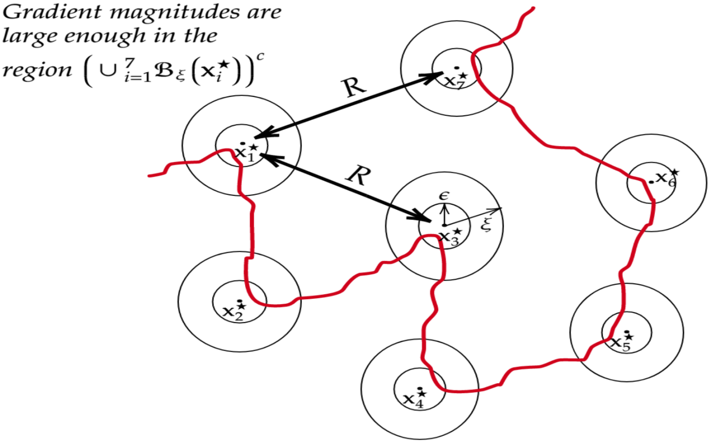
Convergence rate: Assumptions on the global landscape

1. **Minimum separation of stationary points:** Let $\mathcal{S}_*$ be the set of all first-order stationary points of $f(\cdot)$ in some compact domain $\mathcal{U}$. The distance between any two stationary points of $f(\cdot)$ in $\mathcal{U}$ is lower bounded by $R > 0$ and we have that $R > 2\xi$.

2. **Initialization and convergence within the compact domain:** Let $\mathbf{x}_0$ be the initialization point and the sequence $\{\mathbf{x}_k\}$ generated by CCRGD converges to the minimum $\mathbf{x}^*_{optimal} \in \mathcal{S}_*$, where $\left\|\mathbf{x}_0 - \mathbf{x}^*_{optimal}\right\| \leq \zeta$ and $R < \zeta < lR$.

3. **Boundedness of gradient magnitudes:** The gradient magnitude for any $\mathbf{x} \in \mathcal{U} \setminus \bigcup_{j=1}^{l} \bar{\mathcal{B}}_\xi(\mathbf{x}^*_j)$ is lower bounded as $\|\nabla f(\mathbf{x})\| \geq \gamma > \frac{1}{\sqrt{2}} L\xi$. In addition, compactness of $\mathcal{U}$ also implies $\|\nabla f(\mathbf{x})\| \leq \Gamma$ for any $\mathbf{x} \in \mathcal{U}$.

Gradient descent trajectory traversing cascaded saddles



A 2 − D hexagonal lattice of saddle point neighborhoods

Convergence rate of CCRGD to a local minimum

Theorem (Convergence rate of CCRGD (Dixit and Bajwa, 2021))

Suppose $\mathbf{x}_0 \in \mathcal{B}_\xi(\mathbf{x}_0^*)$ *for a strict saddle* $\mathbf{x}_0^* \in \mathcal{S}_*$*, and let* $\mathcal{Y} = \{\mathcal{B}_\xi(\mathbf{x}_i^*)\}_{i=0}^{\mathcal{Q}}$ *be an ordered sequence of cascaded saddle neighborhoods traversed by the trajectory* $\{\mathbf{x}_k\}$*. Then, defining* $\bar{K}_{shell} := \max_{x^* \in \mathcal{Y}} K_{shell}(x^*)$*, the total time* K_{max} *for the trajectory to reach an* ϵ*-neighborhood of the local minimum* $\mathbf{x}^*_{optimal}$ *satisfies:*

$$K_{max} < \left(\frac{4R_{eff}}{R}\right)^n \left((\underbrace{K_{exit}}_{1} + \underbrace{\bar{K}_{shell}}_{2}) + \underbrace{\frac{2L}{\gamma^2}\left(\Gamma + \frac{L}{2}\mathit{diam}(\mathcal{U})\right)(\hat{R} + \xi)}_{3} \right)$$

where $R_{eff} = R_\omega(\zeta)$*,* $\hat{R} = R_\omega(R)$ *and the function* $R_\omega(\cdot)$ *is bounded as:*

$$R_\omega(z) \leq z + 2\left(\Gamma + \frac{L}{2}\mathit{diam}(\mathcal{U})\right)\frac{z}{\gamma} + N_0(z)K_{exit}\left(\frac{1}{\beta} + \frac{L}{2\beta^2}\right)\frac{L^2\epsilon^2}{\gamma} + N_0(z)(K_{exit} + \bar{K}_{shell})\xi.$$

Outline

Minimization of a modified Rastrigin function

Optimization problem

The problem corresponds to minimization of a modified Rastrigin function:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) := \sum_{i=1}^{n} a_i \cos(b_i x_i),$$

which differs from the standard Rastrigin function in the sense that this modified function does not have the quadratic terms added to it.

Minimization of a modified Rastrigin function

Optimization problem

The problem corresponds to minimization of a modified Rastrigin function:

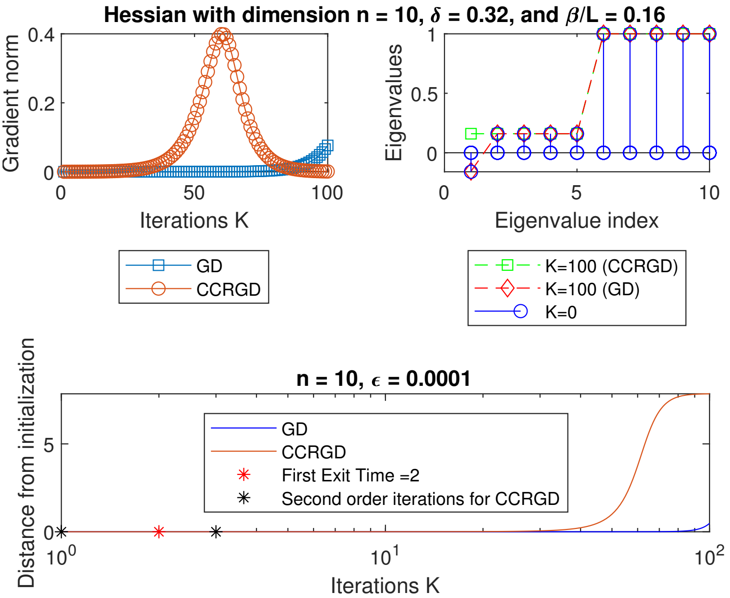
$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) := \sum_{i=1}^{n} a_i \cos(b_i x_i),$$

which differs from the standard Rastrigin function in the sense that this modified function does not have the quadratic terms added to it.

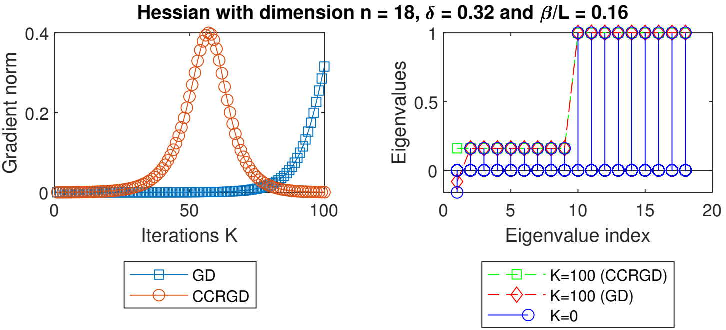
Numerical setup

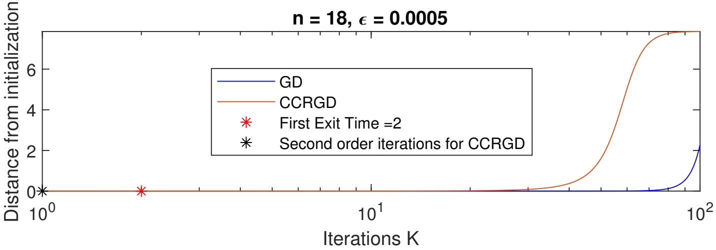
- **Set** $a_i = 1$ for $i = 1$ and $a_i = -1$ elsewhere; **Set** $b_i = 1$ for $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$ and $b_i = 0.4$ for $\lfloor \frac{n}{2} \rfloor + 1 \leq i \leq n$
  - The point $\mathbf{x}^* = \mathbf{0}$ is a strict saddle point for this problem
- **Initialization:** The iterate $\mathbf{x}_0$ is initialized in an ϵ neighborhood of the strict saddle point $\mathbf{x}^*$ with a very small unstable subspace projection

Convergence plots: $n = 10$



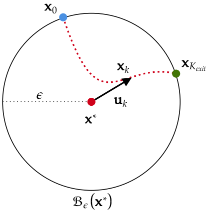
Convergence plots: $n = 18$





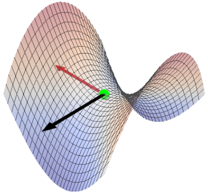
Concluding Remarks

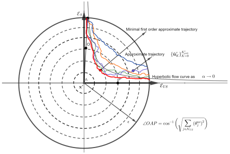
While first-order methods almost surely avoid strict saddle neighborhoods, ensuring they escape the saddle in linear time requires a handle on discrete trajectories



Developments presented in this talk

- A matrix perturbation-based analytical approach that helps characterize the behavior of discrete trajectories in small saddle neighborhoods
- A sufficient condition on the unstable subspace projection of the initialization for linear exit time
- An analysis of discrete trajectories within the shells surrounding saddle neighborhoods
- A gradient descent-based algorithm, and its convergence analysis to a local minimum, that utilizes the sufficient condition for fast saddle escape





Anandkumar, Animashree and Rong Ge (2016). "Efficient approaches for escaping higher order saddle points in non-convex optimization". In: *Proc. Conf. Learning Theory*, pp. 81–102.

Daneshmand, Hadi et al. (2018). "Escaping saddles with stochastic gradients". In: *Proc. 35th International Conference on Machine Learning*, pp. 1155–1164.

Dixit, Rishabh and Waheed U Bajwa (2020). "Exit Time Analysis for Approximations of Gradient Descent Trajectories Around Saddle Points". In: *arXiv preprint arXiv:2006.01106*.

— (2021). "Boundary Conditions for Linear Exit Time Gradient Trajectories Around Saddle Points: Analysis and Algorithm". In: *arXiv preprint arXiv:2101.02625*.

Du, Simon S. et al. (2017). "Gradient descent can take exponential time to escape saddle points". In: *Proc. Advances in Neural Information Processing Systems*, pp. 1067–1077.

Erdogdu, Murat A., Lester Mackey, and Ohad Shamir (2018). "Global non-convex optimization with discretized diffusions". In: *Proc. Advances in Neural Information Processing Systems (NeurIPS'18)*, pp. 9671–9680.

Gelfand, Saul B. and Sanjoy K. Mitter (1991). "Recursive Stochastic Algorithms for Global Optimization in $\mathbb{R}^d$". In: *SIAM J. Control Optim.* 29.5, pp. 999–1018. DOI: 10.1137/0329055.

Jin, Chi, Rong Ge, et al. (2017). "How to escape saddle points efficiently". In: *Proc. 34th International Conference on Machine Learning*. JMLR. org, pp. 1724–1732.

Jin, Chi, Praneeth Netrapalli, and Michael I. Jordan (2018). "Accelerated gradient descent escapes saddle points faster than gradient descent". In: *Proc. 31st Conference on Learning Theory*, pp. 1042–1085.

Kifer, Yuri (1981). "The exit problem for small random perturbations of dynamical systems with a hyperbolic fixed point". In: *Israel Journal of Mathematics* 40.1, pp. 74–96.

Lee, Jason D. et al. (2017). "First-order methods almost always avoid saddle points". In: *arXiv preprint arXiv:1710.07406*.

- Li, Xingguo et al. (2019). "Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization". In: *IEEE Transactions on Information Theory* 65.6, pp. 3489–3514.
- Ma, Cong et al. (2020). "Implicit Regularization in Nonconvex Statistical Estimation: Gradient Descent Converges Linearly for Phase Retrieval, Matrix Completion, and Blind Deconvolution.". In: *Foundations of Computational Mathematics* 20.3, pp. 451–632.
- Mertikopoulos, Panayotis et al. (2020). "On the Almost Sure Convergence of Stochastic Gradient Descent in Non-Convex Problems". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 1117–1128.
- Mokhtari, Aryan, Asuman Ozdaglar, and Ali Jadbabaie (2018). "Escaping saddle points in constrained optimization". In: *Proc. Advances in Neural Information Processing Systems*, pp. 3629–3639.
- Murray, Ryan, Brian Swenson, and Soummya Kar (2019). "Revisiting normalized gradient descent: Fast evasion of saddle points". In: *IEEE Transactions on Automatic Control* 64.11, pp. 4818–4824.
- O'Neill, Michael and Stephen J. Wright (2019). "Behavior of accelerated gradient methods near critical points of nonconvex functions". In: *Mathematical Programming* 176.1-2, pp. 403–427.
- Paternain, Santiago, Aryan Mokhtari, and Alejandro Ribeiro (2019). "A Newton-based method for nonconvex optimization with fast evasion of saddle points". In: *SIAM Journal on Optimization* 29.1, pp. 343–368.
- Raginsky, Maxim, Alexander Rakhlin, and Matus Telgarsky (2017). "Non-convex learning via stochastic gradient Langevin dynamics: A nonasymptotic analysis". In: *Proc. Conf. Learning Theory (COLT'17)*. Amsterdam, Netherlands, pp. 1674–1703.
- Reddi, Sashank J. et al. (2018). "A generic approach for escaping saddle points". In: *Proc. 21st Intl. Conf. Artificial Intelligence and Statistics (AISTATS'18)*, pp. 1233–1242.
- Shi, Bin, Weijie J. Su, and Michael I. Jordan (2020). "On learning rates and Schrödinger operators". In: *arXiv preprint*. URL: https://arxiv.org/abs/2004.06977.
- Shub, Michael (2013). *Global stability of dynamical systems*. Springer Science & Business Media.

Bibliography III

Xu, Yi, Jing Rong, and Tianbao Yang (2018). "First-order stochastic algorithms for escaping from saddle points in almost linear time". In: *Proc. Advances in Neural Information Processing Systems*, pp. 5530–5540.

Yang, Jiaojiao, Wenqing Hu, and Chris Junchi Li (2021). "On the fast convergence of random perturbations of the gradient flow". In: *Asymptotic Analysis* 122.3-4, pp. 371–393.