Convex Optimization
Lecture on Non-convex Optimization

Tiebin Mi, **Zhenyu Liao**, Caiming Qiu

School of Electronic Information and Communications (EIC)
Huazhong University of Science and Technology (HUST)

May 18, 2023

Outline

Introduction and basic concepts

Some Non-convex Optimization Methods

Applications

Before we start: a brief self introduction

- **Zhenyu Liao**
  - 2010-2014: **B.Sc.** in Optical and Electronic Information, HUST
  - 2014-2016: **M.Sc.** in Signal and Image Processing, University of Paris-Saclay, France.
  - 2016-2019: **Ph.D.** in Statistics and Machine Learning, University of Paris-Saclay, France, under the supervision of Prof. Romain Couillet
  - 2020-2021: **Postdoctoral Scholar** at ICSI and Department of Statistics, University of California, Berkeley, hosted by Prof. Michael Mahoney.
  - 2021-now: **Research Associated Professor** at School of Electronic Information and Communications, HUST.
- **Homepage**: `https://zhenyu-liao.github.io/`
- **Research interest**: machine learning, signal processing, high-dimensional statistics

About today's lecture

- Introduction to non-convex optimization
- Basic concepts and mathematical tools
- Some non-convex optimization methods: non-convex projected GD, alternating minimization, stochastic optimization
- Some applications in signal processing and machine learning: sparse recovery, low-rank matrix recovery, and phase retrieval (MAY SKIP)
- **Reference**: Prateek Jain and Purushottam Kar. "Non-Convex Optimization for Machine Learning". In: *Foundations and Trends® in Machine Learning* 10.3-4 (Dec. 2017), 142–363. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000058

About today's lecture

- Introduction to non-convex optimization
- Basic concepts and mathematical tools
- Some non-convex optimization methods: non-convex projected GD, alternating minimization, stochastic optimization
- Some applications in signal processing and machine learning: sparse recovery, low-rank matrix recovery, and phase retrieval (MAY SKIP)
- **Reference**: Prateek Jain and Purushottam Kar. "Non-Convex Optimization for Machine Learning". In: *Foundations and Trends® in Machine Learning* 10.3-4 (Dec. 2017), 142–363. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000058

About today's lecture

- Introduction to non-convex optimization
- Basic concepts and mathematical tools
- Some non-convex optimization methods: non-convex projected GD, alternating minimization, stochastic optimization
- Some applications in signal processing and machine learning: sparse recovery, low-rank matrix recovery, and phase retrieval (MAY SKIP)
- **Reference**: Prateek Jain and Purushottam Kar. "Non-Convex Optimization for Machine Learning". In: *Foundations and Trends® in Machine Learning* 10.3-4 (Dec. 2017), 142–363. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000058

About today's lecture

- Introduction to non-convex optimization
- Basic concepts and mathematical tools
- Some non-convex optimization methods: non-convex projected GD, alternating minimization, stochastic optimization
- Some applications in signal processing and machine learning: sparse recovery, low-rank matrix recovery, and phase retrieval (MAY SKIP)
- **Reference**: Prateek Jain and Purushottam Kar. "Non-Convex Optimization for Machine Learning". In: *Foundations and Trends® in Machine Learning* 10.3-4 (Dec. 2017), 142–363. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000058

About today's lecture

- Introduction to non-convex optimization
- Basic concepts and mathematical tools
- Some non-convex optimization methods: non-convex projected GD, alternating minimization, stochastic optimization
- Some applications in signal processing and machine learning: sparse recovery, low-rank matrix recovery, and phase retrieval (MAY SKIP)
- **Reference**: Prateek Jain and Purushottam Kar. "Non-Convex Optimization for Machine Learning". In: *Foundations and Trends® in Machine Learning* 10.3-4 (Dec. 2017), 142–363. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000058

Outline

Introduction and basic concepts

Some Non-convex Optimization Methods

Applications

Non-convex optimization

» generic form of analytic optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{x}\in\mathcal{C},$$

with variable $\mathbf{x}\in\mathbb{R}^p$, objective function $f:\mathbb{R}^p\to\mathbb{R}$, and $\mathcal{C}\subset\mathbb{R}^p$ the constraint set.

» the problem is convex if **both** the objective f is a convex function and $\mathcal{C}$ is a convex set

» Examples of non-convex optimization problems:

- » **sparse regression**: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{y}-\mathbf{X}^\top\mathbf{w}\|^2, \quad \text{s.t. } \|\mathbf{w}\|_0 \le s \ll p$
- » **recommendation system**: (low rank) matrix completion problem as $\hat{\mathbf{A}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \sum_{(i,j)\in\Omega}(X_{ij}-A_{ij})^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{X}) \le r$

» life is hard and math is difficult, we resort to convex relaxation, and hope the gap is small

Non-convex optimization

- generic form of analytic optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x}) \qquad \text{s.t.} \quad \mathbf{x} \in \mathcal{C},$$

  with variable $\mathbf{x} \in \mathbb{R}^p$, objective function $f : \mathbb{R}^p \to \mathbb{R}$, and $\mathcal{C} \subset \mathbb{R}^p$ the constraint set.
- the problem is convex if **both** the objective f is a convex function and $\mathcal{C}$ is a convex set
- Examples of non-convex optimization problems:
  - **sparse regression**: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2, \quad \text{s.t. } \|\mathbf{w}\|_0 \le s \ll p$
  - **recommendation system**: (low rank) matrix completion problem as $\hat{\mathbf{A}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \sum_{(i,j)\in\Omega} (X_{ij} - A_{ij})^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{X}) \le r$
- life is hard and math is difficult, we resort to convex relaxation, and hope the gap is small

Non-convex optimization

- generic form of analytic optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{x}\in\mathcal{C},$$

  with variable $\mathbf{x}\in\mathbb{R}^p$, objective function $f:\mathbb{R}^p\to\mathbb{R}$, and $\mathcal{C}\subset\mathbb{R}^p$ the constraint set.
- the problem is convex if **both** the objective f is a convex function and $\mathcal{C}$ is a convex set
- Examples of non-convex optimization problems:
  - **sparse regression**: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{y}-\mathbf{X}^\mathsf{T}\mathbf{w}\|^2, \quad \text{s.t. } \|\mathbf{w}\|_0 \le s \ll p$
  - **recommendation system**: (low rank) matrix completion problem as $\hat{\mathbf{A}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \sum_{(i,j)\in\Omega}(X_{ij}-A_{ij})^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{X})\le r$
- life is hard and math is difficult, we resort to convex relaxation, and hope the gap is small

Non-convex optimization

- generic form of analytic optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{x} \in \mathcal{C},$$

  with variable $\mathbf{x} \in \mathbb{R}^p$, objective function $f : \mathbb{R}^p \rightarrow \mathbb{R}$, and $\mathcal{C} \subset \mathbb{R}^p$ the constraint set.
- the problem is convex if **both** the objective f is a convex function and $\mathcal{C}$ is a convex set
- Examples of non-convex optimization problems:
  - **sparse regression**: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|^2, \quad \text{s.t. } \|\mathbf{w}\|_0 \leq s \ll p$
  - **recommendation system**: (low rank) matrix completion problem as $\hat{\mathbf{A}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \sum_{(i,j)\in\Omega}(X_{ij} - A_{ij})^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{X}) \leq r$
- life is hard and math is difficult, we resort to convex relaxation, and hope the gap is small

Non-convex optimization

- generic form of analytic optimization problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^p} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{C}, \end{aligned}$$

  with variable $\mathbf{x} \in \mathbb{R}^p$, objective function $f : \mathbb{R}^p \to \mathbb{R}$, and $\mathcal{C} \subset \mathbb{R}^p$ the constraint set.
- the problem is convex if **both** the objective f is a convex function and $\mathcal{C}$ is a convex set
- Examples of non-convex optimization problems:
  - **sparse regression**: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|^2, \quad \text{s.t. } \|\mathbf{w}\|_0 \leq s \ll p$
  - **recommendation system**: (low rank) matrix completion problem as $\hat{\mathbf{A}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \sum_{(i,j)\in\Omega}(X_{ij} - A_{ij})^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{X}) \leq r$
- life is hard and math is difficult, we resort to convex relaxation, and hope the gap is small

Non-convex optimization

- generic form of analytic optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{x} \in \mathcal{C},$$

  with variable $\mathbf{x} \in \mathbb{R}^p$, objective function $f : \mathbb{R}^p \to \mathbb{R}$, and $\mathcal{C} \subset \mathbb{R}^p$ the constraint set.
- the problem is convex if **both** the objective f is a convex function and $\mathcal{C}$ is a convex set
- Examples of non-convex optimization problems:
  - **sparse regression**: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|^2, \quad \text{s.t. } \|\mathbf{w}\|_0 \leq s \ll p$
  - **recommendation system**: (low rank) matrix completion problem as $\hat{\mathbf{A}} = \arg\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \sum_{(i,j)\in\Omega}(X_{ij} - A_{ij})^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{X}) \leq r$
- life is hard and math is difficult, we resort to convex relaxation, and hope the gap is small

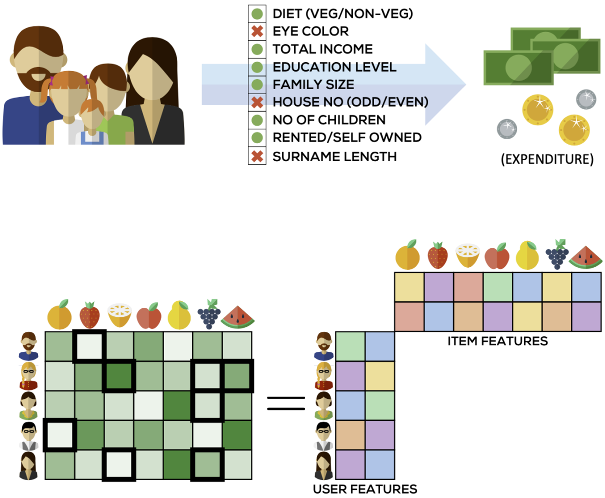


Figure: Examples of applications of non-convex optimization

Convex versus non-convex optimization

» facing a non-convex optimization problem, we may either

(i) resort to **convex relation** of the problem, and hope that the problem is nice enough for the gap to be small; or

(ii) (somewhat naively) solve it using **non-convex** optimization approaches (such as gradient descent, alternating minimization, and the expectation-maximization algorithm, etc.) and ?

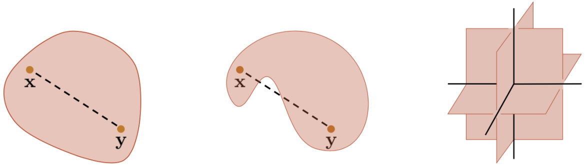
» in fact it turns out that if the problems possess nice structure, both approaches work, and non-convex techniques may even be more **efficient** (in term of complexity)!

Convex versus non-convex optimization

- facing a non-convex optimization problem, we may either
  - (i) resort to **convex relation** of the problem, and hope that the problem is nice enough for the gap to be small; or
  - (ii) (somewhat naively) solve it using **non-convex** optimization approaches (such as gradient descent, alternating minimization, and the expectation-maximization algorithm, etc.) and ?
- in fact it turns out that if the problems possess nice structure, both approaches work, and non-convex techniques may even be more **efficient** (in term of complexity)!

Recap on convex analysis

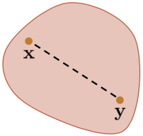
- Convex combination: for $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, $\mathbf{x}_{\boldsymbol{\theta}} \equiv \sum_{i=1}^{n} \theta_i \mathbf{x}_i$ with $\theta_0 \geq 0$ and $\sum_{i=1}^{n} \theta_i = 1$.
- Convex set: $\mathcal{C}$ such that if $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ then for any $\lambda \in [0, 1]$, $(1 - \lambda)\mathbf{x} + \lambda \mathbf{y} \in \mathcal{C}$
- Convex function: (if continuously differentiable) $f : \mathbb{R}^p \to \mathbb{R}$ if $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ then $f(\mathbf{y}) \geq f(\mathbf{x}) - \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle$, with $\nabla f(\mathbf{x})$ the gradient of f at $\mathbf{x}$



$\mathcal{C} \subseteq \mathbb{R}^d$ CONVEX SET — $\mathcal{C} \subseteq \mathbb{R}^d$ NON-CONVEX SET — $\mathcal{B}_0(2) \subseteq \mathbb{R}^3$ NON-CONVEX SET

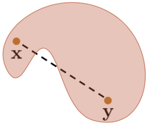
Recap on convex analysis

- Convex combination: for $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, $\mathbf{x}_{\boldsymbol{\theta}} \equiv \sum_{i=1}^n \theta_i \mathbf{x}_i$ with $\theta_0 \geq 0$ and $\sum_{i=1}^n \theta_i = 1$.
- Convex set: $\mathcal{C}$ such that if $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ then for any $\lambda \in [0,1]$, $(1-\lambda)\mathbf{x} + \lambda \mathbf{y} \in \mathcal{C}$
- Convex function: (if continuously differentiable) $f : \mathbb{R}^p \to \mathbb{R}$ if $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ then $f(\mathbf{y}) \geq f(\mathbf{x}) - \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle$, with $\nabla f(\mathbf{x})$ the gradient of f at $\mathbf{x}$



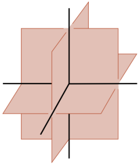
$\mathcal{C} \subseteq \mathbb{R}^d$

CONVEX SET



$\mathcal{C} \subseteq \mathbb{R}^d$

NON-CONVEX SET

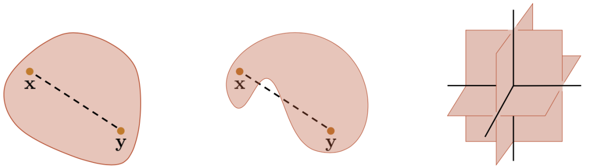


$\mathcal{B}_0(2) \subseteq \mathbb{R}^3$

NON-CONVEX SET

Recap on convex analysis

- Convex combination: for $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, $\mathbf{x}_{\boldsymbol{\theta}} \equiv \sum_{i=1}^n \theta_i \mathbf{x}_i$ with $\theta_0 \geq 0$ and $\sum_{i=1}^n \theta_i = 1$.
- Convex set: $\mathcal{C}$ such that if $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ then for any $\lambda \in [0,1]$, $(1-\lambda)\mathbf{x} + \lambda\mathbf{y} \in \mathcal{C}$
- Convex function: (if continuously differentiable) $f : \mathbb{R}^p \to \mathbb{R}$ if $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ then $f(\mathbf{y}) \geq f(\mathbf{x}) - \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle$, with $\nabla f(\mathbf{x})$ the gradient of f at $\mathbf{x}$



$\mathcal{C} \subseteq \mathbb{R}^d$ CONVEX SET — $\mathcal{C} \subseteq \mathbb{R}^d$ NON-CONVEX SET — $\mathcal{B}_0(2) \subseteq \mathbb{R}^3$ NON-CONVEX SET

Convex projection

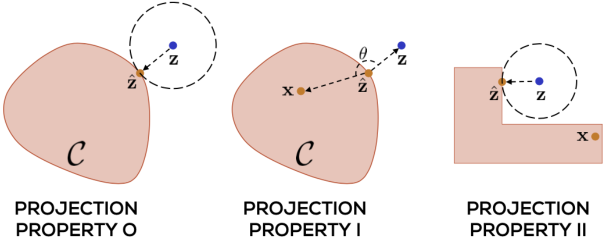
» For any closed set (convex or not) $\mathcal{C} \subset \mathbb{R}^p$ and $\mathbf{z} \in \mathbb{R}^p$, **projection** onto $\mathcal{C}$ as $\Pi_{\mathcal{C}}(\mathbf{z}) \equiv \arg\min_{\mathbf{x}\in\mathcal{C}} \|\mathbf{x} - \mathbf{z}\|$

» properties of $\Pi_{\mathcal{C}}(\cdot)$:

(i) any closed set $\mathcal{C}$, then for all $\mathbf{x} \in \mathcal{C}$, $\|\Pi_{\mathcal{C}}(\mathbf{z}) - \mathbf{z}\| \leq \|\mathbf{x} - \mathbf{z}\|$

(ii) convex set $\mathcal{C}$, then for all $\mathbf{x} \in \mathcal{C}$, $\langle \mathbf{x} - \Pi_{\mathcal{C}}(\mathbf{z}), \mathbf{z} - \Pi_{\mathcal{C}}(\mathbf{z}) \rangle \leq 0$

(iii) **contraction property**: convex $\mathcal{C}$, then for all $\mathbf{x} \in \mathcal{C}$, $\|\Pi_{\mathcal{C}}(\mathbf{z}) - \mathbf{x}\| \leq \|\mathbf{z} - \mathbf{x}\|$



Convex projection

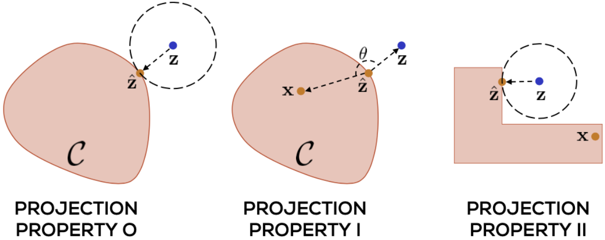
» For any closed set (convex or not) $\mathcal{C} \subset \mathbb{R}^p$ and $\mathbf{z} \in \mathbb{R}^p$, **projection** onto $\mathcal{C}$ as $\Pi_{\mathcal{C}}(\mathbf{z}) \equiv \arg\min_{\mathbf{x}\in\mathcal{C}} \|\mathbf{x} - \mathbf{z}\|$

» properties of $\Pi_{\mathcal{C}}(\cdot)$:

(i) any closed set $\mathcal{C}$, then for all $\mathbf{x} \in \mathcal{C}$, $\|\Pi_{\mathcal{C}}(\mathbf{z}) - \mathbf{z}\| \leq \|\mathbf{x} - \mathbf{z}\|$

(ii) convex set $\mathcal{C}$, then for all $\mathbf{x} \in \mathcal{C}$, $\langle \mathbf{x} - \Pi_{\mathcal{C}}(\mathbf{z}), \mathbf{z} - \Pi_{\mathcal{C}}(\mathbf{z}) \rangle \leq 0$

(iii) **contraction property**: convex $\mathcal{C}$, then for all $\mathbf{x} \in \mathcal{C}$, $\|\Pi_{\mathcal{C}}(\mathbf{z}) - \mathbf{x}\| \leq \|\mathbf{z} - \mathbf{x}\|$



Convex projection: a few (practical) examples

» for $\mathcal{C} = \mathcal{B}_2(1)$, that is, the unit L_2 ball, the projection is equivalent to normalization

$$\Pi_{\mathcal{B}_2(1)}(\mathbf{z}) = \begin{cases} \mathbf{z}/\|\mathbf{z}\|, & \text{if } \|\mathbf{z}\| \geq 1 \\ \mathbf{z}, & \text{otherwise} \end{cases} \tag{1}$$

» for $\mathcal{C} = \mathcal{B}_1(1)$, the unit L_1 ball, the projection is equivalent to soft-thresholding: $\hat{\mathbf{z}} = \Pi_{\mathcal{B}_1(1)}(\mathbf{z})$, then $\hat{z}_i = \max(z_i - \theta, 0)$ for a threshold θ determined by a sorting on $\mathbf{z}$

» for $\mathcal{C} = \mathcal{B}_0(1)$, non-convex set! but hard-thresholding, see later

Convex projection: a few (practical) examples

- for $\mathcal{C} = \mathcal{B}_2(1)$, that is, the unit L_2 ball, the projection is equivalent to normalization

$$\Pi_{\mathcal{B}_2(1)}(\mathbf{z}) = \begin{cases} \mathbf{z}/\|\mathbf{z}\|, & \text{if } \|\mathbf{z}\| \geq 1 \\ \mathbf{z}, & \text{otherwise} \end{cases} \tag{1}$$

- for $\mathcal{C} = \mathcal{B}_1(1)$, the unit L_1 ball, the projection is equivalent to soft-thresholding: $\hat{\mathbf{z}} = \Pi_{\mathcal{B}_1(1)}(\mathbf{z})$, then $\hat{z}_i = \max(z_i - \theta, 0)$ for a threshold θ determined by a sorting on $\mathbf{z}$
- for $\mathcal{C} = \mathcal{B}_0(1)$, non-convex set! but hard-thresholding, see later

Convex projection: a few (practical) examples

» for $\mathcal{C} = \mathcal{B}_2(1)$, that is, the unit L_2 ball, the projection is equivalent to normalization

$$\Pi_{\mathcal{B}_2(1)}(\mathbf{z}) = \begin{cases} \mathbf{z}/\|\mathbf{z}\|, & \text{if } \|\mathbf{z}\| \geq 1 \\ \mathbf{z}, & \text{otherwise} \end{cases} \tag{1}$$

» for $\mathcal{C} = \mathcal{B}_1(1)$, the unit L_1 ball, the projection is equivalent to soft-thresholding: $\hat{\mathbf{z}} = \Pi_{\mathcal{B}_1(1)}(\mathbf{z})$, then $\hat{z}_i = \max(z_i - \theta, 0)$ for a threshold θ determined by a sorting on $\mathbf{z}$

» for $\mathcal{C} = \mathcal{B}_0(1)$, non-convex set! but hard-thresholding, see later

$$\min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x}), \quad \text{s.t. } \mathbf{x} \in \mathcal{C}. \tag{2}$$

Algorithm Projected Gradient Descent (PGD)

Input: Convex objective f, convex constraint set $\mathcal{C}$, step lengths η_t

Output: A point $\hat{\mathbf{x}} \in \mathcal{C}$ with near-optimal objective value

1: $\mathbf{x}(0) = \mathbf{0}$

2: **for** $t = 1, 2, \ldots, T$ **do**

3: $\quad \mathbf{z}(t+1) \leftarrow \mathbf{x}(t) - \eta_t \cdot \nabla f(\mathbf{x}(t))$

4: $\quad \mathbf{x}(t+1) \leftarrow \Pi_{\mathcal{C}}(\mathbf{z}(t+1))$

5: **end for**

6: (OPTION 1) **return** $\hat{\mathbf{x}}_{\text{final}} = \mathbf{x}(T)$

7: (OPTION 2) **return** $\hat{\mathbf{x}}_{\text{avg}} = (\sum_{t=1}^{T} \mathbf{x}(t))/T$

8: (OPTION 3) **return** $\hat{\mathbf{x}}_{\text{best}} = \arg\min_{t\in[T]} f(\mathbf{x}(t))$

A few comments on PGD

- in the proof of the convergence of PGD, we generally get step length $\eta_t = 1/\sqrt{T}$, with T the **total** number of iterations: **horizon-aware**
- **horizon-oblivious**: take $\eta_t = 1/\sqrt{t}$ also works, in theory
- in practice: the step length η_t is tuned globally by doing a **grid search** over several possible values (akin to the horizon-aware setting), or per-iteration using **line search** mechanisms (akin to the horizon-oblivious setting), to obtain a step length value that assures good convergence rates
  - **line search**: for a given direction $\mathbf{g}(\mathbf{x}(t))$, choose $\eta_t \geq 0$ that (exactly or "loosely") minimize $h(\eta_t) = f(\mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t)))$, and update as $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t))$

A few comments on PGD

- in the proof of the convergence of PGD, we generally get step length $\eta_t = 1/\sqrt{T}$, with T the **total** number of iterations: **horizon-aware**
- **horizon-oblivious**: take $\eta_t = 1/\sqrt{t}$ also works, in theory
- in practice: the step length η_t is tuned globally by doing a **grid search** over several possible values (akin to the horizon-aware setting), or per-iteration using **line search** mechanisms (akin to the horizon-oblivious setting), to obtain a step length value that assures good convergence rates
  - **line search**: for a given direction $\mathbf{g}(\mathbf{x}(t))$, choose $\eta_t \geq 0$ that (exactly or "loosely") minimize $h(\eta_t) = f(\mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t)))$, and update as $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t))$

A few comments on PGD

- in the proof of the convergence of PGD, we generally get step length $\eta_t = 1/\sqrt{T}$, with T the **total** number of iterations: **horizon-aware**
- **horizon-oblivious**: take $\eta_t = 1/\sqrt{t}$ also works, in theory
- in practice: the step length η_t is tuned globally by doing a **grid search** over several possible values (akin to the horizon-aware setting), or per-iteration using **line search** mechanisms (akin to the horizon-oblivious setting), to obtain a step length value that assures good convergence rates
    - **line search**: for a given direction $\mathbf{g}(\mathbf{x}(t))$, choose $\eta_t \geq 0$ that (exactly or "loosely") minimize $h(\eta_t) = f(\mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t)))$, and update as $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t))$

A few comments on PGD

- in the proof of the convergence of PGD, we generally get step length $\eta_t = 1/\sqrt{T}$, with T the **total** number of iterations: **horizon-aware**
- **horizon-oblivious**: take $\eta_t = 1/\sqrt{t}$ also works, in theory
- in practice: the step length η_t is tuned globally by doing a **grid search** over several possible values (akin to the horizon-aware setting), or per-iteration using **line search** mechanisms (akin to the horizon-oblivious setting), to obtain a step length value that assures good convergence rates
  - **line search**: for a given direction $\mathbf{g}(\mathbf{x}(t))$, choose $\eta_t \geq 0$ that (exactly or "loosely") minimize $h(\eta_t) = f(\mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t)))$, and update as $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \cdot \mathbf{g}(\mathbf{x}(t))$

Outline

Introduction and basic concepts

Some Non-convex Optimization Methods

Applications

Projected gradient descent (again), but non-convex

- PGD practically applies to convex and non-convex problems (we will see why)
- however, the projection onto a non-convex $\mathcal{C}$ can already be **NP-hard**

For $\mathbf{z} \in \mathbb{R}^p$, let σ be the permutation that sorts the entries of $\mathbf{z}$ in decreasing order, $|z_{\sigma(1)}| \geq \ldots \geq |z_{\sigma(p)}|$, then $\Pi_{\mathcal{B}_0(s)}(\mathbf{z}) = [z_i \cdot \mathbf{1}_{\sigma(i) \leq s}]$, with $\mathcal{B}_0(s) \equiv \{\mathbf{x} \in \mathbb{R}^p, \|\mathbf{x}\|_0 \leq s\}$.

Projection into sparse vectors

- also known as the hard-thresholding

For $\mathbf{A} \in \mathbb{R}^{m \times n}$ with singular value decomposition $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T}$, then $\Pi_{\mathcal{B}_{\text{rank}}(r)}(\mathbf{A}) = \mathbf{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\mathbf{V}_{(r)}^\mathsf{T}$ for any $r \leq \min(m, n)$, with $\mathcal{B}_{\text{rank}}(r) \equiv \{\mathbf{A} \in \mathbb{R}^{m \times n}, \text{rank}(\mathbf{A}) \leq r\}$.

Projection into low-rank matrices (Eckart-Young-Mirsky theorem)

Projected gradient descent (again), but non-convex

» PGD practically applies to convex and non-convex problems (we will see why)

» however, the projection onto a non-convex $\mathcal{C}$ can already be **NP-hard**

For $\mathbf{z} \in \mathbb{R}^p$, let σ be the permutation that sorts the entries of $\mathbf{z}$ in decreasing order, $|z_{\sigma(1)}| \geq \ldots \geq |z_{\sigma(p)}|$, then $\Pi_{\mathcal{B}_0(s)}(\mathbf{z}) = [z_i \cdot \mathbf{1}_{\sigma(i) \leq s}]$, with $\mathcal{B}_0(s) \equiv \{\mathbf{x} \in \mathbb{R}^p, \|\mathbf{x}\|_0 \leq s\}$.

Projection into sparse vectors

» also known as the hard-thresholding

For $\mathbf{A} \in \mathbb{R}^{m \times n}$ with singular value decomposition $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, then $\Pi_{\mathcal{B}_{\text{rank}}(r)}(\mathbf{A}) = \mathbf{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\mathbf{V}_{(r)}^\top$ for any $r \leq \min(m, n)$, with $\mathcal{B}_{\text{rank}}(r) \equiv \{\mathbf{A} \in \mathbb{R}^{m \times n}, \text{rank}(\mathbf{A}) \leq r\}$.

Projection into low-rank matrices (Eckart-Young-Mirsky theorem)

Projected gradient descent (again), but non-convex

- PGD practically applies to convex and non-convex problems (we will see why)
- however, the projection onto a non-convex $\mathcal{C}$ can already be **NP-hard**

For $\mathbf{z} \in \mathbb{R}^p$, let σ be the permutation that sorts the entries of $\mathbf{z}$ in decreasing order, $|z_{\sigma(1)}| \geq \ldots \geq |z_{\sigma(p)}|$, then $\Pi_{\mathcal{B}_0(s)}(\mathbf{z}) = [z_i \cdot \mathbf{1}_{\sigma(i) \leq s}]$, with $\mathcal{B}_0(s) \equiv \{\mathbf{x} \in \mathbb{R}^p, \|\mathbf{x}\|_0 \leq s\}$.

Projection into sparse vectors

- also known as the hard-thresholding

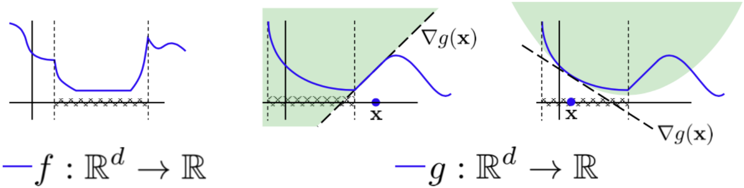
For $\mathbf{A} \in \mathbb{R}^{m \times n}$ with singular value decomposition $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, then $\Pi_{\mathcal{B}_{\text{rank}}(r)}(\mathbf{A}) = \mathbf{U}_{(r)}\boldsymbol{\Sigma}_{(r)}\mathbf{V}_{(r)}^\top$ for any $r \leq \min(m, n)$, with $\mathcal{B}_{\text{rank}}(r) \equiv \{\mathbf{A} \in \mathbb{R}^{m \times n}, \text{rank}(\mathbf{A}) \leq r\}$.

Projection into low-rank matrices (Eckart-Young-Mirsky theorem)

Intuition on how this might work for non-convex problems

» (generally) non-convex can be **restricted convex** if $f : \mathbb{R}^p \to \mathbb{R}$ over a (possibly non-smooth) region $\mathcal{C} \subset \mathbb{R}^p$ satisfies $\langle \mathbf{x} - \Pi_{\mathcal{C}}(\mathbf{z}), \mathbf{z} - \Pi_{\mathcal{C}}(\mathbf{z}) \rangle \leq 0$

» so everything should work as in the convex case with (almost) the same PGD approach, and this is indeed the case



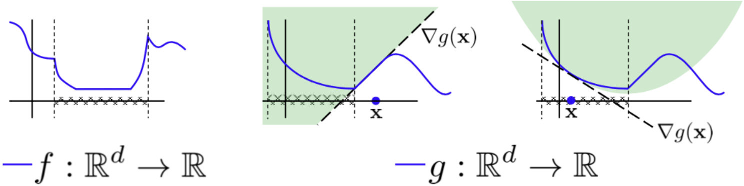
RESTRICTED CONVEXITY

A NON-CONVEX FUNCTION THAT SATISFIES RESTRICTED STRONG CONVEXITY

Intuition on how this might work for non-convex problems

» (generally) non-convex can be **restricted convex** if $f : \mathbb{R}^p \to \mathbb{R}$ over a (possibly non-smooth) region $\mathcal{C} \subset \mathbb{R}^p$ satisfies $\langle \mathbf{x} - \Pi_{\mathcal{C}}(\mathbf{z}), \mathbf{z} - \Pi_{\mathcal{C}}(\mathbf{z}) \rangle \leq 0$

» so everything should work as in the convex case with (almost) the same PGD approach, and this is indeed the case



RESTRICTED CONVEXITY

A NON-CONVEX FUNCTION THAT SATISFIES RESTRICTED STRONG CONVEXITY

Alternative Minimization

- useful when the optimization concerns with **two or more** groups of variables, e.g., in low-rank matrix completion, find $\mathbf{X} \in \mathbb{R}^{m\times n}$ such that $\operatorname{rank}(\mathbf{X}) = r \Leftrightarrow \mathbf{X} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U} \in \mathbb{R}^{m\times r}, \mathbf{V} \in \mathbb{R}^{n\times r}$
- in these case, the problem may not be **jointly convex** in all the variables
- **Joint convexity**: for $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continuously differentiable in two variables, if for every $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_1, \mathbf{y}_1) \in \mathbb{R}^{p\times q}$ one has $f(\mathbf{x}_2, \mathbf{y}_2) \geq f(\mathbf{x}_1, \mathbf{y}_1) + \langle \nabla f(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) - (\mathbf{x}_1, \mathbf{y}_1)\rangle$, same as convexity in $\mathbf{z} = [\mathbf{x},\ \mathbf{y}]^\mathsf{T}$
- f is **marginally convex** in its first variable if for every given $\mathbf{y} \in \mathbb{R}^q$, the function $(\cdot, \mathbf{y}) : \mathbb{R}^p \to \mathbb{R}$ is convex, that is $f(\mathbf{x}_2, \mathbf{y}) \geq f(\mathbf{x}_1, \mathbf{y}) + \langle \nabla_x f(\mathbf{x}_1, \mathbf{y}), \mathbf{x}_2 - \mathbf{x}_1\rangle$
- the idea is simple: solve for one variable (which is convex), with other variables **fixed**

Alternative Minimization

- » useful when the optimization concerns with **two or more** groups of variables, e.g., in low-rank matrix completion, find $\mathbf{X} \in \mathbb{R}^{m \times n}$ such that $\operatorname{rank}(\mathbf{X}) = r \Leftrightarrow \mathbf{X} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U} \in \mathbb{R}^{m \times r}, \mathbf{V} \in \mathbb{R}^{n \times r}$
- » in these case, the problem may not be **jointly convex** in all the variables
- » **Joint convexity**: for $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continuously differentiable in two variables, if for every $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_1, \mathbf{y}_1) \in \mathbb{R}^{p \times q}$ one has $f(\mathbf{x}_2, \mathbf{y}_2) \geq f(\mathbf{x}_1, \mathbf{y}_1) + \langle \nabla f(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) - (\mathbf{x}_1, \mathbf{y}_1) \rangle$, same as convexity in $\mathbf{z} = [\mathbf{x},\ \mathbf{y}]^T$
- » f is **marginally convex** in its first variable if for every given $\mathbf{y} \in \mathbb{R}^q$, the function $(\cdot, \mathbf{y}) : \mathbb{R}^p \to \mathbb{R}$ is convex, that is $f(\mathbf{x}_2, \mathbf{y}) \geq f(\mathbf{x}_1, \mathbf{y}) + \langle \nabla_x f(\mathbf{x}_1, \mathbf{y}), \mathbf{x}_2 - \mathbf{x}_1 \rangle$
- » the idea is simple: solve for one variable (which is convex), with other variables **fixed**

Alternative Minimization

- useful when the optimization concerns with **two or more** groups of variables, e.g., in low-rank matrix completion, find $\mathbf{X} \in \mathbb{R}^{m\times n}$ such that $\mathrm{rank}(\mathbf{X}) = r \Leftrightarrow \mathbf{X} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U} \in \mathbb{R}^{m\times r}, \mathbf{V} \in \mathbb{R}^{n\times r}$
- in these case, the problem may not be **jointly convex** in all the variables
- **Joint convexity**: for $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continuously differentiable in two variables, if for every $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_1, \mathbf{y}_1) \in \mathbb{R}^{p\times q}$ one has $f(\mathbf{x}_2, \mathbf{y}_2) \geq f(\mathbf{x}_1, \mathbf{y}_1) + \langle \nabla f(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) - (\mathbf{x}_1, \mathbf{y}_1)\rangle$, same as convexity in $\mathbf{z} = [\mathbf{x},\ \mathbf{y}]^\mathsf{T}$
- f is **marginally convex** in its first variable if for every given $\mathbf{y} \in \mathbb{R}^q$, the function $(\cdot, \mathbf{y}) : \mathbb{R}^p \to \mathbb{R}$ is convex, that is $f(\mathbf{x}_2, \mathbf{y}) \geq f(\mathbf{x}_1, \mathbf{y}) + \langle \nabla_x f(\mathbf{x}_1, \mathbf{y}), \mathbf{x}_2 - \mathbf{x}_1\rangle$
- the idea is simple: solve for one variable (which is convex), with other variables **fixed**

Alternative Minimization

- useful when the optimization concerns with **two or more** groups of variables, e.g., in low-rank matrix completion, find $\mathbf{X} \in \mathbb{R}^{m \times n}$ such that $\mathrm{rank}(\mathbf{X}) = r \Leftrightarrow \mathbf{X} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U} \in \mathbb{R}^{m \times r}, \mathbf{V} \in \mathbb{R}^{n \times r}$
- in these case, the problem may not be **jointly convex** in all the variables
- **Joint convexity**: for $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continuously differentiable in two variables, if for every $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_1, \mathbf{y}_1) \in \mathbb{R}^{p \times q}$ one has $f(\mathbf{x}_2, \mathbf{y}_2) \geq f(\mathbf{x}_1, \mathbf{y}_1) + \langle \nabla f(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) - (\mathbf{x}_1, \mathbf{y}_1) \rangle$, same as convexity in $\mathbf{z} = [\mathbf{x},\ \mathbf{y}]^\mathsf{T}$
- f is **marginally convex** in its first variable if for every given $\mathbf{y} \in \mathbb{R}^q$, the function $(\cdot, \mathbf{y}) : \mathbb{R}^p \to \mathbb{R}$ is convex, that is $f(\mathbf{x}_2, \mathbf{y}) \geq f(\mathbf{x}_1, \mathbf{y}) + \langle \nabla_\mathbf{x} f(\mathbf{x}_1, \mathbf{y}), \mathbf{x}_2 - \mathbf{x}_1 \rangle$
- the idea is simple: solve for one variable (which is convex), with other variables **fixed**

Alternative Minimization

- useful when the optimization concerns with **two or more** groups of variables, e.g., in low-rank matrix completion, find $\mathbf{X} \in \mathbb{R}^{m\times n}$ such that $\mathrm{rank}(\mathbf{X}) = r \Leftrightarrow \mathbf{X} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U} \in \mathbb{R}^{m\times r}, \mathbf{V} \in \mathbb{R}^{n\times r}$
- in these case, the problem may not be **jointly convex** in all the variables
- **Joint convexity**: for $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continuously differentiable in two variables, if for every $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_1, \mathbf{y}_1) \in \mathbb{R}^{p\times q}$ one has $f(\mathbf{x}_2, \mathbf{y}_2) \geq f(\mathbf{x}_1, \mathbf{y}_1) + \langle \nabla f(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) - (\mathbf{x}_1, \mathbf{y}_1)\rangle$, same as convexity in $\mathbf{z} = [\mathbf{x},\ \mathbf{y}]^\mathsf{T}$
- f is **marginally convex** in its first variable if for every given $\mathbf{y} \in \mathbb{R}^q$, the function $(\cdot, \mathbf{y}) : \mathbb{R}^p \to \mathbb{R}$ is convex, that is $f(\mathbf{x}_2, \mathbf{y}) \geq f(\mathbf{x}_1, \mathbf{y}) + \langle \nabla_{\mathbf{x}} f(\mathbf{x}_1, \mathbf{y}), \mathbf{x}_2 - \mathbf{x}_1\rangle$
- the idea is simple: solve for one variable (which is convex), with other variables **fixed**

Generalized Alternating Minimization (gAM)

Algorithm Generalized Alternating Minimization (gAM)

Input: Objective function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$
Output: A point $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \in \mathcal{X} \times \mathcal{Y}$ with near-optimal objective value

1: $(\mathbf{x}(0), \mathbf{y}(0)) \leftarrow \mathsf{INIT}()$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $\quad \mathbf{x}(t+1) \leftarrow \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \mathbf{y}(t))$
4: $\quad \mathbf{y}(t+1) \leftarrow \arg\min_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}(t+1), \mathbf{y})$
5: **end for**
6: **return** $(\mathbf{x}(T), \mathbf{y}(T))$

» we can of course use gradient descent to solve the marginal optimization problem

gAM always work well? No!

For any given $\mathbf{y} \in \mathcal{Y}$, we say $\tilde{\mathbf{x}}$ is a marginally optimal coordinate with respect to $\mathbf{y}$, and denote $\tilde{\mathbf{x}} \in \mathsf{mOPT}_f(\mathbf{y})$ if $f(\tilde{\mathbf{x}}, \mathbf{y}) \leq f(\mathbf{x}, \mathbf{y})$ for all $\mathbf{x} \in \mathcal{X}$, and similarly for $\tilde{\mathbf{y}} \in \mathsf{mOPT}_f(\mathbf{x})$.

Marginally Optimum Coordinate

A point $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is a **bistable point** if $\tilde{\mathbf{x}} \in \mathsf{mOPT}_f(\mathbf{y})$ and $\tilde{\mathbf{y}} \in \mathsf{mOPT}_f(\mathbf{x})$.

Bistable Point

» the optimum of the optimization problem must be a bistable point

» but gAM must stop at a bistable point

gAM always work well? No!

For any given $\mathbf{y} \in \mathcal{Y}$, we say $\tilde{\mathbf{x}}$ is a marginally optimal coordinate with respect to $\mathbf{y}$, and denote $\tilde{\mathbf{x}} \in \mathsf{mOPT}_f(\mathbf{y})$ if $f(\tilde{\mathbf{x}}, \mathbf{y}) \leq f(\mathbf{x}, \mathbf{y})$ for all $\mathbf{x} \in \mathcal{X}$, and similarly for $\tilde{\mathbf{y}} \in \mathsf{mOPT}_f(\mathbf{x})$.

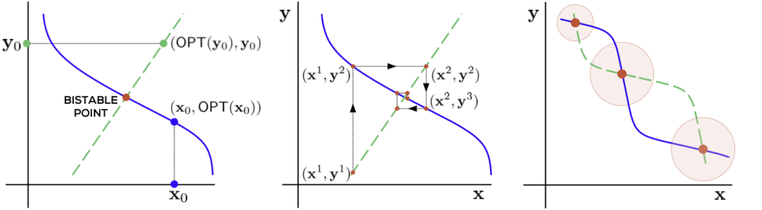
Marginally Optimum Coordinate

A point $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is a **bistable point** if $\tilde{\mathbf{x}} \in \mathsf{mOPT}_f(\mathbf{y})$ and $\tilde{\mathbf{y}} \in \mathsf{mOPT}_f(\mathbf{x})$.

Bistable Point

- the optimum of the optimization problem must be a bistable point
- but gAM must stop at a bistable point

gAM and its convergence (or not) in non-convex problems



MARGINAL OPTIMALITY PLOT

gAM ITERATES CONVERGE TO THE BISTABLE POINT

MULTIPLE BISTABLE POINTS WITH RESPECTIVE REGIONS OF ATTRACTION

» when having multiple bistable points, convergence depends on initialization (so in fact the problem **structure**), with detailed analysis on the "region of attraction" of different bistable points

Convergence of gAM for convex problems

Things are (again) nice for convex problems

- for differentiable (jointly) convex functions, all bistable points are global minima, so any one is good enough
- **(Block) Coordinate Minimization** approach: solve a single p-dimensional variable $x \in \mathbb{R}^p$ as p one-dimensional variables $\{x_1, \ldots, x_p\}$, useful in large-scale convex optimization
- may not work well for **non-differentiable** optimization problems

For **non-convex** problems:

- we can only converge to bistable points, and hope they are (or at least close, in some sense, to) global minima

Convergence of gAM for convex problems

Things are (again) nice for convex problems

- for differentiable (jointly) convex functions, all bistable points are global minima, so any one is good enough
- **(Block) Coordinate Minimization** approach: solve a single p-dimensional variable $x \in \mathbb{R}^p$ as p one-dimensional variables $\{x_1, \ldots, x_p\}$, useful in large-scale convex optimization
- may not work well for **non-differentiable** optimization problems

For **non-convex** problems:

- we can only converge to bistable points, and hope they are (or at least close, in some sense, to) global minima

Convergence of gAM for convex problems

Things are (again) nice for convex problems

- for differentiable (jointly) convex functions, all bistable points are global minima, so any one is good enough
- **(Block) Coordinate Minimization** approach: solve a single p-dimensional variable $x \in \mathbb{R}^p$as p one-dimensional variables $\{x_1, \ldots, x_p\}$, useful in large-scale convex optimization
- may not work well for **non-differentiable** optimization problems

For **non-convex** problems:

- we can only converge to bistable points, and hope they are (or at least close, in some sense, to) global minima

Convergence of gAM for convex problems

Things are (again) nice for convex problems

- for differentiable (jointly) convex functions, all bistable points are global minima, so any one is good enough
- **(Block) Coordinate Minimization** approach: solve a single p-dimensional variable $x \in \mathbb{R}^p$as p one-dimensional variables $\{x_1, \dots, x_p\}$, useful in large-scale convex optimization
- may not work well for **non-differentiable** optimization problems

For **non-convex** problems:

- we can only converge to bistable points, and hope they are (or at least close, in some sense, to) global minima

Convergence of gAM for convex problems

Things are (again) nice for convex problems

- for differentiable (jointly) convex functions, all bistable points are global minima, so any one is good enough
- **(Block) Coordinate Minimization** approach: solve a single p-dimensional variable $x \in \mathbb{R}^p$ as p one-dimensional variables $\{x_1, \ldots, x_p\}$, useful in large-scale convex optimization
- may not work well for **non-differentiable** optimization problems

For **non-convex** problems:

- we can only converge to bistable points, and hope they are (or at least close, in some sense, to) global minima

Convergence of gAM for non-convex problems

A point $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is a bistable with respect to a continuously differentiable function $f : \mathbb{R}^p \times \mathbb{R}^q$ that is marginally convex in both its variables **if and only if** $\nabla f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$.

Lemma (Bistable points are stationary points)

A function $f : \mathbb{R}^p \times \mathbb{R}^q$ is said to be C-robust bistable if for some $C > 0$, every $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^p \times \mathbb{R}^q$, $\tilde{\mathbf{x}} \in \mathsf{mOPT}_f(\mathbf{y})$ and $\tilde{\mathbf{y}} \in \mathsf{mOPT}_f(\mathbf{x})$ we have

$$f(\mathbf{x}, \mathbf{y}_*) + f(\mathbf{x}_*, \mathbf{y}) - 2f_* \leq C\left(2f(\mathbf{x}, \mathbf{y}) - f(\mathbf{x}, \tilde{\mathbf{y}}) - f(\tilde{\mathbf{x}}, \mathbf{y})\right), \tag{3}$$

with $(\mathbf{x}_*, \mathbf{y}_*)$ any optimal points with $f(\mathbf{x}_*, \mathbf{y}_*) = f_*$.

Robust Bistability Property

- reduce locally the value of f with marginal optimization
- if no more can be made ($f(\mathbf{x}, \tilde{\mathbf{y}}) \approx f(\mathbf{x}, \mathbf{y}) \approx f(\tilde{\mathbf{x}}, \mathbf{y})$), close to the optimum

Convergence of gAM for non-convex problems

A point $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is a bistable with respect to a continuously differentiable function $f : \mathbb{R}^p \times \mathbb{R}^q$ that is marginally convex in both its variables **if and only if** $\nabla f(\mathbf{x}, \mathbf{y}) = \mathbf{0}$.

Lemma (Bistable points are stationary points)

A function $f : \mathbb{R}^p \times \mathbb{R}^q$ is said to be C-robust bistable if for some $C > 0$, every $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^p \times \mathbb{R}^q$, $\tilde{\mathbf{x}} \in \mathsf{mOPT}_f(\mathbf{y})$ and $\tilde{\mathbf{y}} \in \mathsf{mOPT}_f(\mathbf{x})$ we have

$$f(\mathbf{x}, \mathbf{y}_*) + f(\mathbf{x}_*, \mathbf{y}) - 2f_* \leq C\left(2f(\mathbf{x}, \mathbf{y}) - f(\mathbf{x}, \tilde{\mathbf{y}}) - f(\tilde{\mathbf{x}}, \mathbf{y})\right), \tag{3}$$

with $(\mathbf{x}_*, \mathbf{y}_*)$ any optimal points with $f(\mathbf{x}_*, \mathbf{y}_*) = f_*$.

Robust Bistability Property

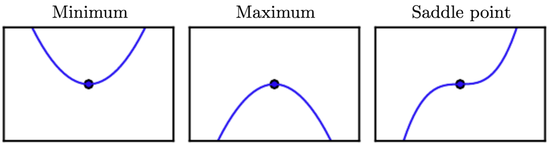
- » reduce locally the value of f with marginal optimization
- » if no more can be made ($f(\mathbf{x}, \tilde{\mathbf{y}}) \approx f(\mathbf{x}, \mathbf{y}) \approx f(\tilde{\mathbf{x}}, \mathbf{y})$), close to the optimum

The Expectation Maximization (EM) algorithm

Very important and interesting,
but skipped here due to time constraint and its different form, see [1, Chapter 5]!

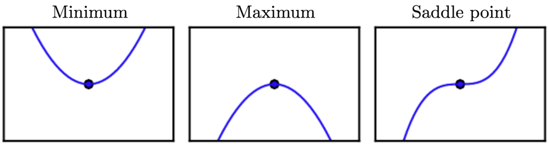
Stochastic Optimization Techniques

- in (ML and SP) applications, objectives functions can be non-convex as well
- gradient descent $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \nabla f(\mathbf{x}(t))$ stalls at **stationary points** with $\nabla f(\mathbf{x}(t)) = 0$
  - local minima, $\nabla^2 f(\mathbf{x}) \succ 0$
  - local maxima, $\nabla^2 f(\mathbf{x}) \prec 0$
  - saddle points contains both positive and negative eigenvalues: we do **not** know, but important, since they are **many** of them



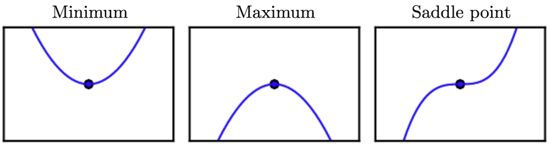
Stochastic Optimization Techniques

- in (ML and SP) applications, objectives functions can be non-convex as well
- gradient descent $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \nabla f(\mathbf{x}(t))$ stalls at **stationary points** with $\nabla f(\mathbf{x}(t)) = 0$
  - local minima, $\nabla^2 f(\mathbf{x}) \succ 0$
  - local maxima, $\nabla^2 f(\mathbf{x}) \prec 0$
  - saddle points contains both positive and negative eigenvalues: we do **not** know, but important, since they are **many** of them



Stochastic Optimization Techniques

- in (ML and SP) applications, objectives functions can be non-convex as well
- gradient descent $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \nabla f(\mathbf{x}(t))$ stalls at **stationary points** with $\nabla f(\mathbf{x}(t)) = 0$
  - local minima, $\nabla^2 f(\mathbf{x}) \succ 0$
  - local maxima, $\nabla^2 f(\mathbf{x}) \prec 0$
  - saddle points contains both positive and negative eigenvalues: we do **not** know, but important, since they are **many** of them

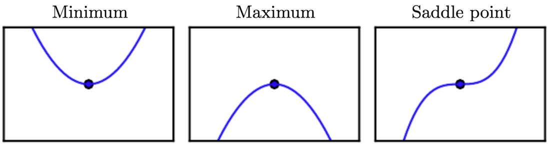


Stochastic Optimization Techniques

» in (ML and SP) applications, objectives functions can be non-convex as well

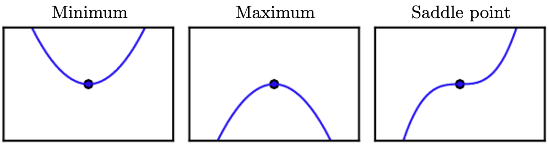
» gradient descent $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \nabla f(\mathbf{x}(t))$ stalls at **stationary points** with $\nabla f(\mathbf{x}(t)) = 0$

- local minima, $\nabla^2 f(\mathbf{x}) \succ 0$
- local maxima, $\nabla^2 f(\mathbf{x}) \prec 0$
- saddle points contains both positive and negative eigenvalues: we do **not** know, but important, since they are **many** of them



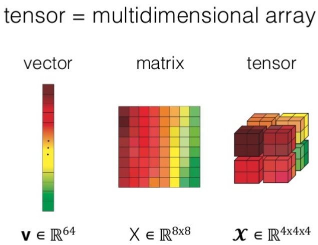
Stochastic Optimization Techniques

- in (ML and SP) applications, objectives functions can be non-convex as well
- gradient descent $\mathbf{x}(t+1) = \mathbf{x}(t) - \eta_t \nabla f(\mathbf{x}(t))$ stalls at **stationary points** with $\nabla f(\mathbf{x}(t)) = 0$
  - local minima, $\nabla^2 f(\mathbf{x}) \succ 0$
  - local maxima, $\nabla^2 f(\mathbf{x}) \prec 0$
  - saddle points contains both positive and negative eigenvalues: we do **not** know, but important, since they are **many** of them



Motivating example: Orthogonal Tensor Decomposition

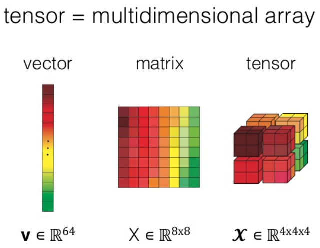
- use outer product $\otimes$ to construct 2nd order tensor, for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p$, $\mathbf{u} \otimes \mathbf{v} \equiv \mathbf{u}\mathbf{v}^\top \in \mathbb{R}^{p \times p}$
- 4th-order tensor (4-dimensional array) that has orthogonal decomposition $\mathcal{T} = \sum_{i=1}^{r} \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i$, with $\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$ (orthonormal)



Motivating example: Orthogonal Tensor Decomposition

» use outer product $\otimes$ to construct 2nd order tensor, for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p$, $\mathbf{u} \otimes \mathbf{v} \equiv \mathbf{u}\mathbf{v}^\top \in \mathbb{R}^{p \times p}$

» 4th-order tensor (4-dimensional array) that has orthogonal decomposition $\mathcal{T} = \sum_{i=1}^r \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i$, with $\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$ (orthonormal)



» like a matrix $\in \mathbb{R}^{p\times p}$ defines a bilinear form $\mathbf{A}: (\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x}^\top \mathbf{A}\mathbf{y}$, (orthonormal) tensor defines multi-linear form as

$$\mathcal{T}(\mathbf{v},\mathbf{v},\mathbf{v},\mathbf{v}) = \sum_{i=1}^{r} (\mathbf{u}_i^\top \mathbf{v})^4 \in \mathbb{R}, \ldots, \mathcal{T}(I,I,I,\mathbf{v}) = \sum_{i=1}^{r} (\mathbf{u}_i^\top \mathbf{v}) \cdot (\mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i) \in \mathbb{R}^{p\times p\times p}$$

» the problem of tensor decomposition: recover all $\mathbf{u}_i$, $i = 1, \ldots, r$, do this one by one

» in need to solve $\max_{\|\mathbf{u}\|=1} \mathcal{T}(\mathbf{u},\mathbf{u},\mathbf{u},\mathbf{u}) = \sum_{i=1}^{n} (\mathbf{u}_i^\top \mathbf{u})^4$

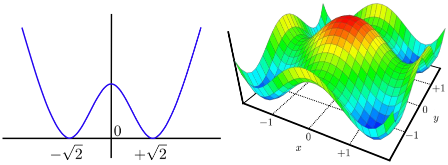


Figure 6.1: The function on the left $f(x) = x^4 - 4 \cdot x^2 + 4$ has two global optima $\{-\sqrt{2}, \sqrt{2}\}$ separated by a local maxima at 0. Using this function, we construct on the right, a higher dimensional function $g(x, y) = f(x) + f(y) + 8$ which now has 4 global minima separated by 4 saddle points. The number of such minima and saddle points can explode exponentially in learning problems with symmetry (indeed $g(x, y, z) = f(x)+f(y)+f(z)+12$ has 8 local minima and saddle points). Plot on the right courtesy `academo.org`

- like a matrix $\in \mathbb{R}^{p\times p}$ defines a bilinear form $\mathbf{A}\colon (\mathbf{x},\mathbf{y}) \mapsto \mathbf{x}^\top \mathbf{A}\mathbf{y}$, (orthonormal) tensor defines multi-linear form as

$$\mathcal{T}(\mathbf{v},\mathbf{v},\mathbf{v},\mathbf{v}) = \sum_{i=1}^{r} (\mathbf{u}_i^\top \mathbf{v})^4 \in \mathbb{R}, \ldots, \mathcal{T}(I,I,I,\mathbf{v}) = \sum_{i=1}^{r} (\mathbf{u}_i^\top \mathbf{v}) \cdot (\mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i) \in \mathbb{R}^{p\times p\times p}$$

- the problem of tensor decomposition: recover all $\mathbf{u}_i$, $i = 1,\ldots,r$, do this one by one
- in need to solve $\max_{\|\mathbf{u}\|=1} \mathcal{T}(\mathbf{u},\mathbf{u},\mathbf{u},\mathbf{u}) = \sum_{i=1}^{n} (\mathbf{u}_i^\top \mathbf{u})^4$

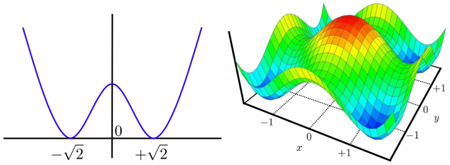


Figure 6.1: The function on the left $f(x) = x^4 - 4\cdot x^2 + 4$ has two global optima $\{-\sqrt{2}, \sqrt{2}\}$ separated by a local maxima at 0. Using this function, we construct on the right, a higher dimensional function $g(x,y) = f(x) + f(y) + 8$ which now has 4 global minima separated by 4 saddle points. The number of such minima and saddle points can explode exponentially in learning problems with symmetry (indeed $g(x,y,z) = f(x)+f(y)+f(z)+12$ has 8 local minima and saddle points). Plot on the right courtesy `academo.org`

- like a matrix $\in \mathbb{R}^{p\times p}$ defines a bilinear form $\mathbf{A}\colon (\mathbf{x},\mathbf{y}) \mapsto \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{y}$, (orthonormal) tensor defines multi-linear form as

$$\mathcal{T}(\mathbf{v},\mathbf{v},\mathbf{v},\mathbf{v}) = \sum_{i=1}^{r} (\mathbf{u}_i^\mathsf{T}\mathbf{v})^4 \in \mathbb{R}, \ldots, \mathcal{T}(I,I,I,\mathbf{v}) = \sum_{i=1}^{r} (\mathbf{u}_i^\mathsf{T}\mathbf{v}) \cdot (\mathbf{u}_i \otimes \mathbf{u}_i \otimes \mathbf{u}_i) \in \mathbb{R}^{p\times p\times p}$$

- the problem of tensor decomposition: recover all $\mathbf{u}_i$, $i = 1,\ldots,r$, do this one by one
- in need to solve $\underline{\max_{\|\mathbf{u}\|=1} \mathcal{T}(\mathbf{u},\mathbf{u},\mathbf{u},\mathbf{u}) = \sum_{i=1}^{n} (\mathbf{u}_i^\mathsf{T}\mathbf{u})^4}$

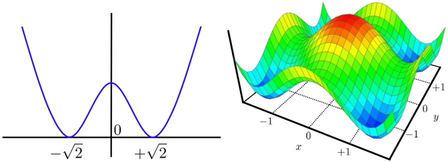


Figure 6.1: The function on the left $f(x) = x^4 - 4\cdot x^2 + 4$ has two global optima $\{-\sqrt{2}, \sqrt{2}\}$ separated by a local maxima at 0. Using this function, we construct on the right, a higher dimensional function $g(x,y) = f(x) + f(y) + 8$ which now has 4 global minima separated by 4 saddle points. The number of such minima and saddle points can explode exponentially in learning problems with symmetry (indeed $g(x,y,z) = f(x)+f(y)+f(z)+12$ has 8 local minima and saddle points). Plot on the right courtesy `academo.org`

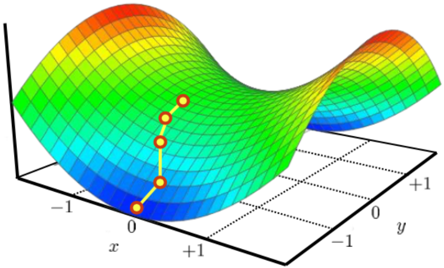
- in the case of Orthogonal Tensor Decomposition, symmetry in problem:
  - recover the components in any order we like (a lot of equivalent global optima); but
  - convex combinations of the components are **not** optima: in fact, r isolated optima spread out in space, interspersed with **saddle points** (just like in the pictures)
- In this case, what should we do?
  - (i) apply second-order (e.g., Newton's method) to "escape" from saddle points: this is however **not** always possible due to high complexity
  - (ii) what to do if we are only allowed to use first-order methods? **Add some noise!**
- **intuition**: if a saddle point $\mathbf{x}$ of f contains direction of steep gradient, then there is some chance for gradient descent to "discover" and "fall" along it

- in the case of Orthogonal Tensor Decomposition, symmetry in problem:
  - recover the components in any order we like (a lot of equivalent global optima); but
  - convex combinations of the components are **not** optima: in fact, r isolated optima spread out in space, interspersed with **saddle points** (just like in the pictures)
- In this case, what should we do?
  (i) apply second-order (e.g., Newton's method) to "escape" from saddle points: this is however **not** always possible due to high complexity
  (ii) what to do if we are only allowed to use first-order methods? **Add some noise!**
- **intuition**: if a saddle point $\mathbf{x}$ of f contains direction of steep gradient, then there is some chance for gradient descent to "discover" and "fall" along it

- in the case of Orthogonal Tensor Decomposition, symmetry in problem:
  - recover the components in any order we like (a lot of equivalent global optima); but
  - convex combinations of the components are **not** optima: in fact, r isolated optima spread out in space, interspersed with **saddle points** (just like in the pictures)
- In this case, what should we do?
  (i) apply second-order (e.g., Newton's method) to "escape" from saddle points: this is however **not** always possible due to high complexity
  (ii) what to do if we are only allowed to use first-order methods? **Add some noise!**
- **intuition**: if a saddle point $\mathbf{x}$ of f contains direction of steep gradient, then there is some chance for gradient descent to "discover" and "fall" along it

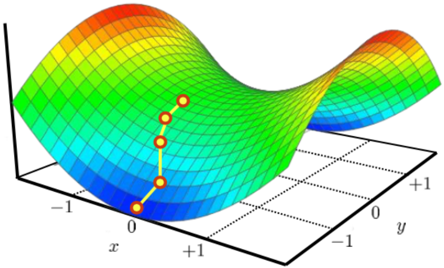
Intuition of noisy gradient descent

- existence of the steep gradient direction makes the saddle unstable and behaves like a **local maxima** along this direction
- so slight perturbation of the gradient may cause gradient descent to roll down
- see below for the two-dimensional toy example of $f(x, y) = x^2 - y^2$, with saddle at $(0, 0)$ and minimum Hessian eigenvalue $= -2$



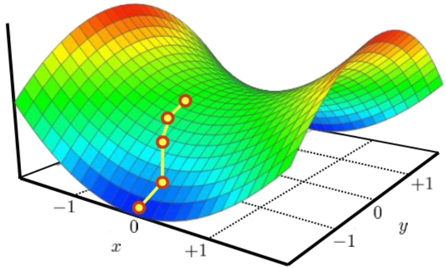
Intuition of noisy gradient descent

- existence of the steep gradient direction makes the saddle unstable and behaves like a **local maxima** along this direction
- so slight perturbation of the gradient may cause gradient descent to roll down
- see below for the two-dimensional toy example of $f(x, y) = x^2 - y^2$, with saddle at $(0, 0)$ and minimum Hessian eigenvalue $= -2$



Intuition of noisy gradient descent

- existence of the steep gradient direction makes the saddle unstable and behaves like a **local maxima** along this direction
- so slight perturbation of the gradient may cause gradient descent to roll down
- see below for the two-dimensional toy example of $f(x, y) = x^2 - y^2$, with saddle at $(0, 0)$ and minimum Hessian eigenvalue $= -2$



The strict saddle property

For unconstrained optimization problem $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$, we say $f : \mathbb{R}^p \to \mathbb{R}$ satisfy the strict saddle property if, for **every** point $\mathbf{x} \in \mathbb{R}^p$ we have at least one the following properties holds:

- **Non-stationary point**: $\|\nabla f(\mathbf{x})\| \geq C_1$;
- **Strict saddle point**: $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \leq -C_2$;
- **Approximate local minimum**: $\|\mathbf{x} - \mathbf{x}_*\| \geq C_3$ for some local minimum $\mathbf{x}_*$.

for some $C_1, C_2, C_3 > 0$.

Strict saddle property

- assume the property of f and is in fact **very restrictive**
- however, there does exist such problem in practice: e.g., the Orthogonal Tensor Decomposition

The strict saddle property

For unconstrained optimization problem $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$, we say $f : \mathbb{R}^p \to \mathbb{R}$ satisfy the strict saddle property if, for **every** point $\mathbf{x} \in \mathbb{R}^p$ we have at least one the following properties holds:

- **Non-stationary point**: $\|\nabla f(\mathbf{x})\| \geq C_1$;
- **Strict saddle point**: $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \leq -C_2$;
- **Approximate local minimum**: $\|\mathbf{x} - \mathbf{x}_*\| \geq C_3$ for some local minimum $\mathbf{x}_*$.

for some $C_1, C_2, C_3 > 0$.

Strict saddle property

- assume the property of f and is in fact **very restrictive**
- however, there does exist such problem in practice: e.g., the Orthogonal Tensor Decomposition

The strict saddle property

For unconstrained optimization problem $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$, we say $f : \mathbb{R}^p \to \mathbb{R}$ satisfy the strict saddle property if, for **every** point $\mathbf{x} \in \mathbb{R}^p$ we have at least one the following properties holds:

- **Non-stationary point**: $\|\nabla f(\mathbf{x})\| \geq C_1$;
- **Strict saddle point**: $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \leq -C_2$;
- **Approximate local minimum**: $\|\mathbf{x} - \mathbf{x}_*\| \geq C_3$ for some local minimum $\mathbf{x}_*$.

for some $C_1, C_2, C_3 > 0$.

Strict saddle property

- assume the property of f and is in fact **very restrictive**
- however, there does exist such problem in practice: e.g., the Orthogonal Tensor Decomposition

The strict saddle property

For unconstrained optimization problem $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$, we say $f : \mathbb{R}^p \to \mathbb{R}$ satisfy the strict saddle property if, for **every** point $\mathbf{x} \in \mathbb{R}^p$ we have at least one the following properties holds:

- **Non-stationary point**: $\|\nabla f(\mathbf{x})\| \geq C_1$;
- **Strict saddle point**: $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \leq -C_2$;
- **Approximate local minimum**: $\|\mathbf{x} - \mathbf{x}_*\| \geq C_3$ for some local minimum $\mathbf{x}_*$.

for some $C_1, C_2, C_3 > 0$.

Strict saddle property

- assume the property of f and is in fact **very restrictive**
- however, there does exist such problem in practice: e.g., the Orthogonal Tensor Decomposition

The strict saddle property

For unconstrained optimization problem $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$, we say $f : \mathbb{R}^p \to \mathbb{R}$ satisfy the strict saddle property if, for **every** point $\mathbf{x} \in \mathbb{R}^p$ we have at least one the following properties holds:

- **Non-stationary point**: $\|\nabla f(\mathbf{x})\| \geq C_1$;
- **Strict saddle point**: $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \leq -C_2$;
- **Approximate local minimum**: $\|\mathbf{x} - \mathbf{x}_*\| \geq C_3$ for some local minimum $\mathbf{x}_*$.

for some $C_1, C_2, C_3 > 0$.

Strict saddle property

- assume the property of f and is in fact **very restrictive**
- however, there does exist such problem in practice: e.g., the Orthogonal Tensor Decomposition

Noisy Gradient Descent (NGD)

Algorithm Noisy Gradient Descent (NGD)

Input: Objective f, max step length $\eta_{\max}$, tolerance ϵ
Output: A locally optimal point $\hat{\mathbf{x}} \in \mathbb{R}^p$

1: $\mathbf{x}(0) \leftarrow \mathsf{INIT}()$
2: Set $T \leftarrow 1/\eta^2$, where $\eta = \min\left\{\epsilon^2/\log^2(1/\epsilon), \eta_{\max}\right\}$
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Sample perturbation $\mathbf{z}(t) \sim \mathbb{S}^{p-1}$ `//Random pt. on unit sphere`
5: $\mathbf{g}(t) \leftarrow \nabla f(\mathbf{x}(t)) + \mathbf{z}(t)$
6: $\mathbf{x}(t+1) \leftarrow \mathbf{x}(t) - \eta \cdot \mathbf{g}(t)$
7: **end for**
8: **return** $\mathbf{x}(T)$

A few comments on NGD

- At **each** step, perturbs the gradient using a unit vector pointing at a **random** direction, to continue to make progress even at saddle points
- for standard Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, take $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- we have $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ so that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$ unbiased estimate of true gradient (common and important in stochastic optimization scheme)
- a side remark: the step length $\eta \approx 1/\sqrt{T}$ as in the horizon-aware setting of PGD, essentially for the sake of proof

In case of a constrained optimization with non-convex objective:

- use **Projected Noisy Gradient Descent**
- in fact applies to the Orthogonal Tensor Decomposition problem (which can be shown, with some tedious calculations, to satisfy the **strict saddle** property)

A few comments on NGD

- At **each** step, perturbs the gradient using a unit vector pointing at a **random** direction, to continue to make progress even at saddle points
- for standard Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, take $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- we have $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ so that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$ unbiased estimate of true gradient (common and important in stochastic optimization scheme)
- a side remark: the step length $\eta \approx 1/\sqrt{T}$ as in the horizon-aware setting of PGD, essentially for the sake of proof

In case of a constrained optimization with non-convex objective:

- use **Projected Noisy Gradient Descent**
- in fact applies to the Orthogonal Tensor Decomposition problem (which can be shown, with some tedious calculations, to satisfy the **strict saddle** property)

A few comments on NGD

- At **each** step, perturbs the gradient using a unit vector pointing at a **random** direction, to continue to make progress even at saddle points
- for standard Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, take $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- we have $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ so that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$ unbiased estimate of true gradient (common and important in stochastic optimization scheme)
- a side remark: the step length $\eta \approx 1/\sqrt{T}$ as in the horizon-aware setting of PGD, essentially for the sake of proof

In case of a constrained optimization with non-convex objective:

- use **Projected Noisy Gradient Descent**
- in fact applies to the Orthogonal Tensor Decomposition problem (which can be shown, with some tedious calculations, to satisfy the **strict saddle** property)

A few comments on NGD

- At **each** step, perturbs the gradient using a unit vector pointing at a **random** direction, to continue to make progress even at saddle points
- for standard Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, take $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- we have $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ so that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$ unbiased estimate of true gradient (common and important in stochastic optimization scheme)
- a side remark: the step length $\eta \approx 1/\sqrt{T}$ as in the horizon-aware setting of PGD, essentially for the sake of proof

In case of a constrained optimization with non-convex objective:

- use **Projected Noisy Gradient Descent**
- in fact applies to the Orthogonal Tensor Decomposition problem (which can be shown, with some tedious calculations, to satisfy the **strict saddle** property)

A few comments on NGD

- At **each** step, perturbs the gradient using a unit vector pointing at a **random** direction, to continue to make progress even at saddle points
- for standard Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, take $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- we have $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ so that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$ unbiased estimate of true gradient (common and important in stochastic optimization scheme)
- a side remark: the step length $\eta \approx 1/\sqrt{T}$ as in the horizon-aware setting of PGD, essentially for the sake of proof

In case of a constrained optimization with non-convex objective:

- use **Projected Noisy Gradient Descent**
- in fact applies to the Orthogonal Tensor Decomposition problem (which can be shown, with some tedious calculations, to satisfy the **strict saddle** property)

A few comments on NGD

- At **each** step, perturbs the gradient using a unit vector pointing at a **random** direction, to continue to make progress even at saddle points
- for standard Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, take $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
- we have $\mathbb{E}[\mathbf{z}] = \mathbf{0}$ so that $\mathbb{E}[\mathbf{g}|\mathbf{x}] = \nabla f(\mathbf{x})$ unbiased estimate of true gradient (common and important in stochastic optimization scheme)
- a side remark: the step length $\eta \approx 1/\sqrt{T}$ as in the horizon-aware setting of PGD, essentially for the sake of proof

In case of a constrained optimization with non-convex objective:

- use **Projected Noisy Gradient Descent**
- in fact applies to the Orthogonal Tensor Decomposition problem (which can be shown, with some tedious calculations, to satisfy the **strict saddle** property)

Outline

Introduction and basic concepts

Some Non-convex Optimization Methods

Applications

Sparse Regression

$$\boxed{\min_{\mathbf{w}\in\mathbb{R}^p, \|\mathbf{w}\|_0 \le s} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|_2} \tag{4}$$

with some (given) $\mathbf{X} \in \mathbb{R}^{p\times n}$, $\mathbf{y} \in \mathbb{R}^n$, and sparsity constraint $s > 0$.

» known to be an NP-hard problem

» can be solved via PGD, which, in the setting, known as **Iterative Hard-thresholding** if, the problem (e.g., the sensing matrix $\mathbf{X}$) is nice enough: nullspace property, restricted eigenvalue property, Restricted Isometry Property (RIP), etc.

- random design (i.i.d. Gaussian, Bernoulli entries)
- deterministic design: incoherent matrix

Sparse Regression

$$\boxed{\min_{\mathbf{w}\in\mathbb{R}^p, \|\mathbf{w}\|_0 \le s} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|_2} \tag{4}$$

with some (given) $\mathbf{X} \in \mathbb{R}^{p\times n}$, $\mathbf{y} \in \mathbb{R}^n$, and sparsity constraint $s > 0$.

» known to be an NP-hard problem

» can be solved via PGD, which, in the setting, known as **Iterative Hard-thresholding** if, the problem (e.g., the sensing matrix $\mathbf{X}$) is nice enough: nullspace property, restricted eigenvalue property, Restricted Isometry Property (RIP), etc.

- random design (i.i.d. Gaussian, Bernoulli entries)
- deterministic design: incoherent matrix

Sparse Regression

$$\boxed{\min_{\mathbf{w}\in\mathbb{R}^p, \|\mathbf{w}\|_0\le s} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|_2} \tag{4}$$

with some (given) $\mathbf{X} \in \mathbb{R}^{p\times n}$, $\mathbf{y} \in \mathbb{R}^n$, and sparsity constraint $s > 0$.

- known to be an NP-hard problem
- can be solved via PGD, which, in the setting, known as **Iterative Hard-thresholding** if, the problem (e.g., the sensing matrix $\mathbf{X}$) is nice enough: nullspace property, restricted eigenvalue property, Restricted Isometry Property (RIP), etc.
  - random design (i.i.d. Gaussian, Bernoulli entries)
  - deterministic design: incoherent matrix

Sparse Regression

$$\boxed{\min_{\mathbf{w}\in\mathbb{R}^p, \|\mathbf{w}\|_0 \le s} \|\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}\|_2} \tag{4}$$

with some (given) $\mathbf{X} \in \mathbb{R}^{p\times n}$, $\mathbf{y} \in \mathbb{R}^n$, and sparsity constraint $s > 0$.

» known to be an NP-hard problem

» can be solved via PGD, which, in the setting, known as **Iterative Hard-thresholding** if, the problem (e.g., the sensing matrix $\mathbf{X}$) is nice enough: nullspace property, restricted eigenvalue property, Restricted Isometry Property (RIP), etc.

- random design (i.i.d. Gaussian, Bernoulli entries)
- deterministic design: incoherent matrix

Iterative Hard-thresholding

Algorithm Iterative Hard-thresholding (IHT)

Input: Data $\mathbf{X}, \mathbf{y}$, step length η, projection sparsity level k
Output: A sparse model $\hat{\mathbf{w}} \in \mathcal{B}_0(k)$

1: $\mathbf{w}(0) \leftarrow \mathbf{0}$
2: **for** $t = 1, 2, \ldots,$ **do**
3: $\quad \mathbf{z}(t+1) \leftarrow \mathbf{w}(t) - \eta \cdot \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{w}(t) - \mathbf{y})$
4: $\quad \mathbf{w}(t+1) \leftarrow \Pi_{\mathcal{B}_0(k)}(\mathbf{z}(t+1))$ `//in fact, sorting`
5: **end for**
6: **return** $\mathbf{w}(t)$

A few comments on Sparse Recovery

Other popular techniques for Sparse Recovery

- **hard thresholding** techniques: IHT, Gradient Descent with Sparsification (GraDeS), and Hard Thresholding Pursuit (HTP)
- **pursuit** techniques: discover support elements iteratively: at each time step, add a new support element to an active support set (empty when initialized) and solve a traditional least-squares (with **no** sparsity constraint, convex and easy) problem on the active set
- **convex relaxation**: relax the L_0 norm to L_1 norm, solve the so-called LASSO problem, in nice cases (e.g., RIP), can find optimal solution

A few comments on Sparse Recovery

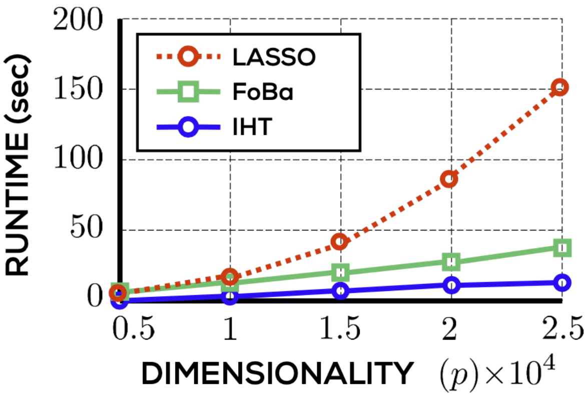
Other popular techniques for Sparse Recovery

- **hard thresholding** techniques: IHT, Gradient Descent with Sparsification (GraDeS), and Hard Thresholding Pursuit (HTP)
- **pursuit** techniques: discover support elements iteratively: at each time step, add a new support element to an active support set (empty when initialized) and solve a traditional least-squares (with **no** sparsity constraint, convex and easy) problem on the active set
- **convex relaxation**: relax the L_0 norm to L_1 norm, solve the so-called LASSO problem, in nice cases (e.g., RIP), can find optimal solution

A few comments on Sparse Recovery

Other popular techniques for Sparse Recovery

- **hard thresholding** techniques: IHT, Gradient Descent with Sparsification (GraDeS), and Hard Thresholding Pursuit (HTP)
- **pursuit** techniques: discover support elements iteratively: at each time step, add a new support element to an active support set (empty when initialized) and solve a traditional least-squares (with **no** sparsity constraint, convex and easy) problem on the active set
- **convex relaxation**: relax the L_0 norm to L_1 norm, solve the so-called LASSO problem, in nice cases (e.g., RIP), can find optimal solution


200
150
100
50
0
RUNTIME (sec)
LASSO
FoBa
IHT
0.5
1
1.5
2
2.5
DIMENSIONALITY $(p) \times 10^4$

Low-rank Matrix Completion

$$\boxed{\min_{\mathbf{X}\in\mathbb{R}^{m\times n},\ \operatorname{rank}(\mathbf{X})\le r} \|\Pi_\Omega(\mathbf{X}-\mathbf{X}_*)\|_F^2} \tag{5}$$

with an "observation" projection $\Pi_\Omega(\mathbf{X})$ defined as

$$[\Pi_\Omega(\mathbf{X})]_{ij} = \begin{cases} X_{ij} & \text{if } (i,j)\in\Omega \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

» is a special case of the (or **Affine Rank Minimization (ARM)**)

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \quad & \|\mathcal{A}(\mathbf{X})-\mathbf{y}\|_2^2 \\ \text{s.t.} \quad & \operatorname{rank}(\mathbf{X}) = r, \end{aligned}$$

with affine transformation $\mathcal{A}_{(i,j)}: \mathbf{X} \mapsto \operatorname{tr}(\mathbf{X}^\mathsf{T}\mathbf{E}_{ij}) = X_{ij}$

» can be solved with PGD

Low-rank Matrix Completion

$$\boxed{\min_{\mathbf{X}\in\mathbb{R}^{m\times n},\ \mathrm{rank}(\mathbf{X})\le r} \|\Pi_\Omega(\mathbf{X}-\mathbf{X}_*)\|_F^2} \tag{5}$$

with an "observation" projection $\Pi_\Omega(\mathbf{X})$ defined as

$$[\Pi_\Omega(\mathbf{X})]_{ij} = \begin{cases} X_{ij} & \text{if } (i,j)\in\Omega \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

- » is a special case of the (or **Affine Rank Minimization (ARM)**)

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \quad & \|\mathcal{A}(\mathbf{X})-\mathbf{y}\|_2^2 \\ \text{s.t.} \quad & \mathrm{rank}(\mathbf{X}) = r, \end{aligned}$$

  with affine transformation $\mathcal{A}_{(i,j)} : \mathbf{X} \mapsto \mathrm{tr}(\mathbf{X}^\mathsf{T}\mathbf{E}_{ij}) = X_{ij}$
- » can be solved with PGD

Singular Value Projection (SVP)

Algorithm Singular Value Projection (SVP)

Input: Linear map $\mathcal{A}(\cdot)$, measurements $\mathbf{y}$, target rank q, step length η
Output: A matrix $\hat{\mathbf{X}}$ with rank at most q

```
X(0) ← 0_{m×n}
for t = 1, 2, ... do
    Y(t+1) ← X(t) − η · A^T(A(X(t)) − y)
    Compute top q singular vectors/values of Y(t+1) to get U_q(t), Σ_q(t), V_q(t)
    X(t+1) ← U_q(t) Σ_q(t) V_q^T(t)
end for
return X(t)
```

$\mathbf{X}(0) \leftarrow \mathbf{0}_{m\times n}$
for $t = 1, 2, \ldots$ **do**
  $\mathbf{Y}(t+1) \leftarrow \mathbf{X}(t) - \eta \cdot \mathcal{A}^\mathsf{T}(\mathcal{A}(\mathbf{X}(t)) - \mathbf{y})$
  Compute top q singular vectors/values of $\mathbf{Y}(t+1)$ to get $\mathbf{U}_q(t), \mathbf{\Sigma}_q(t), \mathbf{V}_q(t)$
  $\mathbf{X}(t+1) \leftarrow \mathbf{U}_q(t)\mathbf{\Sigma}_q(t)\mathbf{V}_q^\mathsf{T}(t)$
end for
return $\mathbf{X}(t)$

A few comments on Low-rank Matrix Completion

- again, as we should have expected, SVP works when things are nice enough (e.g., matrix RIP)
- we can alternatively use alternative minimization to solve

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times k},\mathbf{V}\in\mathbb{R}^{n\times k}} \|\Pi_\Omega(\mathbf{U}\mathbf{V}^\top - \mathbf{X}_*)\|_F^2. \tag{7}$$

- initializations are very important, since one in general has only convergence for non-convex problems

A few comments on Low-rank Matrix Completion

- again, as we should have expected, SVP works when things are nice enough (e.g., matrix RIP)
- we can alternatively use alternative minimization to solve

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times k},\mathbf{V}\in\mathbb{R}^{n\times k}} \|\Pi_\Omega(\mathbf{U}\mathbf{V}^\mathsf{T} - \mathbf{X}_*)\|_F^2. \tag{7}$$

- initializations are very important, since one in general has only convergence for non-convex problems

A few comments on Low-rank Matrix Completion

- again, as we should have expected, SVP works when things are nice enough (e.g., matrix RIP)
- we can alternatively use alternative minimization to solve

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times k},\mathbf{V}\in\mathbb{R}^{n\times k}} \|\Pi_\Omega(\mathbf{U}\mathbf{V}^\top - \mathbf{X}_*)\|_F^2. \tag{7}$$

- initializations are very important, since one in general has only convergence for non-convex problems