6.7220 / 15.084: Recitation 1: Convex Functions and Sets

Shuvomoy Das Gupta

MIT

February 17, 2023

1. Introduction

2. How to recognize convexity?

Why study convex sets or functions?

- We know how to solve convex optimization problems "efficiently"

Why study convex sets or functions?

- We know how to solve convex optimization problems "efficiently"
- These problems are called *disciplined convex program* and have the form:

$$\begin{array}{lll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) & \triangleright\, f_0 : \texttt{convex} \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, & \triangleright\, f_i : \texttt{convex} \\ & h_i(x) = 0, \quad i = 1, \ldots, p. & \triangleright\, h_i : \texttt{affine} \end{array}$$

Why study convex sets or functions?

- ▷ We know how to solve convex optimization problems "efficiently"
- ▷ These problems are called *disciplined convex program* and have the form:

$$\begin{array}{lllr} \underset{x\in\mathbb{R}^d}{\text{minimize}} & f_0(x) & & \triangleright\, f_0 : \texttt{convex} \\ \text{subject to} & f_i(x) \le 0, & i = 1,\ldots,m, & \triangleright\, f_i : \texttt{convex} \\ & h_i(x) = 0, & i = 1,\ldots,p. & \triangleright\, h_i : \texttt{affine} \end{array}$$

- ▷ Most real-world problems are nonconvex
  - We either approximate them as convex problems
  - We solve a sequence of increasingly accurate convex problems
  - We "hope" that the problem is locally convex and apply convex optimization algorithms

A brief history of convex optimization

- 1947: G. Dantzig, who works for US air-forces, presents the Simplex method for solving LP-problems
- 1948: J. Von Neumann establishes the theory of duality for LP-problems
- 1951: H.W. Kuhn and A.W. Tucker reinvent Karush's optimality conditions (known as KKT conditions)
- 1951: H. Markowitz presents his portfolio optimization theory => (1990 Nobel prize)
- 1954: L.R. Ford's and D.R. Fulkerson's research on network problems

A brief history of convex optimization

- 1960-1970: Many of the early works on first-order optimization algorithms are done (mostly developed in Soviet Union)
- 1983: Nesterov comes up with accelerated gradient descent
- 1984: N. Karmarkar's polynomial time algorithm for LP-problems begins a boom period for interior point methods
- 1990s: Semidefinite optimization theory
- 2010-present: First-order methods become very hot again due to machine learning
- 2014: Performance estimation problem: computer-assisted design and analysis of optimization algorithms

1. Introduction

2. How to recognize convexity?

How can you tell if a problem is convex?

- Need to check convexity of a function f
- **Approaches:**
- use basic definition
- first or second order conditions, e.g., $\nabla^2 f(x) \succeq 0$
- via convex calculus: construct f using
  - library of basic examples or atoms that are convex
  - calculus rules or transformations that preserve convexity

Basic convex functions (convex atoms)

- x^p for $p \geq 1$ or $p \leq 0$; $-x^p$ for $0 \leq p \leq 1$ when $x > 0$
- e^{ax} for any a,$-\log x$ for $x > 0$, $x \log x$ for $x > 0$
- $a^\top x + b$
- $x^\top x$; $x^\top x/y$ (for $y > 0$); $\sqrt{x^\top x}$
- $\|x\|$ (any norm)
- $\max(x_1, \ldots, x_n)$
- $\log(e^{x_1} + \ldots + e^{x_n})$
- $\log \det X^{-1}$ (for $X \succ 0$)
- These are also called *atoms* because they are building block of much more complex convex functions. There are many such atoms, most convex programs in practice can be built from these atoms. A more complete list can be found at
  - `https://jump.dev/Convex.jl/stable/operations/`.

Convex calculus rules

- nonnegative scaling: if f is convex then αf is convex if $\alpha \geq 0$
- sum: if f and g are convex, then so is $f + g$
- affine composition: if f is convex, then so is $f(Ax + b)$
- pointwise maximum: if $f_1, f_2, \ldots, f_m$ are convex, then so is $f(x) = \max_{i \in \{1,\ldots,m\}} f_i(x)$
- pointwise supremum: if $f(x, y)$ is convex in x for all $y \in S$, then $g(x) = \sup_{y \in S} f(x, y$ is convex
- partial minimization: if $f(x, y)$ is convex in (x, y) and C is convex, then $g(x) = \min_{y \in C} f(x, y)$ is convex
- composition: if h is convex and increasing and f is convex, then $g(x) = h(f(x))$ is convex

Proving convexity via convex calculus

- piecewise-linear function: $f(x) = \max_{i=1,\dots,k}(a_i^\top x + b_i)$
- ℓ_1-regularized least-squares cost: $\|Ax - b\|_2^2 + \lambda \|x\|_1$ with $\lambda \geq 0$
- support-function of a set: $S_C(x) = \max_{y \in C} x^\top y$ where C is any set
- distance to convex set: $f(x) = \min_{y \in C} \|x - y\|_2$

Proving convexity via computer

- The Julia package `Convex.jl` can recognize convexity in a functions if it can be constructed via convex calculus
- Sometimes, `Convex.jl` would not be able to prove convexity, in that case we may have to prove convexity using pen and paper
- One useful approach is “restriction on a line”

“Restriction on a line” approach for proving convexity

Show that $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the single variable function $g_{u,v}(t) = f(u + tv)$ is convex for any $u, v \in \mathbb{R}^n$ (t is a scalar).

“Restriction on a line” approach for proving convexity

Show that $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the single variable function $g_{u,v}(t) = f(u + tv)$ is convex for any $u, v \in \mathbb{R}^n$ (t is a scalar).

Proof:

(f cvx$\Rightarrow g_{u,v}$ cvx)

f is convex if and only if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y), \quad \lambda \in [0,1],\ x, y \in \mathbb{R}^n.$$

In the inequality above, set $x \leftarrow u + t_1 v$, $y \leftarrow u + t_2 v$, then we have

$$\begin{aligned} f(\lambda(u + t_1 v) + (1-\lambda)(u + t_2 v)) &\leq \lambda f(u + t_1 v) + (1-\lambda) f(u + t_2 v) \\ \Leftrightarrow f(u + (\lambda t_1 + (1-\lambda) t_2) v) &= g_{u,v}(\lambda t_1 + (1-\lambda) t_2) \\ &\leq \lambda g_{u,v}(t_1) + (1-\lambda) g_{u,v}(t_2), \end{aligned}$$

which is equivalent to saying that $g_{u,v}$ is convex.

"Restriction on a line" approach for proving convexity

Show that $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the single variable function $g_{u,v}(t) = f(u + tv)$ is convex for any $u, v \in \mathbb{R}^n$.

“Restriction on a line” approach for proving convexity

Show that $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the single variable function $g_{u,v}(t) = f(u + tv)$ is convex for any $u, v \in \mathbb{R}^n$.
Proof: ($g_{u,v}$ cvx $\Rightarrow f$ cvx)
$g_{u,v}$ is convex if and only if

$$g_{u,v}(\lambda t_1 + (1-\lambda)t_2) \leq \lambda g_{u,v}(t_1) + (1-\lambda) g_{u,v}(t_2), \quad \lambda \in [0,1],\, t_1, t_2 \in \mathbb{R}$$

“Restriction on a line” approach for proving convexity

Show that $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the single variable function $g_{u,v}(t) = f(u + tv)$ is convex for any $u, v \in \mathbb{R}^n$.

Proof: ($g_{u,v}$ cvx $\Rightarrow f$ cvx)

$g_{u,v}$ is convex if and only if

$$g_{u,v}(\lambda t_1 + (1-\lambda)t_2) \le \lambda g_{u,v}(t_1) + (1-\lambda)g_{u,v}(t_2), \quad \lambda \in [0,1],\, t_1, t_2 \in \mathbb{R}$$

Now let $t_1 = 1$ and $t_2 = 0$, and set $u \leftarrow x, v \leftarrow y - x$. Then

$$\begin{aligned}
&g_{x,y-x}(\lambda t_1 + (1-\lambda)t_2) \le \lambda g_{x,y-x}(t_1) + (1-\lambda)g_{x,y-x}(t_2) \\
\Leftrightarrow &g_{x,y-x}(\lambda) \le \lambda g_{x,y-x}(1) + (1-\lambda)g_{x,y-x}(0) \qquad \triangleright\, g_{x,y-x}(\lambda) = f(x + \lambda(y-x)) \\
\Leftrightarrow &f(x + \lambda(y-x)) = f(\lambda y + (1-\lambda)x) \le \lambda f(y) + (1-\lambda)f(x).
\end{aligned}$$

The last line means that f is convex on $\mathbb{R}^n$ as x, y could be any points in $\mathbb{R}^n$.

“Restriction on a line” approach for proving convexity

Show that $f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the single variable function $g_{u,v}(t) = f(u + tv)$ is convex for any $u, v \in \mathbb{R}^n$.

Proof: ($g_{u,v}$ cvx $\Rightarrow$ f cvx)

$g_{u,v}$ is convex if and only if

$$g_{u,v}(\lambda t_1 + (1-\lambda)t_2) \leq \lambda g_{u,v}(t_1) + (1-\lambda) g_{u,v}(t_2), \quad \lambda \in [0,1],\, t_1, t_2 \in \mathbb{R}$$

Now let $t_1 = 1$ and $t_2 = 0$, and set $u \leftarrow x, v \leftarrow y - x$. Then

$$\begin{aligned}
&g_{x,y-x}(\lambda t_1 + (1-\lambda)t_2) \leq \lambda g_{x,y-x}(t_1) + (1-\lambda) g_{x,y-x}(t_2) \\
\Leftrightarrow &g_{x,y-x}(\lambda) \leq \lambda g_{x,y-x}(1) + (1-\lambda) g_{x,y-x}(0) \qquad \triangleright\, g_{x,y-x}(\lambda) = f(x + \lambda(y-x)) \\
\Leftrightarrow &f(x + \lambda(y-x)) = f(\lambda y + (1-\lambda)x) \leq \lambda f(y) + (1-\lambda) f(x).
\end{aligned}$$

The last line means that f is convex on $\mathbb{R}^n$ as x, y could be any points in $\mathbb{R}^n$.

- ▷ Note that if the domain of f is not the entire space $\mathbb{R}^n$, you need to show that for any x, y, $g(t) = f(x + ty)$ is convex for all values of t such that x and $x + ty$ are in the domain of f.

Applications

Application II: Let $f = -\ln\det X$, with $\textsf{dom}\, f := \{X \in \mathbb{S}^n : X \succ 0\}$. Show f is convex.

Applications

Application II: Let $f = -\ln\det X$, with $\textsf{dom}\, f := \{X \in \mathbb{S}^n : X \succ 0\}$. Show f is convex.

- Let $H \in \mathbb{S}^n$, and $g(t) = f(X + tH) = -\ln\det(X + tH)$, so that

$$
\begin{aligned}
\textsf{dom}\, g &= \{t \in \mathbb{R} : X + tH \succ 0\}, \\
\det(X + tH) &= \det(X^{1/2}) \det(I + tX^{-1/2} H X^{-1/2}) \det(X^{1/2}) \\
&= \det(X) \det(I + t\tilde{H}), \quad \text{where } \tilde{H} = X^{-1/2} H X^{-1/2}, \\
\det(I + t\tilde{H}) &= \det(I + tUDU^\top) \\
&= \det(I + tU^\top UD) \\
&= \det(I + tD) \\
&= \textstyle\prod_{i=1}^n (1 + t\lambda_i) \quad \triangleright\ \lambda_i \equiv \text{eigenvalues of } \tilde{H}.
\end{aligned}
$$

Applications

Application II: Let $f = -\ln\det X$, with $\mathsf{dom}\, f := \{X \in \mathbb{S}^n : X \succ 0\}$. Show f is convex.

- ▷ Let $H \in \mathbb{S}^n$, and $g(t) = f(X + tH) = -\ln\det(X + tH)$, so that

$$
\begin{aligned}
\mathsf{dom}\, g &= \{t \in \mathbb{R} : X + tH \succ 0\}, \\
\det(X + tH) &= \det(X^{1/2})\det(I + tX^{-1/2}HX^{-1/2})\det(X^{1/2}) \\
&= \det(X)\det(I + t\tilde{H}), \quad \text{where } \tilde{H} = X^{-1/2}HX^{-1/2}, \\
\det(I + t\tilde{H}) &= \det(I + tUDU^\top) \\
&= \det(I + tU^\top UD) \\
&= \det(I + tD) \\
&= \textstyle\prod_{i=1}^n (1 + t\lambda_i) \quad \triangleright\, \lambda_i \equiv \text{eigenvalues of } \tilde{H}.
\end{aligned}
$$

Therefore,

$$
g(t) = \underbrace{-\ln\det X}_{\text{constant}} + \textstyle\sum_{i=1}^n \underbrace{[-\ln(1 + t\lambda_i)]}_{\text{convex in } t},
$$

Recitation 2: Separating hyperplane theorem, convex calculus, and convex relaxation

Shuvomoy Das Gupta

MIT

February 24, 2023

1. Separating hyperplane theorem and variants

2. Proving Schur's complement using convex calculus

3. Convex relaxation of nonconvex problems

Separating hyperplane theorem

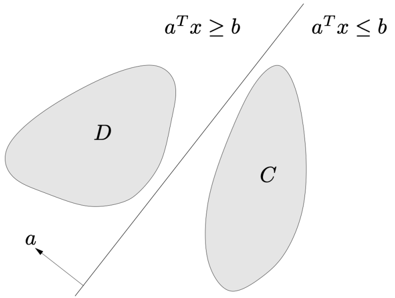
- Suppose C and D are two convex sets that do not intersect, i.e., $C \bigcap D = \emptyset$
  - $\Rightarrow$ there exist $a \neq 0$ and b such that $a^\top x \leq b$ for all $x \in C$ and $a^\top x \geq b$ for all $x \in D$.
  - i.e., the affine function $a^\top x - b$ is nonpositive on C and nonnegative on D.

Separating hyperplane theorem

- Suppose C and D are two convex sets that do not intersect, i.e., $C \bigcap D = \emptyset$
  - $\Rightarrow$ there exist $a \neq 0$ and b such that $a^\top x \leq b$ for all $x \in C$ and $a^\top x \geq b$ for all $x \in D$.
  - i.e., the affine function $a^\top x - b$ is nonpositive on C and nonnegative on D.
- The hyperplane $\{x \mid a^\top x = b\}$ is called a separating hyperplane for the sets C and D, or is said to separate the sets C and D.

Separating hyperplane theorem

- ▷ Suppose C and D are two convex sets that do not intersect, i.e., $C \bigcap D = \emptyset$
  - $\Rightarrow$ there exist $a \neq 0$ and b such that $a^\top x \leq b$ for all $x \in C$ and $a^\top x \geq b$ for all $x \in D$.
  - i.e., the affine function $a^\top x - b$ is nonpositive on C and nonnegative on D.
- ▷ The hyperplane $\{x \mid a^\top x = b\}$ is called a separating hyperplane for the sets C and D, or is said to separate the sets C and D.
- ▷ [Figure 2.19, boyd vandenberghe]



Converse separating hyperplane theorem is false

- Consider $C = D = \{0\}$, then we have exist $a = 1 \neq 0$ and $b = 0$ such that $a^\top x \leq b$ for all $x \in C$ and $a^\top x \geq b$ for all $x \in D$
- But of course $C \bigcap D = \{0\}$

Applications of separating hyperplane theorem

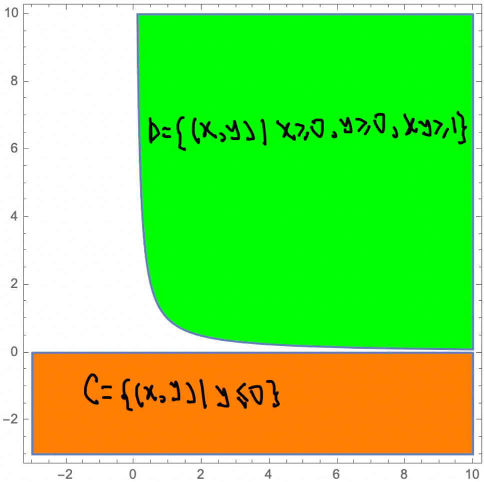
- Support vector machine
- Establishing strong duality under constraint qualification
- Collision detection
- Computing worst-case function for a given algorithm

Strict separating hyperplane theorem

- Suppose C and D are two closed, convex sets that do not intersect, i.e., $C \bigcap D = \emptyset$, and at least one of them are *bounded*
  - $\Rightarrow$ there exist $a \neq 0$ and b such that $a^\top x < b$ for all $x \in C$ and $a^\top x > b$ for all $x \in D$.
  - i.e., the affine function $a^\top x - b$ is negative on C and positive on D.

Boundedness is required for strict separation

- Consider $C = \{(x, y) \mid y \leq 0\}$, $D = \{(x, y) \mid x \geq 0, y \geq 0, xy \geq 1\}$



1. Separating hyperplane theorem and variants

2. Proving Schur's complement using convex calculus

3. Convex relaxation of nonconvex problems

Schur's complement

▷ Schur's complement: If A is invertible and $A \succ 0$ and

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0,$$

Then $C - B^T A^{-1} B \succeq 0$.

Schur's complement

- Schur's complement: If A is invertible and $A \succ 0$ and

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0,$$

  Then $C - B^T A^{-1} B \succeq 0$.

- Have many applications:
  - Used in numerical linear algebra
  - Power system harmonic analysis (Kron reduction)
  - Convex relaxation of nonconvex problems
  - Generating mathematical proofs in performance estimation literature

Schur's complement

- ▷ Schur's complement: If A is invertible and $A \succ 0$ and

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0,$$

Then $C - B^T A^{-1} B \succeq 0$.

- ▷ Have many applications:
  - Used in numerical linear algebra
  - Power system harmonic analysis (Kron reduction)
  - Convex relaxation of nonconvex problems
  - Generating mathematical proofs in performance estimation literature
- ▷ Convex calculus rule: If $f(x, y)$ is a jointly convex function in x and y and S is a convex set, then $g(y) = \inf_{x \in S} f(x, y)$ is convex.

Schur's complement

- ▷ Schur's complement: If A is invertible and $A \succ 0$ and

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0,$$

  Then $C - B^T A^{-1} B \succeq 0$.

- ▷ Have many applications:
  - Used in numerical linear algebra
  - Power system harmonic analysis (Kron reduction)
  - Convex relaxation of nonconvex problems
  - Generating mathematical proofs in performance estimation literature

- ▷ Convex calculus rule: If $f(x, y)$ is a jointly convex function in x and y and S is a convex set, then $g(y) = \inf_{x \in S} f(x, y)$ is convex.

- ▷ Prove Schur's complement using convex calculus

Schur's complement

- Define

$$f(x,y) = \begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^\top Ax + 2(By)^\top x + y^\top Cy$$

- f is jointly convex in (x, y)

Schur's complement

- Define

$$f(x,y) = \begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^\top Ax + 2(By)^\top x + y^\top Cy$$

- f is jointly convex in (x, y)
- Define $g(y) = \inf_x f(x, y)$, convex in y
  - Minimum over all x is achieved when $\nabla_x f(x, y) = 0$
  - $x^\star = -A^{-1}By$

Schur's complement

- Define

$$f(x,y) = \begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^\top A x + 2(By)^\top x + y^\top C y$$

- f is jointly convex in (x, y)
- Define $g(y) = \inf_x f(x, y)$, convex in y
  - Minimum over all x is achieved when $\nabla_x f(x, y) = 0$
  - $x^\star = -A^{-1}By$
- $$\begin{aligned} g(y) &= f(x^\star, y) \\ &= (-A^{-1}By)^\top A(-A^{-1}By) + 2(By)^\top(-A^{-1}By) + y^\top Cy \\ &= -y^\top B^\top A^{-1}By + y^\top Cy \\ &= y^T(-B^\top A^{-1}B + C)y \end{aligned}$$
- Applying the convexity result, we know that $g(y)$ is convex and hence its Hessian is positive semidefinite
- $-B^T A^{-1} B + C \succeq 0$

1. Separating hyperplane theorem and variants

2. Proving Schur's complement using convex calculus

3. Convex relaxation of nonconvex problems

Nonconvex quadratic program

- Consider

$$\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & x^\top Q x \\ \text{subject to} & x^\top A_i x \geq 0 \end{array}$$

- Under what condition this problem will be nonconvex?
- Construct a convex relaxation of this nonconvex problem

Nonconvex quadratic program

- Key idea: $x^\top Q x = \mathbf{tr}\left(x^\top Q x\right) = \mathbf{tr}\left(Q x x^\top\right)$ because $\mathbf{tr}\,(AB) = \mathbf{tr}\,(BA)$

Nonconvex quadratic program

- Key idea: $x^\top Qx = \mathbf{tr}\left(x^\top Qx\right) = \mathbf{tr}\left(Qxx^\top\right)$ because $\mathbf{tr}\,(AB) = \mathbf{tr}\,(BA)$
- Define a new variable $X = xx^\top$
- So $\left(\begin{array}{ll} \underset{x\in\mathbb{R}^d}{\text{minimize}} & x^\top Qx \\ \text{subject to} & x^\top A_i x \geq 0 \end{array}\right) = \left(\begin{array}{ll} \underset{x\in\mathbb{R}^d, X\in\mathbb{S}^d}{\text{minimize}} & \mathbf{tr}\,(QX) \\ \text{subject to} & \mathbf{tr}\,(A_i X) \geq 0, \\ & X = xx^\top \end{array}\right)$

Nonconvex quadratic program

- Key idea: $x^\top Qx = \mathbf{tr}\left(x^\top Qx\right) = \mathbf{tr}\left(Qxx^\top\right)$ because $\mathbf{tr}(AB) = \mathbf{tr}(BA)$
- Define a new variable $X = xx^\top$
- So $\left(\begin{array}{ll} \underset{x\in\mathbb{R}^d}{\text{minimize}} & x^\top Qx \\ \text{subject to} & x^\top A_i x \geq 0 \end{array}\right) = \left(\begin{array}{ll} \underset{x\in\mathbb{R}^d, X\in\mathbb{S}^d}{\text{minimize}} & \mathbf{tr}(QX) \\ \text{subject to} & \mathbf{tr}(A_i X) \geq 0, \\ & X = xx^\top \end{array}\right)$
- Consider the relaxation

$$
\begin{aligned}
& X \succeq xx^\top \\
\Leftrightarrow & X - xx^\top \succeq 0 \\
\Leftrightarrow & \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0
\end{aligned}
$$

Nonconvex quadratic program

- Key idea: $x^\top Qx = \mathbf{tr}\left(x^\top Qx\right) = \mathbf{tr}\left(Qxx^\top\right)$ because $\mathbf{tr}(AB) = \mathbf{tr}(BA)$
- Define a new variable $X = xx^\top$
- So $\left(\begin{array}{ll} \underset{x\in\mathbb{R}^d}{\text{minimize}} & x^\top Qx \\ \text{subject to} & x^\top A_i x \geq 0 \end{array}\right) = \left(\begin{array}{ll} \underset{x\in\mathbb{R}^d, X\in\mathbb{S}^d}{\text{minimize}} & \mathbf{tr}(QX) \\ \text{subject to} & \mathbf{tr}(A_i X) \geq 0, \\ & X = xx^\top \end{array}\right)$
- Consider the relaxation

$$
\begin{aligned}
& X \succeq xx^\top \\
\Leftrightarrow & X - xx^\top \succeq 0 \\
\Leftrightarrow & \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0
\end{aligned}
$$

- Drop the rank constraint, which leads to the convex relaxation
-

$$
\left(\begin{array}{ll} \underset{x\in\mathbb{R}^d, X\in\mathbb{S}^d}{\text{minimize}} & \mathbf{tr}(QX) \\ \text{subject to} & \mathbf{tr}(A_i X) \geq 0, \\ & \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0. \end{array}\right)
$$

Applications of SDP relaxation

Applications: Such relaxations works very well in

- Max-cut problem
- Optimal power flow problem (tight for tree structured network)
- Computing bounds on variables for nonconvex problem
- Various combination optimization problems
- See *Boyd, Stephen, and Lieven Vandenberghe. "Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization." Communications, Computation, Control, and Signal Processing: a tribute to Thomas Kailath (1997): 279-287*. `https://www.seas.ucla.edu/~vandenbe/publications/sdp_relaxations.pdf`
- Solving SDP is very easy in `Julia`: `https://shuvomoy.github.io/blogs/posts/Solving_semidefinite_programming_problems_in_Julia/`

Duality and KKT points

“What you seek is seeking you”-Rumi

Shuvomoy Das Gupta

MIT

Nonlinear Optimization Recitation 3

History of duality

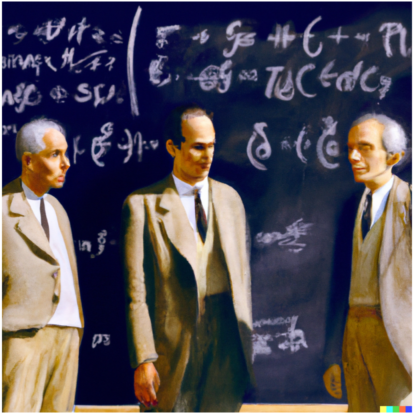


Figure 1: Werner Fenchel (left), John von Neumann (middle), and Joseph-Louis Lagrange (right): three key figures in duality (generated by DALL-E-2)

History of duality

- The first step towards duality is constructing a *Lagrangian*
- Lagrangian is named after Joseph-Louis Lagrange (1736-1813)
  - He invented Lagrangian while studying general equations of equilibrium for problems with constraints

[1] George B. Dantzig, Impact of Linear Programming on Computer Development. (**Highly Recommended**) https://apps.dtic.mil/sti/pdfs/ADA157659.pdf

History of duality

- The first step towards duality is constructing a *Lagrangian*
- Lagrangian is named after Joseph-Louis Lagrange (1736-1813)
  - He invented Lagrangian while studying general equations of equilibrium for problems with constraints
- John von Neumann (1903-1957) came up with duality theory for linear programs, it took him *one hour*[1]
  - Fall 1947: Dantzig visited Neumann to tell him about simplex, which led to linear programming duality

[1] George B. Dantzig, Impact of Linear Programming on Computer Development. (**Highly Recommended**) https://apps.dtic.mil/sti/pdfs/ADA157659.pdf

History of duality

- The first step towards duality is constructing a *Lagrangian*
- Lagrangian is named after Joseph-Louis Lagrange (1736-1813)
  - He invented Lagrangian while studying general equations of equilibrium for problems with constraints
- John von Neumann (1903-1957) came up with duality theory for linear programs, it took him *one hour*[1]
  - Fall 1947: Dantzig visited Neumann to tell him about simplex, which led to linear programming duality
- Convex optimization duality relies on the notion of Fenchel conjugate
  - Due to Werner Fenchel (1905 – 1988)

[1] George B. Dantzig, Impact of Linear Programming on Computer Development. (**Highly Recommended**) https://apps.dtic.mil/sti/pdfs/ADA157659.pdf

1. Lagrangian for a nonlinear problem

2. Weak duality and strong duality

3. Duality $\Rightarrow$ KKT conditions

Lagrangian

- Standard form problem (not necessarily convex)
- $$p^\star = \left(\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m. \end{array} \right) \tag{$\mathcal{P}$}$$
- Throughout this recitation we will assume that $p^\star$ is finite and optimal solution $x^\star$ exists

Lagrangian

- Standard form problem (not necessarily convex)
-

$$p^\star = \left(\begin{array}{lll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \le 0, & i = 1, \ldots, m. \end{array} \right) \qquad (\mathcal{P})$$

- Throughout this recitation we will assume that $p^\star$ is finite and optimal solution $x^\star$ exists
- Lagrangian

$$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\ge 0} f_i(x) \qquad (\mathcal{L})$$

- What is the nature of $L(x, \lambda)$?

Lagrangian

- Standard form problem (not necessarily convex)
- $$p^\star = \begin{pmatrix} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m. \end{pmatrix} \qquad (\mathcal{P})$$
- Throughout this recitation we will assume that $p^\star$ is finite and optimal solution $x^\star$ exists
- Lagrangian
$$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) \qquad (\mathcal{L})$$
- What is the nature of $L(x, \lambda)$?
- Interpretation: Lagrangian is a "sort of" penalized form $(\mathcal{P})$
- λ_i is the Lagrange multiplier associated with $f_i(x) \leq 0$

Lagrangian

- $$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) \qquad (\mathcal{L})$$
- λ_i acts a penalty term for per unit violation of $f_i(x) \leq 0$
- For a given x, if $f_i(x) > 0$ then $\lambda_i f_i(x)$ will introduce penalty in $L(x, \lambda)$
- If $f_i(x) \leq 0$, then $\lambda_i f_i(x)$ will introduce subsidy in $L(x, \lambda)$

Lagrangian

- $$L(x,\lambda) = f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) \qquad (\mathcal{L})$$
- λ_i acts a penalty term for per unit violation of $f_i(x) \leq 0$
- For a given x, if $f_i(x) > 0$ then $\lambda_i f_i(x)$ will introduce penalty in $L(x,\lambda)$
- If $f_i(x) \leq 0$, then $\lambda_i f_i(x)$ will introduce subsidy in $L(x,\lambda)$
- Minimizing Lagrangian is a "sort of" proxy for minimizing the original problem $(\mathcal{P})$
- Natural idea: cannot solve $(\mathcal{P})$, lets minimize the Lagrangian for a given $\lambda \geq 0$

Lagrange dual function

▷

$$
\begin{aligned}
g(\lambda) &= \min_{x \in \mathbb{R}^d} \quad L(x,\lambda) \\
&= \min_{x \in \mathbb{R}^d} f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) \\
&= -\underbrace{\left(\max_{x \in \mathbb{R}^d} - f_0(x) - \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) \right)}_{\text{convex in } \lambda}
\end{aligned}
$$

Lagrange dual function

- $$\begin{aligned} g(\lambda) &= \min_{x\in\mathbb{R}^d} \quad L(x,\lambda) \\ &= \min_{x\in\mathbb{R}^d} f_0(x) + \sum_{i=1}^m \underbrace{\lambda_i}_{\geq 0} f_i(x) \\ &= -\Big(\underbrace{\max_{x\in\mathbb{R}^d} - f_0(x) - \sum_{i=1}^m \underbrace{\lambda_i}_{\geq 0} f_i(x)}_{\text{convex in } \lambda} \Big) \end{aligned}$$
- $g(\lambda)$ is concave in λ
- If we wanted to maximize $g(\lambda)$, it is an "easy" problem
- Computing $g(\lambda)$ is easy if $(\mathcal{P})$ is convex and can by found by solving

$$\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) = 0$$

1. Lagrangian for a nonlinear problem
2. Weak duality and strong duality
3. Duality $\Rightarrow$ KKT conditions

Towards weak duality

- First nontrivial statement about duality
- If we have a feasible x for $(\mathcal{P})$ and $\lambda \geq 0$, then $g(\lambda) \leq f_0(x)$
- We have $p^\star \geq d^\star$ where

$$d^\star = \left(\begin{array}{ll} \underset{\lambda}{\text{maximize}} & g(\lambda) \\ \text{subject to} & \lambda \geq 0 \end{array} \right) \tag{$\mathcal{D}$}$$

- $(\mathcal{D})$ is called the dual problem

Towards weak duality

- First nontrivial statement about duality
- If we have a feasible x for $(\mathcal{P})$ and $\lambda \geq 0$, then $g(\lambda) \leq f_0(x)$
- We have $p^\star \geq d^\star$ where

$$d^\star = \begin{pmatrix} \underset{\lambda}{\text{maximize}} & g(\lambda) \\ \text{subject to} & \lambda \geq 0 \end{pmatrix} \tag{$\mathcal{D}$}$$

- $(\mathcal{D})$ is called the dual problem
- Proof:
  - $f_0(x) \geq f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) = L(x, \lambda) \geq \min_x L(x, \lambda) = g(\lambda)$
  - $x^\star$ is a feasible point, so $p^\star = f_0(x^\star) \geq g(\lambda) = d^\star$

Weak duality: “A thing of beauty is a joy for ever”

- We just showed, for any $\lambda \geq 0$, we have $g(\lambda) \leq p^\star$
- If we want to maximize $g(\lambda)$, it is an “easy” problem
- Natural idea: lets maximize $g(\lambda)$ to make it as close to $p^\star$ as possible

$$d^\star = \begin{pmatrix} \underset{\lambda}{\text{maximize}} & g(\lambda) \\ \text{subject to} & \lambda \geq 0 \end{pmatrix} \tag{$\mathcal{D}$}$$

- Of course, $d^\star = \max_{\lambda \geq 0} g(\lambda) \leq p^\star$: this is weak duality

Weak duality: "A thing of beauty is a joy for ever"

- We just showed, for any $\lambda \geq 0$, we have $g(\lambda) \leq p^\star$
- If we want to maximize $g(\lambda)$, it is an "easy" problem
- Natural idea: lets maximize $g(\lambda)$ to make it as close to $p^\star$ as possible

$$d^\star = \left(\begin{array}{ll} \underset{\lambda}{\text{maximize}} & g(\lambda) \\ \text{subject to} & \lambda \geq 0 \end{array} \right) \tag{$\mathcal{D}$}$$

- Of course, $d^\star = \max_{\lambda \geq 0} g(\lambda) \leq p^\star$: this is weak duality
- Weak duality always holds
- $(\mathcal{D})$ is always a convex optimization problem, no matter what the primal $(\mathcal{P})$ is
- Can be used to find nontrivial lower bounds for difficult problems

Strong duality

- If both ($\mathcal{P}$) and ($\mathcal{D}$) have the same optimal value, we say strong duality holds
- At strong duality $d^\star = p^\star$
- Does not hold in general
- Usually holds for convex problems

1. Lagrangian for a nonlinear problem

2. Weak duality and strong duality

3. Duality $\Rightarrow$ KKT conditions

How duality leads to KKT conditions

- KKT point is one of the centerpieces of modern optimization
  - Tells what an optimal point $(\mathcal{P})$ for will look like from the view point of $(\mathcal{D})$
  - It is a system of equations involving both primal and dual variables
  - Both primal and dual variables seek an equilibrium like state at optimality

How duality leads to KKT conditions

- KKT point is one of the centerpieces of modern optimization
  - Tells what an optimal point $(\mathcal{P})$ for will look like from the view point of $(\mathcal{D})$
  - It is a system of equations involving both primal and dual variables
  - Both primal and dual variables seek an equilibrium like state at optimality
- Many primal-dual solvers compute a KKT point
- Short form of Karush–Kuhn–Tucker conditions
- Harold W. Kuhn and Albert W. Tucker first published the KKT conditions in 1951
- Later it was discovered that William Karush did it in his master's thesis in 1939

KKT Conditions for any problem

- Suppose ($\mathcal{P}$) is any problem (not necessarily convex). Consider optimal primal variable $x^\star$ and optimal dual variable $\lambda^\star$ and suppose strong duality holds.

KKT Conditions for any problem

- Suppose ($\mathcal{P}$) is any problem (not necessarily convex). Consider optimal primal variable $x^\star$ and optimal dual variable $\lambda^\star$ and suppose strong duality holds.
- Then they will satisfy
  - primal feasibility: $f_i(x^\star) \leq 0$ for $i = 1, 2, \ldots, m$
  - dual feasibility: $\lambda_i^\star \geq 0$ for $i = 1, 2, \ldots, m$
  - $x^\star$ is a minimizer of the Lagrangian at $\lambda^\star$: $\nabla f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) = 0$
  - complementary slackness: $\lambda_i^\star f_i(x^\star) = 0$, for $i = 1, 2, \ldots, m$

KKT Conditions for any problem

- Suppose ($\mathcal{P}$) is any problem (not necessarily convex). Consider optimal primal variable $x^\star$ and optimal dual variable $\lambda^\star$ and suppose strong duality holds.
- Then they will satisfy
  - primal feasibility: $f_i(x^\star) \leq 0$ for $i = 1, 2, \ldots, m$
  - dual feasibility: $\lambda_i^\star \geq 0$ for $i = 1, 2, \ldots, m$
  - $x^\star$ is a minimizer of the Lagrangian at $\lambda^\star$: $\nabla f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) = 0$
  - complementary slackness: $\lambda_i^\star f_i(x^\star) = 0$, for $i = 1, 2, \ldots, m$
- These are the KKT conditions for the primal-dual system
- For any optimization problem with differentiable objective and constraints for which strong duality holds, any pair of primal and dual optimal point must satisfy KKT conditions.
- For convex primal problem, KKT condition is also sufficient for the points to be primal and dual optimal.

KKT Conditions for convex problem

- Suppose ($\mathcal{P}$) is a convex problem. Consider optimal primal variable $x^\star$ and optimal dual variable $\lambda^\star$. Then they will satisfy
  - primal feasibility: $f_i(x^\star) \leq 0$ for $i = 1, 2, \ldots, m$
  - dual feasibility: $\lambda_i^\star \geq 0$ for $i = 1, 2, \ldots, m$
  - $x^\star$ is a minimizer of the Lagrangian at $\lambda^\star$: $\nabla f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) = 0$
  - complementary slackness: $\lambda_i^\star f_i(x^\star) = 0$, for $i = 1, 2, \ldots, m$

KKT Conditions for convex problem

- Suppose ($\mathcal{P}$) is a convex problem. Consider optimal primal variable $x^\star$ and optimal dual variable $\lambda^\star$. Then they will satisfy
  - primal feasibility: $f_i(x^\star) \leq 0$ for $i = 1, 2, \ldots, m$
  - dual feasibility: $\lambda_i^\star \geq 0$ for $i = 1, 2, \ldots, m$
  - $x^\star$ is a minimizer of the Lagrangian at $\lambda^\star$: $\nabla f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) = 0$
  - complementary slackness: $\lambda_i^\star f_i(x^\star) = 0$, for $i = 1, 2, \ldots, m$
- For a KKT pair in this setup strong duality will hold automatically

KKT Conditions for convex problem

- For a KKT pair strong duality will hold automatically

$$
\begin{aligned}
&\nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) = 0 \\
\Rightarrow & d^\star = \max_{\lambda \geq 0} g(\lambda) \\
&= g(\lambda^\star) \\
&= \min_x L(x, \lambda^\star) \\
&= f_0(x^\star) + \sum_{i=1}^{m} \overbrace{\lambda_i f_i(x^\star)}^{=0 \text{ (comp. slack.)}} \\
&= f_0(x^\star) \\
&= p^\star
\end{aligned}
$$

Proof of complementary slackness

- $x^\star$ is an optimal solution to ($\mathcal{P}$), and $\lambda^\star$ is an optimal solution to ($\mathcal{D}$), and strong duality holds

Proof of complementary slackness

- $x^\star$ is an optimal solution to $(\mathcal{P})$, and $\lambda^\star$ is an optimal solution to $(\mathcal{D})$, and strong duality holds
- Then

$$
\begin{aligned}
f_0(x^\star) &= g(\lambda^\star) = \min_x \left(L(x,\lambda^\star)\right) \\
&= \min_x \left(f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x) \right) \\
&\leq f_0(x^\star) + \sum_{i=1}^m \underbrace{\lambda_i^\star}_{\geq 0} \underbrace{f_i(x^\star)}_{\leq 0} \quad \triangleright x^\star \text{ optimal for } (\mathcal{P}), \\
&\leq f_0(x^\star), \quad \triangleright \text{but this is LHS}
\end{aligned}
$$

One of the cutest proofs

- $$f_0(x^\star) + \sum_{i=1}^{m} \underbrace{\lambda_i^\star}_{\geq 0} \underbrace{f_i(x^\star)}_{\leq 0} = f_0(x^\star) \Rightarrow \underbrace{\sum_{i=1}^{m} \lambda_i^\star f_i(x^\star)}_{\leq 0} = 0$$
- If we add a bunch of nonpositive numbers and they add up to zero, then the only possibility is that each of them is individually zero!
- So, we have $\lambda_i f_i(x^\star) = 0$ for $i = 1, \ldots, m$
- Note that this also implies why $\operatorname{argmin}_x \left(f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x)\right) = x^\star$i.e., the third KKT conditions
- One of the cutest proofs that I have every seen!

Duality for convex QCQP

- Consider the convex QCQP where $P_0 \succ 0$

$$p^\star = \left(\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & \frac{1}{2} x^\top P_0 x + q_0^\top x + r_0 \\ \text{subject to} & \frac{1}{2} x^\top P_i x + q_i^\top x + r_i \leq 0, \quad i = 1, \ldots, m. \end{array} \right)$$

Duality for convex QCQP

- Consider the convex QCQP where $P_0 \succ 0$

$$p^\star = \left(\begin{array}{lll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & \frac{1}{2} x^\top P_0 x + q_0^\top x + r_0 & \\ \text{subject to} & \frac{1}{2} x^\top P_i x + q_i^\top x + r_i \leq 0, & i = 1, \ldots, m. \end{array} \right)$$

- Lagrangian

$$\begin{aligned} L(x, \lambda) &= \frac{1}{2} x^\top P_0 x + q_0^\top x + r_0 + \sum_{i=1}^m \lambda_i \left(\frac{1}{2} x^\top P_i x + q_i^\top x + r_i \right) \\ &= \frac{1}{2} x^\top \left(P_0 + \sum_{i=1}^m \lambda_i P_i \right) x + (q_0 + \sum_{i=1}^m \lambda_i q_i)^\top x + (r_0 + \sum_{i=1}^m \lambda_i r_i) \\ &= \frac{1}{2} x^\top P(\lambda) x + q(\lambda)^\top x + r(\lambda) \end{aligned}$$

Dual function

- Dual function: As $\lambda \geq 0$, $g(\lambda) = \min_x L(x,\lambda)$ achieved at
$$P(\lambda)x + q(\lambda) = 0 \Rightarrow x = -P(\lambda)^{-1}q(\lambda)$$
- So
$$\begin{aligned}
g(\lambda) = L(x,\lambda) &= \frac{1}{2}x^\top P(\lambda)x + q(\lambda)^\top x + r(\lambda) \\
&= \frac{1}{2}\left(-P(\lambda)^{-1}q(\lambda)\right)^\top P(\lambda)\left(-P(\lambda)^{-1}q(\lambda)\right) \\
&\quad + q(\lambda)^\top\left(-P(\lambda)^{-1}q(\lambda)\right) + r(\lambda) \\
&= \frac{1}{2}q(\lambda)^\top \underbrace{P(\lambda)^{-1}P(\lambda)}_{I} P(\lambda)^{-1}q(\lambda) - q(\lambda)^\top P(\lambda)^{-1}q(\lambda) + r(\lambda) \\
&= -\frac{1}{2}q(\lambda)^\top P(\lambda)^{-1}q(\lambda) + r(\lambda)
\end{aligned}$$

Dual problem

$$d^\star = \begin{pmatrix} \underset{\lambda}{\text{maximize}} & -\frac{1}{2} q(\lambda)^\top P(\lambda)^{-1} q(\lambda) + r(\lambda) \\ \text{subject to} & \lambda \geq 0 \end{pmatrix}$$

- Strong duality will hold if there is a point x that is strictly feasible
- Why is $-\frac{1}{2} q(\lambda)^\top P(\lambda)^{-1} q(\lambda) + r(\lambda)$ is concave?
- $g(\lambda) = \min_x \frac{1}{2} x^\top \left(P_0 + \sum_{i=1}^m \lambda_i P_i\right) x + (q_0 + \sum_{i=1}^m \lambda_i q_i)^\top x + (r_0 + \sum_{i=1}^m \lambda_i r_i)$
- Pointwise minimum of a family of affine functions of $\lambda \Rightarrow$ it is concave (See Section 3.2.3 of Boyd and Vandenberghe)

15.084/6.7220 Solving optimization problems in practice

Shuvomoy Das Gupta

Outline

Download the notebook

Miscellaneous topics

Solving optimization problems in practice

Please download the notebook

- From Canavas please download the zip file `Notebook.zip` in the module `Recitation 4`
- Extract the zip file to a location of your choice
- Change directory to that folder $\|Ax + b\|_2 \leq c^\top x + d$

$$f(x) = \begin{cases} \infty, & x \leq 0 \\ 0, & x > 0 \end{cases}$$

- Open Julia from terminal and type

```
cd("C:\\Desktkop'')
using IJulia
notebook()
```

Outline

Download the notebook

Miscellaneous topics

Solving optimization problems in practice

What happened?

- Recall the messed up primal problem covered in the class yesterday

$$\left(\begin{array}{ll} \underset{x\in\mathbb{R}^d}{\text{minimize}} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 - 2 \leq 0, \\ & \sqrt{2} - x \leq 0 \end{array}\right) \qquad \text{(MESS)}$$

- Convex problem :)
- KKT conditions did not hold...? What happened?
- Lets try to understand this (MESS) step by step

Build the primal-dual system

- Consider a slightly general problem with $a \geq 0$

$$\left(\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 - 2 \leq 0, \\ & a - x \leq 0 \end{array} \right) \tag{$\mathcal{P}$}$$

where in (MESS) we had $a = \sqrt{2}$

Build the primal-dual system

- Consider a slightly general problem with $a \geq 0$

$$\begin{pmatrix} \underset{x\in\mathbb{R}^d}{\text{minimize}} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 - 2 \leq 0, \\ & a - x \leq 0 \end{pmatrix} \tag{$\mathcal{P}$}$$

where in (MESS) we had $a = \sqrt{2}$

- Lagrangian
-

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1 + x_2 + \underbrace{\lambda_1}_{\geq 0}\left(x_1^2 + x_2^2 - 2\right) + \underbrace{\lambda_2}_{\geq 0}(a - x)$$

Build the primal-dual system

- Consider a slightly general problem with $a \geq 0$

$$\begin{pmatrix} \underset{x \in \mathbb{R}^d}{\text{minimize}} & x_1 + x_2 \\ \text{subject to} & x_1^2 + x_2^2 - 2 \leq 0, \\ & a - x \leq 0 \end{pmatrix} \qquad (\mathcal{P})$$

where in (MESS) we had $a = \sqrt{2}$

- Lagrangian
-

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1 + x_2 + \underbrace{\lambda_1}_{\geq 0} \left(x_1^2 + x_2^2 - 2\right) + \underbrace{\lambda_2}_{\geq 0} (a - x)$$

- Dual function $g(\lambda_1, \lambda_2) = \min_x L(x_1, x_2, \lambda_1, \lambda_2)$, can be computed by taking derivative of Lagrangian w.r.t (x_1, x_2) and set it equal to zero
  - optimal solution this problem appears at $(-1/2\lambda_1, (\lambda_2 - 1)/2\lambda_1)$

Dual problem

- So, dual function in closed form is:

$$g(\lambda_1, \lambda_2) = -\frac{8\lambda_1^2 + \lambda_2^2 - 4a\lambda_2\lambda_1 - 2\lambda_2 + 2}{4\lambda_1}$$

Dual problem

- So, dual function in closed form is:

$$g(\lambda_1, \lambda_2) = -\frac{8\lambda_1^2 + \lambda_2^2 - 4a\lambda_2\lambda_1 - 2\lambda_2 + 2}{4\lambda_1}$$

- Dual problem is

$$\left(\begin{array}{ll} \underset{\lambda_1,\lambda_2}{\text{maximize}} & -\frac{8\lambda_1^2+\lambda_2^2-4a\lambda_2\lambda_1-2\lambda_2+2}{4\lambda_1} \\ \text{subject to} & \lambda_1 \geq 0,\ \lambda_2 \geq 0. \end{array}\right)$$

Dual problem

- So, dual function in closed form is:

$$g(\lambda_1, \lambda_2) = -\frac{8\lambda_1^2 + \lambda_2^2 - 4a\lambda_2\lambda_1 - 2\lambda_2 + 2}{4\lambda_1}$$

- Dual problem is

$$\left(\begin{array}{ll} \underset{\lambda_1, \lambda_2}{\text{maximize}} & -\frac{8\lambda_1^2 + \lambda_2^2 - 4a\lambda_2\lambda_1 - 2\lambda_2 + 2}{4\lambda_1} \\ \text{subject to} & \lambda_1 \geq 0,\ \lambda_2 \geq 0. \end{array} \right)$$

- $g(\lambda_1, \lambda_2)$ concave, so
  - take derivative w.r.t (λ_1, λ_2)
  - set it equal to zero
  - if the found (λ_1, λ_2) is positive, we have the optimal solution

Lets solve the dual problem

- So $\nabla g(\lambda_1, \lambda_2) = \left(\frac{-8\lambda_1^2 + \lambda_2^2 - 2\lambda_2 + 2}{4\lambda_1^2}, \frac{2a\lambda_1 - \lambda_2 + 1}{2\lambda_1} \right) = (0, 0)$
- For the solution to exist we need $\lambda_1 \neq 0$

Lets solve the dual problem

- So $\nabla g(\lambda_1, \lambda_2) = \left(\frac{-8\lambda_1^2 + \lambda_2^2 - 2\lambda_2 + 2}{4\lambda_1^2}, \frac{2a\lambda_1 - \lambda_2 + 1}{2\lambda_1} \right) = (0, 0)$
- For the solution to exist we need $\lambda_1 \neq 0$
- Resultant equations are

$$\begin{aligned} -8\lambda_1^2 + \lambda_2^2 - 2\lambda_2 + 2 &= 0 \\ 2a\lambda_1 - \lambda_2 + 1 &= 0 \end{aligned}$$

- Solutions are

$$\left\{ \left(-\frac{1}{2\sqrt{2-a^2}}, 1 - \frac{a}{\sqrt{2-a^2}} \right), \left(\frac{1}{2\sqrt{2-a^2}}, 1 + \frac{a}{\sqrt{2-a^2}} \right) \right\}$$

Lets solve the dual problem

- So $\nabla g(\lambda_1, \lambda_2) = \left(\frac{-8\lambda_1^2 + \lambda_2^2 - 2\lambda_2 + 2}{4\lambda_1^2}, \frac{2a\lambda_1 - \lambda_2 + 1}{2\lambda_1} \right) = (0, 0)$
- For the solution to exist we need $\lambda_1 \neq 0$
- Resultant equations are

$$\begin{aligned} -8\lambda_1^2 + \lambda_2^2 - 2\lambda_2 + 2 &= 0 \\ 2a\lambda_1 - \lambda_2 + 1 &= 0 \end{aligned}$$

- Solutions are

$$\left\{ \left(-\frac{1}{2\sqrt{2-a^2}}, 1 - \frac{a}{\sqrt{2-a^2}} \right), \left(\frac{1}{2\sqrt{2-a^2}}, 1 + \frac{a}{\sqrt{2-a^2}} \right) \right\}$$

- Only the second one can be feasible as long as $2 - a^2 \geq 0$ (finite when strictly positive)
- $x_1^\star = -\sqrt{2-a^2}$ and $x_2^\star = a \Rightarrow$ on the verge of infeasibility for $a = \sqrt{2}$
- But in $(\mathcal{P})$ we set $a = \sqrt{2}$ thus no finite dual can exist, primal is very ill-posed $\Rightarrow$ **Root cause of all problem :)**

KKT conditions for $(\mathcal{P})$

- Extended arithmetic: $0 \times \infty = 0 = 0 \times (-\infty)$
  - See Rockafellar Wets Variational Inequality §E, Chapter 1

KKT conditions for $(\mathcal{P})$

- Extended arithmetic: $0 \times \infty = 0 = 0 \times (-\infty)$
  - See Rockafellar Wets Variational Inequality §E, Chapter 1
- Primal solution $x^\star = \left(-\sqrt{2-a^2}, a\right)$, dual solution $\lambda^\star = \left(1/(2\sqrt{2-a^2}), 1 + a/\sqrt{2-a^2}\right)$
- Primal feasibility: $x_1^{\star 2} + x_2^{\star 2} - 2 = 0$, $x_2^\star - a = 0$
- Dual feasibility $\lambda^\star \geq 0$ as long as $2 - a^2 \geq 0$
- Vanishing gradient of Lagrangian: $(2\lambda_1^\star x_1^\star + 1, -\lambda_2^\star + 2\lambda_1^\star x_2^\star + 1) = (0, 0)$
- Complementary slackness $\lambda_1^\star \left(x_1^{\star 2} + x_2^{\star 2} - 2\right) = 0$ and $\lambda^\star(x_2^\star - a) = 0$

Recipe for constructing duals

- Standard form problem (not necessarily convex)

$$p^\star = \begin{pmatrix} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m, \\ & h_i(x) = 0, & i = 1, \ldots, p. \end{pmatrix} \qquad (\mathcal{P})$$

Recipe for constructing duals

- Standard form problem (not necessarily convex)

$$p^\star = \left(\begin{array}{lll} \underset{x\in\mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \le 0, & i = 1, \dots, m, \\ & h_i(x) = 0, & i = 1, \dots, p. \end{array} \right) \qquad (\mathcal{P})$$

- Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\ge 0} f_i(x) + \sum_{i=1}^{p} \underbrace{\nu_i}_{\text{free}} h_i(x) \qquad (\mathcal{L})$$

- Dual function

$$g(\lambda, \nu) = \min_x L(x, \lambda, \nu)$$

Recipe for constructing duals

- Standard form problem (not necessarily convex)

$$p^\star = \left(\begin{array}{lll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m, \\ & h_i(x) = 0, & i = 1, \ldots, p. \end{array} \right) \qquad (\mathcal{P})$$

- Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} f_i(x) + \sum_{i=1}^{p} \underbrace{\nu_i}_{\text{free}} h_i(x) \qquad (\mathcal{L})$$

- Dual function

$$g(\lambda, \nu) = \min_x L(x, \lambda, \nu)$$

- Dual problem

$$d^\star = \left(\begin{array}{ll} \underset{\lambda, \nu}{\text{maximize}} & g(\lambda, \nu) \\ \text{subject to} & \lambda \geq 0, \\ & \nu : \ \text{free} \end{array} \right) \qquad (\mathcal{D})$$

A convexity proof (you can skip this)

- Show that

$$h(\lambda) = (q_0 + \sum_{i=1}^{m} \lambda_i q_i)^\top \left(P_0 + \sum_{i=1}^{m} \lambda_i P_i\right)^{-1} (q_0 + \sum_{i=1}^{m} \lambda_i q_i)$$

is convex in λ if $P_0 + \sum_{i=1}^{m} \lambda_i P_i \succ 0$.

A convexity proof (you can skip this)

- Show that

$$h(\lambda) = (q_0 + \sum_{i=1}^{m} \lambda_i q_i)^\top \left(P_0 + \sum_{i=1}^{m} \lambda_i P_i \right)^{-1} (q_0 + \sum_{i=1}^{m} \lambda_i q_i)$$

  is convex in λ if $P_0 + \sum_{i=1}^{m} \lambda_i P_i \succ 0$.

- Proof: A function f is convex if its epigraph $\{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid x \in \mathrm{dom} f,\ f(x) \le t\}$ is a convex set

A convexity proof (you can skip this)

- Show that

$$h(\lambda) = (q_0 + \sum_{i=1}^m \lambda_i q_i)^\top \left(P_0 + \sum_{i=1}^m \lambda_i P_i\right)^{-1} (q_0 + \sum_{i=1}^m \lambda_i q_i)$$

  is convex in λ if $P_0 + \sum_{i=1}^m \lambda_i P_i \succ 0$.

- Proof: A function f is convex if its epigraph $\{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid x \in \mathrm{dom} f,\ f(x) \le t\}$ is a convex set

- Epigraph of h

$$\mathrm{epi} f = \Big\{(\lambda, t) \mid (q_0 + \sum_{i=1}^m \lambda_i q_i)^\top (P_0 + \sum_{i=1}^m \lambda_i P_i)^{-1} (q_0 + \sum_{i=1}^m \lambda_i q_i) \le t,$$
$$P_0 + \sum_{i=1}^m \lambda_i P_i \succ 0\Big\}$$

A convexity proof (you can skip this)

- Schur's complement: Let A is invertible and $A \succ 0$. Then

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0 \Leftrightarrow C - B^T A^{-1} B \succeq 0$$

A convexity proof (you can skip this)

- Schur's complement: Let A is invertible and $A \succ 0$. Then

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0 \Leftrightarrow C - B^T A^{-1} B \succeq 0$$

- Note that

$$t - (q_0 + \sum_{i=1}^m \lambda_i q_i)^\top (P_0 + \sum_{i=1}^m \lambda_i P_i)^{-1} (q_0 + \sum_{i=1}^m \lambda_i q_i) \geq 0$$
$$\stackrel{\text{Schur}}{\Rightarrow} \begin{bmatrix} P_0 + \sum_{i=1}^m \lambda_i P_i & q_0 + \sum_{i=1}^m \lambda_i q_i \\ (q_0 + \sum_{i=1}^m \lambda_i q_i)^\top & t \end{bmatrix} \succeq 0,$$

which is a linear matrix inequality in the variable (λ, t), a convex constraint.

Outline

Download the notebook

Miscellaneous topics

Solving optimization problems in practice

Lets open the ipynb file

- We will explore solving problems in practice next

15.084/6.7220 Recitation 5: Miscellaneous topics

"Let the beauty of what you love be what you do." - Rumi

Shuvomoy Das Gupta

Outline

SDP and its many variants

Vector composition

Perspective of a point

Perspective of a function

SDP

- General form SDP

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^\top x \\ \text{subject to} & F_0 + \sum_{i=1}^m x_i F_i \succeq 0 \\ & Ax = b \end{array}$$

- An inequality of the form $F_0 + \sum_{i=1}^m x_i F_i \succeq 0$ is a linear matrix inequality (LMI)

LMIs in different form

- Multiple LMI constraints can be combined to create a single one
-

$$F_0 + x_1 F_1 + \ldots + x_m F_m \succeq 0,$$
$$\tilde{F}_0 + x_1 \tilde{F}_1 + \ldots + x_m \tilde{F}_m \succeq 0$$

can be written as one LMI ($\cdot$ means block matrix with all 0s of appropriate size)

$$\begin{bmatrix} F_0 & \cdot \\ \cdot & \tilde{F}_0 \end{bmatrix} + x_1 \begin{bmatrix} F_1 & \cdot \\ \cdot & \tilde{F}_1 \end{bmatrix} + \ldots + x_m \begin{bmatrix} F_m & \cdot \\ \cdot & \tilde{F}_m \end{bmatrix} \succeq 0.$$

LMIs in different form

- Multiple LMI constraints can be combined to create a single one
-

$$F_0 + x_1 F_1 + \ldots + x_m F_m \succeq 0,$$
$$\tilde{F}_0 + x_1 \tilde{F}_1 + \ldots + x_m \tilde{F}_m \succeq 0$$

  can be written as one LMI ($\cdot$ means block matrix with all 0s of appropriate size)

$$\begin{bmatrix} F_0 & \cdot \\ \cdot & \tilde{F}_0 \end{bmatrix} + x_1 \begin{bmatrix} F_1 & \cdot \\ \cdot & \tilde{F}_1 \end{bmatrix} + \ldots + x_m \begin{bmatrix} F_m & \cdot \\ \cdot & \tilde{F}_m \end{bmatrix} \succeq 0.$$

- Consider $Ax \geq b$ where $A \in \mathbb{R}^{p \times m}$ and $x \in \mathbb{R}^m$, we can write it as the LMI

$$\begin{bmatrix} a_1^\top x - b_1 & \cdot & \cdot \\ \cdot & \vdots & \cdot \\ \cdot & \cdot & a_p^\top x - b \end{bmatrix} = \begin{bmatrix} -b_1 & \cdot & \cdot \\ \cdot & \vdots & \cdot \\ \cdot & \cdot & -b \end{bmatrix} + \sum_{j=1}^{m} x_j \begin{bmatrix} (a_1)_j & \cdot & \cdot \\ \cdot & \vdots & \cdot \\ \cdot & \cdot & (a_p)_j \end{bmatrix} \succeq 0$$

Revisit Recitation 2 and HW2

- Recall that in Recitation 2 and HW2 (if you have done it correctly) we had a constraint $\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0$ and $\mathbf{tr} X \leq \rho^2$
- Is it an LMI?
- Sometimes LMIs are imposed on entire matrix, the SDP solvers internally convert them into LMIs in standard form

Revisit Recitation 2 and HW2

- Recall that in Recitation 2 and HW2 (if you have done it correctly) we had a constraint $\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0$ and $\mathbf{tr} X \leq \rho^2$
- Is it an LMI?
- Sometimes LMIs are imposed on entire matrix, the SDP solvers internally convert them into LMIs in standard form
- For illustrations let $x = (x_1, x_2)$, and $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12} & X_{22} \end{bmatrix}$
- First step: solvers define an concatenated variable $y \triangleq (y_1 := x_1, y_2 := x_2, y_3 := X_{11}, y_4 := X_{12}, y_5 := X_{22})$
- Then

$$\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_2 \\ x_1 & X_{11} & X_{12} \\ x_2 & X_{12} & X_{22} \end{bmatrix} = \begin{bmatrix} 1 & y_1 & y_2 \\ y_1 & y_3 & y_4 \\ y_2 & y_4 & y_5 \end{bmatrix} \succeq 0$$

Revisit Recitation 2 and HW2

- First note

$$\begin{bmatrix} 1 & y_1 & y_2 \\ y_1 & y_3 & y_4 \\ y_2 & y_4 & y_5 \end{bmatrix} = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} + y_1 \begin{bmatrix} \cdot & 1 & \cdot \\ 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} + y_2 \begin{bmatrix} \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot \end{bmatrix}$$

$$+ y_3 \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} + y_4 \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 \\ \cdot & 1 & \cdot \end{bmatrix} + y_5 \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 \end{bmatrix} \succeq 0$$

- Similarly
  $\mathbf{tr} X - \rho^2 = 0 + y_1 [0] + y_2 [0] + y_3 [1] + y_4 [0] + y_5 [1] - \rho^2 \leq 0$, this is also an LMI ($\times(-1)$ will make it $\geq$ form)
- Combine them together using the LMI combination recipe

Revisit Recitation 2 and HW2

- First note

$$\begin{bmatrix} 1 & y_1 & y_2 \\ y_1 & y_3 & y_4 \\ y_2 & y_4 & y_5 \end{bmatrix} = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} + y_1 \begin{bmatrix} \cdot & 1 & \cdot \\ 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} + y_2 \begin{bmatrix} \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot \end{bmatrix}$$
$$+ y_3 \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} + y_4 \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 \\ \cdot & 1 & \cdot \end{bmatrix} + y_5 \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 \end{bmatrix} \succeq 0$$

- Similarly
  $\mathbf{tr} X - \rho^2 = 0 + y_1 [0] + y_2 [0] + y_3 [1] + y_4 [0] + y_5 [1] - \rho^2 \leq 0$, this is also an LMI ($\times(-1)$ will make it $\geq$ form)
- Combine them together using the LMI combination recipe

Thus we have an LMI!

- Modern solvers basically does this thing in a very efficient way
- Just write it in the preliminary form, but your model should not involve any norm, which is very costly
  - See "Matrix completion problem: how to reconstruct a distorted image" on `https://tinyurl.com/5f68w9s6`

Outline

SDP and its many variants

Vector composition

Perspective of a point

Perspective of a function

Vector composition rule

- consider $g : \mathbf{R}^n \to \mathbf{R}^k$ and $h : \mathbf{R}^k \to \mathbf{R}$
- $f(x) = h(g_1(x), g_2(x), \ldots, g_k(x))$ with $\mathbf{dom} f$ convex
- f is convex if
  $$\begin{cases} g_i \text{ convex for all } i,\ h \text{ convex and increasing in each argument} \\ g_i \text{ concave for all } i,\ h \text{ convex and decreasing in each argument} \end{cases}$$
- *implicit:* we are establishing convexity on $\mathbf{dom} f \Rightarrow$
  - we only have to show the conditions on $g_1, \ldots, g_k, h$ on $\mathbf{dom} f$
- example:
- $f(x) = \log \sum_{i=1}^k \exp\left(g_i(x)\right)$ is convex if all the g_is are convex
- Proof: $h(z) = \log \sum_{i=1}^k \exp\left(z_i\right)$ convex and increasing in each argument, and each g_i is convex, so $h(g_1(x), \ldots, g_k(x))$ is convex

Outline

Perspective function

- Suppose $x \in \mathbb{R}^n$
- $\texttt{persp}(x) = (x_1, x_2, \ldots, x_{n-1})/x_n$, $\mathbf{dom}(\texttt{persp}) = \mathbb{R}^{n-1} \times \mathbb{R}_{++}$
- Note that $\texttt{persp} : \mathbb{R}^n \to \mathbb{R}^{n-1}$
- Consider a set $C \in \mathbb{R}^n$ such that $C \subseteq \mathbf{dom}(\texttt{persp})$
- Means that for any $x \in C$ the last component $x_n > 0$
- We want to compute $\texttt{persp}(C)$

Convexity of a set is preserved under `persp`

- If $C \subseteq \mathbf{dom}(\texttt{persp})$ convex set
  - $\Rightarrow$

$$\begin{aligned}\texttt{persp}(C) &= \{\texttt{persp}(x) \mid x \in C\} \\ &= \{(x_1, x_2, \ldots, x_{n-1})/x_n \mid (x_1, \ldots, x_{n-1}, \underbrace{x_n}_{>0}) \in C\} : \text{convex}\end{aligned}$$

Convexity of a set is preserved under $\texttt{persp}$

- If $C \subseteq \textbf{dom}(\texttt{persp})$ convex set
  - $\Rightarrow$

$$\begin{aligned}\texttt{persp}(C) &= \{\texttt{persp}(x) \mid x \in C\} \\ &= \{(x_1, x_2, \ldots, x_{n-1})/x_n \mid (x_1, \ldots, x_{n-1}, \underbrace{x_n}_{>0}) \in C\} : \text{convex}\end{aligned}$$

- If D is a convex set in $\mathbb{R}^n$, what does $\texttt{persp}^{-1}(D)$ do?
- Formally $\texttt{persp}^{-1}(D) = \{(x, t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$

Simple example

- $C = \{(x_1, x_2, x_3) \mid (x_1 - 3)^2 + (x_2 - 3)^2 + (x_3 - 3)^2 \leq 1\}$
- $\mathtt{persp}(C) = \{(x_1, x_2)/x_3 \mid (x_1 - 3)^2 + (x_2 - 3)^2 + (x_3 - 3)^2 \leq 1\}$

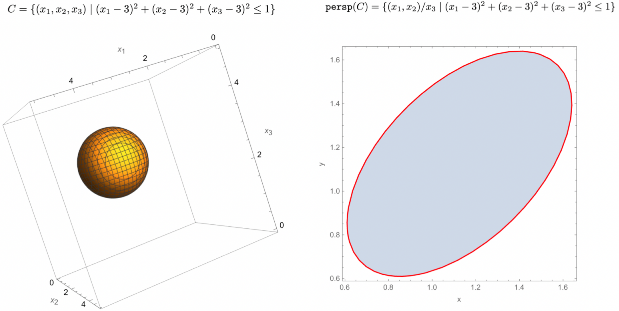


Figure: Simple perspective transformation

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- D is a convex set in $\mathbb{R}^n$, show that $\texttt{persp}^{-1}(D)$ is convex in $\mathbb{R}^{n+1}$
- Goal: for any $\theta \in [0, 1]$, and any $u, v \in \texttt{persp}^{-1}(D)$ want to show $\theta u + (1 - \theta)v \in \texttt{persp}^{-1}(D)$
- By definition $\texttt{persp}^{-1}(D) = \{(x, t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- D is a convex set in $\mathbb{R}^n$, show that $\texttt{persp}^{-1}(D)$ is convex in $\mathbb{R}^{n+1}$
- Goal: for any $\theta \in [0, 1]$, and any $u, v \in \texttt{persp}^{-1}(D)$ want to show $\theta u + (1 - \theta)v \in \texttt{persp}^{-1}(D)$
- By definition $\texttt{persp}^{-1}(D) = \{(x, t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$
- Pick $u, v \in \texttt{persp}^{-1}(D)$, then
  - by definition $(u_1, \ldots, u_n)/u_{n+1} \in D$ with $u_{n+1} > 0$ and $(v_1, \ldots, v_n)/v_{n+1} \in D$ and $v_{n+1} > 0$
  - for convenience use notation $u_{1:n} = (u_1, \ldots, u_n) \in \mathbb{R}^n$ and $v_{1:n} = (v_1, \ldots, v_n) \in \mathbb{R}^n$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- D is a convex set in $\mathbb{R}^n$, show that $\texttt{persp}^{-1}(D)$ is convex in $\mathbb{R}^{n+1}$
- Goal: for any $\theta \in [0,1]$, and any $u, v \in \texttt{persp}^{-1}(D)$ want to show $\theta u + (1-\theta)v \in \texttt{persp}^{-1}(D)$
- By definition $\texttt{persp}^{-1}(D) = \{(x,t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$
- Pick $u, v \in \texttt{persp}^{-1}(D)$, then
  - by definition $(u_1, \ldots, u_n)/u_{n+1} \in D$ with $u_{n+1} > 0$ and $(v_1, \ldots, v_n)/v_{n+1} \in D$ and $v_{n+1} > 0$
  - for convenience use notation $u_{1:n} = (u_1, \ldots, u_n) \in \mathbb{R}^n$ and $v_{1:n} = (v_1, \ldots, v_n) \in \mathbb{R}^n$
- Goal: want to show

$$
\begin{aligned}
&\theta u + (1-\theta)v \\
=&\theta(u_{1:n}, u_{n+1}) + (1-\theta)(v_{1:n}, v_{n+1}) \\
=&(\theta u_{1:n}, \theta u_{n+1}) + ((1-\theta)v_{1:n}, (1-\theta)v_{n+1})
\end{aligned}
$$

  is in $\texttt{persp}^{-1}(D)$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- Goal: $(\theta u_{1:n} + (1-\theta)v_{1:n}, \theta u_{n+1} + (1-\theta)v_{n+1})$ is in $\texttt{persp}^{-1}(D)$
- $\texttt{persp}^{-1}(D) = \{(x,t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- Goal: $(\theta u_{1:n} + (1-\theta)v_{1:n}, \theta u_{n+1} + (1-\theta)v_{n+1})$ is in $\texttt{persp}^{-1}(D)$
- $\texttt{persp}^{-1}(D) = \{(x,t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$
- Equivalent to showing
- (1)

$$\frac{1}{\theta u_{n+1} + (1-\theta)v_{n+1}} (\theta u_{1:n} + (1-\theta)v_{1:n})$$
$$= \left[\frac{\theta}{\theta u_{n+1} + (1-\theta)v_{n+1}}\right] u_{1:n} + \left[\frac{(1-\theta)}{\theta u_{n+1} + (1-\theta)v_{n+1}}\right] v_{1:n} \in D$$

and

- (2) $\theta u_{n+1} + (1-\theta)v_{n+1} > 0$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- Goal: $(\theta u_{1:n} + (1-\theta)v_{1:n}, \theta u_{n+1} + (1-\theta)v_{n+1})$ is in $\texttt{persp}^{-1}(D)$
- $\texttt{persp}^{-1}(D) = \{(x,t) \in \mathbb{R}^{n+1} \mid (x/t) \in D, t > 0\}$
- Equivalent to showing
- (1)

$$\frac{1}{\theta u_{n+1} + (1-\theta)v_{n+1}} \left(\theta u_{1:n} + (1-\theta)v_{1:n}\right)$$
$$= \left[\frac{\theta}{\theta u_{n+1} + (1-\theta)v_{n+1}}\right] u_{1:n} + \left[\frac{(1-\theta)}{\theta u_{n+1} + (1-\theta)v_{n+1}}\right] v_{1:n} \in D$$

and

- (2) $\theta u_{n+1} + (1-\theta)v_{n+1} > 0$
- (2) is obviously true as $u_{n+1} > 0$ and $v_{n+1} > 0$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- We know that D is convex and $u_{1:n}/u_{n+1} \in D$ with $u_{n+1} > 0$ and $v_{1:n}/v_{n+1} \in D$ and $v_{n+1} > 0$
- So for any $\alpha \in [0, 1]$, we have $\alpha \frac{1}{u_{n+1}} u_{1:n} + (1 - \alpha) \frac{1}{v_{n+1}} v_{1:n} \in D$

$\texttt{persp}^{-1}(D)$ is convex if D is convex (you can skip this)

- We know that D is convex and $u_{1:n}/u_{n+1} \in D$ with $u_{n+1} > 0$ and $v_{1:n}/v_{n+1} \in D$ and $v_{n+1} > 0$
- So for any $\alpha \in [0, 1]$, we have $\alpha \frac{1}{u_{n+1}} u_{1:n} + (1 - \alpha) \frac{1}{v_{n+1}} v_{1:n} \in D$
- Want to ensure $\tilde{\alpha} \frac{1}{u_{n+1}} = \frac{\theta}{\theta u_{n+1} + (1-\theta) v_{n+1}}$ and $(1 - \tilde{\alpha}) \frac{1}{v_{n+1}} = \frac{(1-\theta)}{\theta u_{n+1} + (1-\theta) v_{n+1}}$ has a solution in $\tilde{\alpha}$ with $\tilde{\alpha} \in [0, 1]$
- Thankfully $\tilde{\alpha} = \frac{\theta u_{n+1}}{\theta u_{n+1} + (1-\theta) v_{n+1}}$ is the only solution and it is clearly in $[0, 1]$

Outline

SDP and its many variants

Vector composition

Perspective of a point

Perspective of a function

Perspective of a function f

- Notation $y_{1:n} = (y_1, y_2, \ldots, y_n)$
- $f : \mathbb{R}^n \to \mathbb{R}$, then perspective of f is a function $\mathtt{persp}_f : \mathbb{R}^{n+1} \to \mathbb{R}$ defined by

$$\mathtt{persp}_f(\underbrace{y}_{\in \mathbb{R}^{n+1}}) = y_{n+1} \times f\left(\frac{1}{y_{n+1}} y_{1:n}\right)$$

  with domain
  $\mathbf{dom}(\mathtt{persp}_f) = \{y \in \mathbb{R}^{n+1} \mid \frac{1}{y_{n+1}} y_{1:n} \in \mathbf{dom} f,\ y_{n+1} > 0\}$
- If f is a convex function on $\mathbf{dom} f$ then $\mathtt{persp}_f$ is convex (on $\mathbf{dom}(\mathtt{persp}_f)$)
- Recall that a function is convex if and only if its epigraph is a convex set

$\texttt{persp}_f$ is convex if f is convex

- $\mathbf{epi}(\texttt{persp}_f) = \{(y, s) \mid \texttt{persp}_f(y) \leq s\}$, want to show that it is a convex set
- f is a convex function, so $\mathbf{epi} f = \{(x, t) \mid f(x) \leq t\}$ is convex set
- recall: $\texttt{persp}^{-1}(D) = \{(x, r) \mid (x/r) \in D,\, r > 0\}$

$\mathtt{persp}_f$ is convex if f is convex

- $\mathbf{epi}(\mathtt{persp}_f) = \{(y,s) \mid \mathtt{persp}_f(y) \le s\}$, want to show that it is a convex set
- f is a convex function, so $\mathbf{epi} f = \{(x,t) \mid f(x) \le t\}$ is convex set
- recall: $\mathtt{persp}^{-1}(D) = \{(x,r) \mid (x/r) \in D,\ r > 0\}$
- So $\mathtt{persp}^{-1}(\mathbf{epi} f) = \{(x,t,r) \mid (x,t)/r \in \mathbf{epi} f,\ r > 0\}$
- Define $P = \begin{bmatrix} I_{n\times n} & \cdot & \cdot \\ \cdot & \cdot & 1 \\ \cdot & 1 & \cdot \end{bmatrix}$ which is an invertible permutation matrix, $P(x,t,r) = (x,r,t)$
- Will show $\mathbf{epi}(\mathtt{persp}_f) = P(\mathtt{persp}^{-1}(\mathbf{epi} f))$

$\texttt{persp}_f$ is convex if f is convex

$$
\begin{aligned}
&(y,s) \in \mathbf{epi}\left(\texttt{persp}_f\right) \\
\Leftrightarrow &\texttt{persp}_f(y) \le s \\
\Leftrightarrow &y_{n+1} f\left(\frac{1}{y_{n+1}} y_{1:n}\right) \le s,\ y_{n+1} > 0 \\
\Leftrightarrow &f\left(\frac{1}{y_{n+1}} y_{1:n}\right) \le \frac{s}{y_{n+1}},\ y_{n+1} > 0 \\
\Leftrightarrow &\left(\frac{1}{y_{n+1}} y_{1:n}, \frac{s}{y_{n+1}}\right) \in \mathbf{epi} f,\ y_{n+1} > 0 \\
\Leftrightarrow &\frac{1}{y_{n+1}}\left(y_{1:n}, s\right) \in \mathbf{epi} f,\ y_{n+1} > 0 \\
&\text{/*recall } \texttt{persp}^{-1}(D) = \{(\tilde{x},\tilde{r}) \mid \tilde{x}/\tilde{r} \in D,\ \tilde{r} > 0\}\text{*/} \\
\Leftrightarrow &\left(y_{1:n}, s, y_{n+1}\right) \in \texttt{persp}^{-1}(\mathbf{epi} f) \quad \text{/*multiply both sides by } P\text{*/} \\
\Leftrightarrow &(y_{1:n}, y_{n+1}, s) = (y,s) \in P(\texttt{persp}^{-1}(\mathbf{epi} f))
\end{aligned}
$$

$\mathtt{persp}_f$ is convex if f is convex

•

$$
\begin{aligned}
&(y,s) \in \mathbf{epi}\left(\mathtt{persp}_f\right) \\
\Leftrightarrow &(y,s) \in P\left(\mathtt{persp}^{-1}(\mathbf{epi} f)\right)
\end{aligned}
$$

- We have $\mathbf{epi}(\mathtt{persp}_f) = P(\mathtt{persp}^{-1}(\mathbf{epi} f))$
- $\mathbf{epi} f$ convex $\Rightarrow \mathtt{persp}^{-1}(\mathbf{epi} f)$ is convex and $\mathtt{persp}^{-1}$ preserves convexity of a set
- $P\left(\mathtt{persp}^{-1}(\mathbf{epi} f)\right)$ is convex as P is affine (in fact invertible)
- $\mathtt{persp}_f$ is a convex function

15.084/6.7220 Recitation 6: A Journey in First-Order Methods to Understand How GPT-4 Was Trained

"In short: time time to face it, the sparks of #AGI have been ignited."-Sebastien Bubeck

Shuvomoy Das Gupta

March 14, 2023

Take home messages

- GPT-4 was released on March 14, 2023
  - You can access it via ChatGPT
  - Is capable of doing crazy things

Take home messages

- GPT-4 was released on March 14, 2023
  - You can access it via ChatGPT
  - Is capable of doing crazy things
- Sebastien Bubeck, one of the biggest names in optimization+machine learning, gave a full-house talk on GPT-4 on March 22, Wednesday at MIT CSAIL
  - He and his research group had access to GPT-4 while it was being trained without the filters
  - His conclusion: "In short: time time to face it, the sparks of Artificial General Intelligence have been ignited."
  - He and his coauthors released their findings at `https://arxiv.org/abs/2303.12712`

Take home messages

- GPT-4 was released on March 14, 2023
  - You can access it via ChatGPT
  - Is capable of doing crazy things
- Sebastien Bubeck, one of the biggest names in optimization+machine learning, gave a full-house talk on GPT-4 on March 22, Wednesday at MIT CSAIL
  - He and his research group had access to GPT-4 while it was being trained without the filters
  - His conclusion: "In short: time time to face it, the sparks of Artificial General Intelligence have been ignited."
  - He and his coauthors released their findings at `https://arxiv.org/abs/2303.12712`
- Rumor is training of GPT-4 involved 100 trillion parameters
  - Underlying optimization problem has to be the biggest ever
- Key technical challenge: how to solve such large optimization problem?

Take home messages

- GPT-4 was released on March 14, 2023
  - You can access it via ChatGPT
  - Is capable of doing crazy things
- Sebastien Bubeck, one of the biggest names in optimization+machine learning, gave a full-house talk on GPT-4 on March 22, Wednesday at MIT CSAIL
  - He and his research group had access to GPT-4 while it was being trained without the filters
  - His conclusion: "In short: time time to face it, the sparks of Artificial General Intelligence have been ignited."
  - He and his coauthors released their findings at `https://arxiv.org/abs/2303.12712`
- Rumor is training of GPT-4 involved 100 trillion parameters
  - Underlying optimization problem has to be the biggest ever
- Key technical challenge: how to solve such large optimization problem?
- GPT-4 was trained using *some* first-order method

Outline

First-order methods

Subgradient and subdifferential

Computing subgradients using *subgradient calculus*

Looking beyond the midterm

First-order and second-order methods

- Second-order methods
  - Use second-order derivatives or their approximations
  - Focus of 70s–90s. Effective for smaller problems
  - Require fewer iterations to solve the optimization problem to high accuracy, even up to machine precision
- First-order methods
  - Can be described and analyzed with gradients and subgradients
  - Have massively accelerated the training of machine learning
  - Requires at most matrix-vector multiplication
  - No matrix factorization, thus memory is not an issue
  - Sparse matrix-vector multiplication s well studied and can scale on multi-threaded CPUs, GPUs, and distributed setting
  - First-order methods are extremely simple; 2- or 3-line description. Simpler methods are easy to try out and to parallelize

Deep-learning revolution is due to first-order methods

- GPT-3 was trained using the `Adam` algorithm
- GPT-4 was likely trained using some first-order method

Algorithm 8.7 The Adam algorithm

Require: Step size ϵ (Suggested default: 0.001)
Require: Exponential decay rates for moment estimates, ρ_1 and ρ_2 in $[0, 1)$. (Suggested defaults: 0.9 and 0.999 respectively)
Require: Small constant δ used for numerical stabilization (Suggested default: 10^{-8})
Require: Initial parameters $\boldsymbol{\theta}$
  Initialize 1st and 2nd moment variables $\boldsymbol{s} = \boldsymbol{0}$, $\boldsymbol{r} = \boldsymbol{0}$
  Initialize time step $t = 0$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
    $t \leftarrow t + 1$
    Update biased first moment estimate: $\boldsymbol{s} \leftarrow \rho_1 \boldsymbol{s} + (1 - \rho_1)\boldsymbol{g}$
    Update biased second moment estimate: $\boldsymbol{r} \leftarrow \rho_2 \boldsymbol{r} + (1 - \rho_2)\boldsymbol{g} \odot \boldsymbol{g}$

momentum term inspired by Nesterov

    Correct bias in first moment: $\hat{\boldsymbol{s}} \leftarrow \frac{\boldsymbol{s}}{1 - \rho_1^t}$
    Correct bias in second moment: $\hat{\boldsymbol{r}} \leftarrow \frac{\boldsymbol{r}}{1 - \rho_2^t}$
    Compute update: $\Delta\boldsymbol{\theta} = -\epsilon \frac{\hat{\boldsymbol{s}}}{\sqrt{\hat{\boldsymbol{r}}} + \delta}$ (operations applied element-wise)
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
  end while

Figure: The `Adam` algorithm (Source: Deep Learning by Goodfellow, Bengio, and Courville)

GPT-4 training

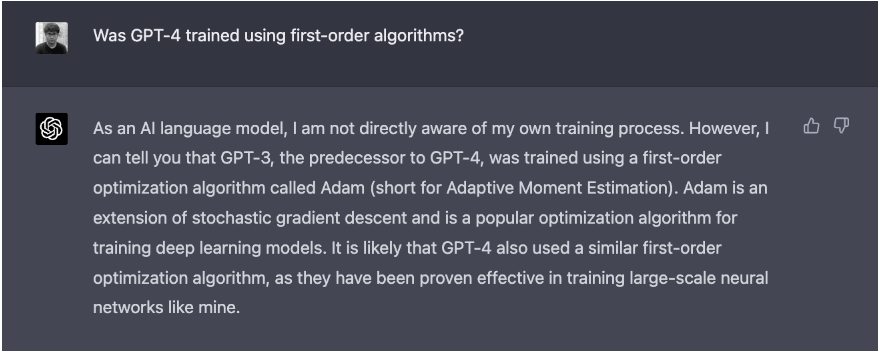


Figure: Usage of first-order methods in GPT-4 training

What is a first-order method?

- Want to solve
$$\operatorname*{minimize}_{x \in \mathbb{R}^d} \quad f(x)$$
- Algorithms that can be described and analyzed with gradients ∇f and/or *subgradients* f'

What is a first-order method?

- Want to solve
$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x)$$
- Algorithms that can be described and analyzed with gradients ∇f and/or *subgradients* f'
- (Sub)Gradient descent:
$$\underbrace{x_{i+1}}_{\text{new iterate}} = \underbrace{x_i}_{\text{prev. iterate}} - \overbrace{h_i}^{\text{stepsize}} \underbrace{f'(x_i)}_{\text{subgradient at prev. iterate}}$$

What is a first-order method?

- Want to solve
$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x)$$
- Algorithms that can be described and analyzed with gradients ∇f and/or *subgradients* f'
- (Sub)Gradient descent:
$$\underbrace{x_{i+1}}_{\text{new iterate}} = \underbrace{x_i}_{\text{prev. iterate}} - \overbrace{h_i}^{\text{stepsize}} \underbrace{f'(x_i)}_{\text{subgradient at prev. iterate}}$$
- Polyak's heavy ball method: $x_{i+1} = x_i - \alpha_i f'(x_i) + \beta_i(x_i - x_{i-1})$

What is a first-order method?

- Want to solve
$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x)$$
- Algorithms that can be described and analyzed with gradients ∇f and/or *subgradients* f'
- (Sub)Gradient descent:
$$\underbrace{x_{i+1}}_{\text{new iterate}} = \underbrace{x_i}_{\text{prev. iterate}} - \overbrace{h_i}^{\text{stepsize}} \underbrace{f'(x_i)}_{\text{subgradient at prev. iterate}}$$
- Polyak's heavy ball method: $x_{i+1} = x_i - \alpha_i f'(x_i) + \beta_i(x_i - x_{i-1})$
- Nesterov's fast gradient method:
$$x_{i+1} = y_i - \frac{1}{L}\nabla f(y_i),$$
$$y_{i+1} = x_{i+1} + \frac{i-1}{i+2}(x_{i+1} - x_i)$$

Generic description of first-order methods

- Roughly speaking, *all practical* first-order methods can be written in the following form:

$$
\begin{aligned}
&\text{pick initial point } x_0\\
&x_1 = x_0 - h_{1,0} f'(x_0)\\
&x_2 = x_1 - h_{2,0} f'(x_0) - h_{2,1} f'(x_1)\\
&x_3 = x_2 - h_{3,0} f'(x_0) - h_{3,1} f'(x_1) - h_{3,2} f'(x_2)\\
&\vdots\\
&x_N = x_{N-1} - \sum_{i=0}^{N-1} h_{N,i} f'(x_i)\\
&\text{return } x_N.
\end{aligned}
\tag{GFOM}
$$

  for some stepsizes or learning rates $\{h_{i,j}\}$
- (Sub-)Gradient descent, Nesterov's accelerated method, Polyak's heavy ball method all lie in (GFOM)

Estimation of function parameters

- Someone asked in the class these algorithms need to know function parameters L, μ and so on
- Are these algorithms just for theoretical analysis and completely useless in practice?

Estimation of function parameters

- Someone asked in the class these algorithms need to know function parameters L, μ and so on
- Are these algorithms just for theoretical analysis and completely useless in practice?
- Not at all!
- *Roughly speaking:*
- L can be computed using line-search technique, cost is $N + \log cL$ to reach the same termination tolerance
- μ can be computed using logarithmic grid search, convergence rate is the same with a change in the constant
  - constant term worsens by a factor of 4

Estimation of function parameters

- Someone asked in the class these algorithms need to know function parameters L, μ and so on
- Are these algorithms just for theoretical analysis and completely useless in practice?
- Not at all!
- *Roughly speaking:*
- L can be computed using line-search technique, cost is $N + \log cL$ to reach the same termination tolerance
- μ can be computed using logarithmic grid search, convergence rate is the same with a change in the constant
  - constant term worsens by a factor of 4
- For more details, please see *d'Aspremont, Alexandre, Damien Scieur, and Adrien Taylor. "Acceleration methods." Foundations and Trends® in Optimization 5.1-2 (2021): 1-245. Link: https://arxiv.org/abs/2101.09545*

Outline

Subgradient

- Want to solve

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x),$$

but f is not differentiable any more.

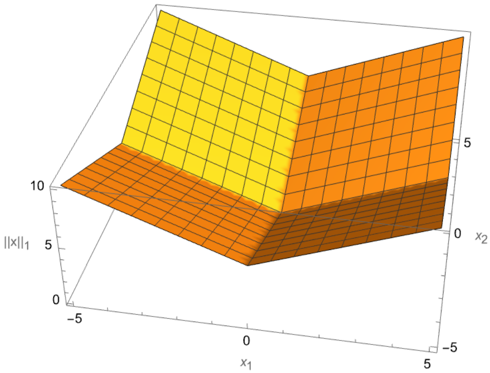


Figure: $\|x\|_1$ for $x \in \mathbb{R}^2$

Subgradient

- Want to solve
$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x),$$
but f is not differentiable any more.
- $g \in \mathbb{R}^d$ is a *subgradient* of convex f at x if
$$f(y) \geq f(x) + g^\top(y - x) \qquad \forall\, y \in \mathbb{R}^d.$$

Subgradient

- Want to solve
$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x),$$
but f is not differentiable any more.
- $g \in \mathbb{R}^d$ is a *subgradient* of convex f at x if
$$f(y) \geq f(x) + g^\top (y - x) \qquad \forall\, y \in \mathbb{R}^d.$$
- The *subdifferential* of convex f at x is
$$\partial f(x) = \{g \in \mathbb{R}^d | f(y) \geq f(x) + g^\top (y - x) \ \text{ for all } y \in \mathbb{R}^d\},$$
i.e., $\partial f(x) = \{\text{subgradients of } f \text{ at } x\}$.

Subgradient

- Want to solve
$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x),$$
but f is not differentiable any more.
- $g \in \mathbb{R}^d$ is a *subgradient* of convex f at x if
$$f(y) \geq f(x) + g^\top (y - x) \qquad \forall\, y \in \mathbb{R}^d.$$
- The *subdifferential* of convex f at x is
$$\partial f(x) = \{g \in \mathbb{R}^d | f(y) \geq f(x) + g^\top (y - x) \text{ for all } y \in \mathbb{R}^d\},$$
i.e., $\partial f(x) = \{\text{subgradients of } f \text{ at } x\}$.
- $\partial f(x)$ is a closed convex set, can be empty
- $\partial f(x) \neq \emptyset$ if $x \in \textbf{relint dom}\, f$
- Convex f is differentiable at $x \Leftrightarrow \partial f(x) = \{\nabla f(x)\}$
- $x^\star$ an optimal solution to $\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) \quad \Leftrightarrow 0 \in \partial f(x^\star)$

Some notation

For $\alpha \in \mathbb{R}$, $x \in \mathbb{R}^d$, $A, B \subseteq \mathbb{R}^d$, $M \in \mathbb{R}^{m \times d}$:

$$
\begin{aligned}
\alpha A &= \{\alpha a \,|\, a \in A\} \\
x + A &= \{x + a \,|\, a \in A\} \\
MA &= \{Ma \,|\, a \in A\} \\
A + B &= \{a + b \,|\, a \in A,\, b \in B\}
\end{aligned}
$$

Subdifferential `Hello World`

- Consider $f(x) = |x|$

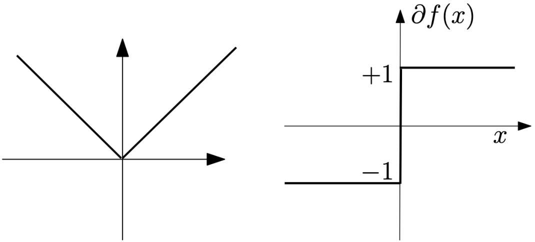


Figure: Subdifferential of $|x|$

- $$\partial|x| = \begin{cases} \{-1\}, & x < 0 \\ \{1\}, & x > 0 \\ [-1, 1], & x = 0 \end{cases} = \begin{cases} \operatorname{sign}(x), & x \neq 0 \\ [-1, 1], & x = 0 \end{cases}$$

Subdifferential `Hello World`

- For $x > 0$, $|x| = x$, so $\partial f(x) = \nabla f(x) = \{1\}$
- For $x < 0$, $|x| = -x$, so $\partial f(x) = \nabla f(x) = \{-1\}$

Subdifferential `Hello World`

- For $x > 0$, $|x| = x$, so $\partial f(x) = \nabla f(x) = \{1\}$
- For $x < 0$, $|x| = -x$, so $\partial f(x) = \nabla f(x) = \{-1\}$
- For $x = 0$, $|x|$ is not differentiable and gradient does not exist
- Want to find a subgradient at $x = 0$ such that $f(y) \geq f(0) + g \times (y - 0)$ for any $y \in \mathbb{R}$.

Subdifferential `Hello World`

- For $x > 0$, $|x| = x$, so $\partial f(x) = \nabla f(x) = \{1\}$
- For $x < 0$, $|x| = -x$, so $\partial f(x) = \nabla f(x) = \{-1\}$
- For $x = 0$, $|x|$ is not differentiable and gradient does not exist
- Want to find a subgradient at $x = 0$ such that $f(y) \geq f(0) + g \times (y - 0)$ for any $y \in \mathbb{R}$.
- Note that $f(y) = |y| = \max_{h:-1 \leq h \leq 1} h \times y$
- Check:
  1. $|-3| = \max_{h:-1 \leq h \leq 1} h \times (-3) = 3$ where $h^\star = -1$
  2. $|5| = \max_{h:-1 \leq h \leq 1} h \times (5) = 5$ where $h^\star = 1$

Subdifferential `Hello World`

- For $x > 0$, $|x| = x$, so $\partial f(x) = \nabla f(x) = \{1\}$
- For $x < 0$, $|x| = -x$, so $\partial f(x) = \nabla f(x) = \{-1\}$
- For $x = 0$, $|x|$ is not differentiable and gradient does not exist
- Want to find a subgradient at $x = 0$ such that $f(y) \geq f(0) + g \times (y - 0)$ for any $y \in \mathbb{R}$.
- Note that $f(y) = |y| = \max_{h:-1 \leq h \leq 1} h \times y$
- Check:
  1. $|-3| = \max_{h:-1 \leq h \leq 1} h \times (-3) = 3$ where $h^\star = -1$
  2. $|5| = \max_{h:-1 \leq h \leq 1} h \times (5) = 5$ where $h^\star = 1$
- So

$$f(y) = |y| = \max_{h:-1 \leq h \leq 1} h \times y \geq \tilde{g}y, \text{ for } -1 \leq \tilde{g} \leq 1,$$

$$\Rightarrow f(y) \geq f(0) + \tilde{g}(y - 0), \text{ where } -1 \leq \tilde{g} \leq 1,$$

- Hence, any $\tilde{g} \in [-1, 1]$ will be a subgradient of f at $x = 0$

Subdifferential of $\|x\|_1$

- Consider $f(x) = \|x\|_1$ which is not differentiable either

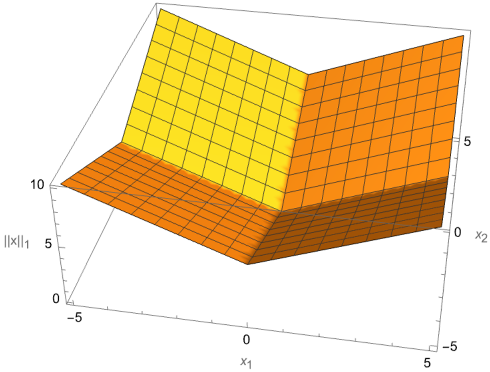


Figure: $\|x\|_1$ for $x \in \mathbb{R}^2$

- Clearly it looks more complicated, how to compute its subgradient?

Computing subdifferential of $\|x\|_1$ at $x = 0$

- We have $f(y) = \|y\|_1 = \sum_{i=1} |y_i|$
- Goal: want to find a subgradient at x such that $f(y) \geq f(0) + g^\top (y - 0)$ for any $y \in \mathbb{R}^d$

$$f(y) = \sum_{i=1}^{d} \underbrace{|y_i|}_{\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i}$$

Computing subdifferential of $\|x\|_1$ at $x = 0$

- We have $f(y) = \|y\|_1 = \sum_{i=1} |y_i|$
- Goal: want to find a subgradient at x such that $f(y) \geq f(0) + g^\top (y - 0)$ for any $y \in \mathbb{R}^d$

$$f(y) = \sum_{i=1}^{d} \underbrace{|y_i|}_{\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i}$$

$$= \sum_{i=1}^{d} \left(\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i \right)$$

Computing subdifferential of $\|x\|_1$ at $x = 0$

- We have $f(y) = \|y\|_1 = \sum_{i=1} |y_i|$
- Goal: want to find a subgradient at x such that $f(y) \geq f(0) + g^\top (y - 0)$ for any $y \in \mathbb{R}^d$

$$
\begin{aligned}
f(y) &= \sum_{i=1}^{d} \underbrace{|y_i|}_{\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i} \\
&= \sum_{i=1}^{d} \left(\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i \right) \\
&= \underbrace{\max_{h_1 : |h_i| \leq 1} h_1 y_1}_{\geq \tilde{g}_1 y_1 \text{ for } |\tilde{g}_d| \leq 1} + \ldots + \underbrace{\max_{h_d : |h_d| \leq 1} h_d y_d}_{\geq \tilde{g}_d y_d \text{ for } |\tilde{g}_d| \leq 1}
\end{aligned}
$$

Computing subdifferential of $\|x\|_1$ at $x = 0$

- We have $f(y) = \|y\|_1 = \sum_{i=1} |y_i|$
- Goal: want to find a subgradient at x such that $f(y) \geq f(0) + g^\top (y - 0)$ for any $y \in \mathbb{R}^d$

$$
\begin{aligned}
f(y) &= \sum_{i=1}^{d} \underbrace{|y_i|}_{\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i} \\
&= \sum_{i=1}^{d} \left(\max_{h_i : -1 \leq h_i \leq 1} h_i \times y_i \right) \\
&= \underbrace{\max_{h_1 : |h_i| \leq 1} h_1 y_1}_{\geq \tilde{g}_1 y_1 \text{ for } |\tilde{g}_d| \leq 1} + \ldots + \underbrace{\max_{h_d : |h_d| \leq 1} h_d y_d}_{\geq \tilde{g}_d y_d \text{ for } |\tilde{g}_d| \leq 1} \\
&\geq \tilde{g}_1 y_1 + \ldots + \tilde{g}_d y_d, \text{ where } |\tilde{g}_i| \leq 1 \text{ for all } i \in \{1, \ldots, d\}
\end{aligned}
$$

Computing subdifferential of $\|x\|_1$ at $x = 0$

- We have $f(y) = \|y\|_1 = \sum_{i=1} |y_i|$
- Goal: want to find a subgradient at x such that $f(y) \geq f(0) + g^\top (y - 0)$ for any $y \in \mathbb{R}^d$

$$
\begin{aligned}
f(y) &= \sum_{i=1}^{d} \underbrace{|y_i|}_{\max_{h_i:-1\leq h_i\leq 1} h_i \times y_i} \\
&= \sum_{i=1}^{d} \left(\max_{h_i:-1\leq h_i\leq 1} h_i \times y_i \right) \\
&= \underbrace{\max_{h_1:|h_i|\leq 1} h_1 y_1}_{\geq \tilde{g}_1 y_1 \text{ for } |\tilde{g}_d|\leq 1} + \ldots + \underbrace{\max_{h_d:|h_d|\leq 1} h_d y_d}_{\geq \tilde{g}_d y_d \text{ for } |\tilde{g}_d|\leq 1} \\
&\geq \tilde{g}_1 y_1 + \ldots + \tilde{g}_d y_d, \text{ where } |\tilde{g}_i| \leq 1 \text{ for all } i \in \{1, \ldots, d\} \\
&= \tilde{g}^\top y, \text{ where } \tilde{g} = (\tilde{g}_1, \ldots, \tilde{g}_d) \text{ with } \|\tilde{g}\|_\infty \leq 1
\end{aligned}
$$

Computing subdifferential of $\|x\|_1$ at $x = 0$

- So, we have $f(y) \geq f(0) + \tilde{g}^\top (y - 0)$ with $\|\tilde{g}\|_\infty \leq 1$
- So any $\tilde{g}$ with $\|\tilde{g}\|_\infty \leq 1$ will be a subgradient of $\|x\|_1$ at $x = 0$

Computing subdifferential of $\|x\|_1$ at $x = 0$

- So, we have $f(y) \geq f(0) + \tilde{g}^\top (y - 0)$ with $\|\tilde{g}\|_\infty \leq 1$
- So any $\tilde{g}$ with $\|\tilde{g}\|_\infty \leq 1$ will be a subgradient of $\|x\|_1$ at $x = 0$
- What about subgradient at any point x?
- For that we are going to use *subgradient calculus* rules

Outline

Subgradient calculus

- Basic rules to compute subgradient or subdifferential
- We will take a look at three of them, there are many more
- Some references that you can take a look into for more details
  - *Chapter 2 of Minimization methods for non-differentiable functions by N Z Shor*
  - *Chapter 2 of Optimization and Nonsmooth Analysis by F H Clarke*

Affine composition rule

- Affine composition rule: Consider some convex function $h : \mathbb{R}^d \to \mathbb{R}$and define $f(x) = h(Ax + b)$. Then

$$\partial \left(f(x)\right) = A^\top \times \left[\partial h(z)\right]_{z=Ax+b}$$

  as long as $Ax + b \in \mathbf{dom}\, h$
- Example Consider $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

Affine composition rule

- Affine composition rule: Consider some convex function $h : \mathbb{R}^d \to \mathbb{R}$and define $f(x) = h(Ax + b)$. Then

$$\partial \left(f(x) \right) = A^\top \times \left[\partial h(z) \right]_{z = Ax+b}$$

  as long as $Ax + b \in \mathbf{dom}\, h$
- Example Consider $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$
- Recall

$$\partial |z| = \begin{cases} \operatorname{sign}(z), & z \neq 0 \\ [-1, 1], & z = 0 \end{cases}$$

Computing subdifferential of $|a^\top x - b|$

- Want to apply $\partial\left(f(x)\right) = A^\top \times \left[\partial h(z)\right]_{z=Ax+b}$, for $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

$$\partial f(x) = h(a^\top x - b) = \partial |a^\top x - b|$$

Computing subdifferential of $|a^\top x - b|$

- Want to apply $\partial \left(f(x) \right) = A^\top \times \left[\partial h(z) \right]_{z=Ax+b}$, for $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

$$\begin{aligned} \partial f(x) &= h(a^\top x - b) = \partial |a^\top x - b| \\ &= (a^\top)^\top \times \left[\partial h(z) \right]_{z=a^\top x - b} \end{aligned}$$

Computing subdifferential of $|a^\top x - b|$

- Want to apply $\partial\left(f(x)\right) = A^\top \times \left[\partial h(z)\right]_{z=Ax+b}$, for $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

$$\begin{aligned}
\partial f(x) &= h(a^\top x - b) = \partial|a^\top x - b| \\
&= (a^\top)^\top \times \left[\partial h(z)\right]_{z=a^\top x - b} \\
&= a \times \left[\partial |z|\right]_{z=a^\top x - b}
\end{aligned}$$

Computing subdifferential of $|a^\top x - b|$

- Want to apply $\partial\left(f(x)\right) = A^\top \times \left[\partial h(z)\right]_{z=Ax+b}$, for $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

$$
\begin{aligned}
\partial f(x) &= h(a^\top x - b) = \partial|a^\top x - b| \\
&= (a^\top)^\top \times \left[\partial h(z)\right]_{z=a^\top x-b} \\
&= a \times \left[\partial|z|\right]_{z=a^\top x-b} \\
&= a \times \left[\begin{cases} \operatorname{sign}(z), & z \neq 0 \\ [-1,1], & z = 0 \end{cases}\right]_{z=a^\top x-b}
\end{aligned}
$$

Computing subdifferential of $|a^\top x - b|$

- Want to apply $\partial\left(f(x)\right) = A^\top \times \left[\partial h(z)\right]_{z=Ax+b}$, for $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

$$
\begin{aligned}
\partial f(x) &= h(a^\top x - b) = \partial |a^\top x - b| \\
&= (a^\top)^\top \times \left[\partial h(z)\right]_{z=a^\top x - b} \\
&= a \times \left[\partial |z|\right]_{z=a^\top x - b} \\
&= a \times \left[\begin{cases} \operatorname{sign}(z), & z \neq 0 \\ [-1, 1], & z = 0 \end{cases}\right]_{z=a^\top x - b} \\
&= a \times \begin{cases} \operatorname{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ [-1, 1], & a^\top x - b = 0 \end{cases}
\end{aligned}
$$

Computing subdifferential of $|a^\top x - b|$

- Want to apply $\partial\left(f(x)\right) = A^\top \times \left[\partial h(z)\right]_{z=Ax+b}$, for $h(x) = |x|$ and $f(x) = h(a^\top x - b) = |a^\top x - b|$

$$
\begin{aligned}
\partial f(x) &= h(a^\top x - b) = \partial|a^\top x - b| \\
&= (a^\top)^\top \times \left[\partial h(z)\right]_{z=a^\top x - b} \\
&= a \times \left[\partial|z|\right]_{z=a^\top x - b} \\
&= a \times \left[\begin{cases} \operatorname{sign}(z), & z \neq 0 \\ [-1, 1], & z = 0 \end{cases}\right]_{z=a^\top x - b} \\
&= a \times \begin{cases} \operatorname{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ [-1, 1], & a^\top x - b = 0 \end{cases} \\
&= \begin{cases} a \times \operatorname{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ a \times [-1, 1], & a^\top x - b = 0 \end{cases}
\end{aligned}
$$

Computing subdifferential of $|a^\top x - b|$

- $$\partial |a^\top x - b| = \begin{cases} a \,\mathrm{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ a\,[-1, 1], & a^\top x - b = 0 \end{cases}$$

Sum rule for computing subdifferential

- Sum rule: Let $h : \mathbb{R}^d \to \mathbb{R}$ and $q : \mathbb{R}^d \to \mathbb{R}$ be convex functions and let $\alpha, \beta \geq 0$. Define

  $$f(x) = \alpha h(x) + \beta q(x).$$

  Then for any $x \in (\mathbf{relint\,dom}\, h) \bigcap (\mathbf{relint\,dom}\, q)$, we have

  $$\partial f(x) = \alpha \partial h(x) + \beta \partial q(x).$$

Sum rule for computing subdifferential

- Sum rule: Let $h : \mathbb{R}^d \to \mathbb{R}$ and $q : \mathbb{R}^d \to \mathbb{R}$ be convex functions and let $\alpha, \beta \geq 0$. Define

$$f(x) = \alpha h(x) + \beta q(x).$$

  Then for any $x \in (\textbf{relint}\,\textbf{dom}\,h) \bigcap (\textbf{relint}\,\textbf{dom}\,q)$, we have

$$\partial f(x) = \alpha \partial h(x) + \beta \partial q(x).$$

- Example: Consider $f(x) = \sum_{i=1}^{m} |a_i^\top x - b_i|$, what is the subdifferential?

Computing subdifferential of $\sum_{i=1}^m |a_i^\top x - b_i|$

- Recall we showed that

$$\partial|a^\top x - b| = \begin{cases} a\,\mathrm{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ a\,[-1,1], & a^\top x - b = 0 \end{cases}$$

Computing subdifferential of $\sum_{i=1}^m |a_i^\top x - b_i|$

- Recall we showed that

$$\partial |a^\top x - b| = \begin{cases} a \operatorname{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ a\,[-1, 1], & a^\top x - b = 0 \end{cases}$$

- So,

$$\partial \sum_{i=1}^m |a_i^\top x - b_i| = \sum_{i=1}^m \begin{cases} a_i \operatorname{sign}(a_i^\top x - b_i), & a_i^\top x - b_i \neq 0 \\ a_i\,[-1, 1], & a_i^\top x - b_i = 0 \end{cases}$$

Computing subdifferential of $\sum_{i=1}^{m} |a_i^\top x - b_i|$

- Recall we showed that

$$\partial |a^\top x - b| = \begin{cases} a \operatorname{sign}(a^\top x - b), & a^\top x - b \neq 0 \\ a\,[-1, 1], & a^\top x - b = 0 \end{cases}$$

- So,

$$\partial \sum_{i=1}^{m} |a_i^\top x - b_i| = \sum_{i=1}^{m} \begin{cases} a_i \operatorname{sign}(a_i^\top x - b_i), & a_i^\top x - b_i \neq 0 \\ a_i\,[-1, 1], & a_i^\top x - b_i = 0 \end{cases}$$

- Special case

$$\begin{aligned} \partial \|x\|_1 &= \partial \left(\sum_{i=1}^{m} |e_i^\top x - 0| \right) \\ &= \sum_{i=1}^{m} \begin{cases} e_i \operatorname{sign}(e_i^\top x) = e_i \times \operatorname{sign}(x_i), & e_i^\top x = x_i \neq 0 \\ e_i\,[-1, 1], & e_i^\top x = x_i = 0 \end{cases} \end{aligned}$$

Subdifferential of pointwise maximum

- Pointwise maximum. Suppose $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \ldots, m$. Define $f(x) = \max_{i=1,\ldots,m} f_i(x)$. Then for any $x \in \mathbf{dom}\, f$ it holds that

$$\partial f(x) = \textbf{convhull} \bigcup_{i \in \texttt{active}(x)} \partial f_i(x)$$

  where $\texttt{active}(x)$ denotes index set of the functions that attain maximum at x

Subdifferential of pointwise maximum

- Pointwise maximum. Suppose $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \ldots, m$. Define $f(x) = \max_{i=1,\ldots,m} f_i(x)$. Then for any $x \in \mathbf{dom}\, f$ it holds that

$$\partial f(x) = \textbf{convhull} \bigcup_{i \in \texttt{active}(x)} \partial f_i(x)$$

  where $\texttt{active}(x)$ denotes index set of the functions that attain maximum at x

- Better to understand this as an algorithm
  - We have a point x where we want to compute $\partial f(x)$
  - Evaluate the function f at x
  - Find out which functions $f_{i^\star}$s attain the maximum at x, i.e., $f_{i^\star}(x) = \max_{i=1,\ldots,m} f_i(x)$.
  - Construct $\texttt{active}(x) = \{i^\star \mid f_{i^\star}(x) = f(x)\}$
  - Compute the subdifferential $\partial f_i(x)$ of all the f_is such that $i \in \texttt{active}(x)$
  - Construct union of all those subdifferentials: $S = \bigcup_{i \in \texttt{active}(x)} \partial f_i(x)$
  - Construct the convex hull of S

Subdifferential of pointwise maximum

- Pointwise maximum. Suppose $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \ldots, m$. Define $f(x) = \max_{i=1,\ldots,m} f_i(x)$. Then for any $x \in \mathbf{dom}\, f$ it holds that

$$\partial f(x) = \textbf{convhull} \bigcup_{i \in \texttt{active}(x)} \partial f_i(x)$$

  where $\texttt{active}(x)$ denotes index set of the functions that attain maximum at x
- While it looks complicated, in practice, we do not need the entire subdifferential to run a subgradient-based algorithm

Subdifferential of pointwise maximum

- Pointwise maximum. Suppose $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \ldots, m$. Define $f(x) = \max_{i=1,\ldots,m} f_i(x)$. Then for any $x \in \mathbf{dom}\, f$ it holds that

$$\partial f(x) = \textbf{convhull} \bigcup_{i \in \texttt{active}(x)} \partial f_i(x)$$

  where $\texttt{active}(x)$ denotes index set of the functions that attain maximum at x
- While it looks complicated, in practice, we do not need the entire subdifferential to run a subgradient-based algorithm
- We just need one subgradient $f'(x)$, to that goal we can modify the algorithm before as follows:
  - We have a point x where we want to compute one subgradient
  - Evaluate the function f at x
  - Find out one function $f_{i^\star}$ that attains the maximum at x, i.e., $f_{i^\star}(x) = \max_{i=1,\ldots,m} f_i(x)$
  - Compute one point in the subdifferential of $f_{i^\star}(x)$, i.e, $f'(x) \in \partial f(x)$

Computing subgradient of $\|Ax - b\|_\infty$

- $f(x) = \|Ax - b\|_\infty = \max_{i=1,\dots,m} |a_i^\top x - b_i|$
- Applying the max rule we have

$$\partial f(x) = \textbf{convhull} \bigcup_{i \in \texttt{active}(x)} \partial |a_i^\top x - b_i|,$$

where

$$\partial |a_i^\top x - b_i| = \begin{cases} a_i \operatorname{sign}(a_i^\top x - b_i), & a_i^\top x - b_i \neq 0 \\ a_i\,[-1, 1], & a_i^\top x - b_i = 0 \end{cases}$$

- One subgradient is given by

$$f'(x) = \begin{cases} a_{i^\star} \operatorname{sign}(a_{i^\star}^\top x - b_{i^\star}), & a_{i^\star}^\top x - b_i \neq 0 \\ a_{i^\star} \times \text{any number in}[-1, 1], & a_{i^\star}^\top x - b_i = 0 \end{cases}$$

where $|a_{i^\star}^\top x - b_{i^\star}| = \max_{i=1,\dots,m} |a_i^\top x - b_i|$

Outline

First-order methods

Subgradient and subdifferential

Computing subgradients using *subgradient calculus*

Looking beyond the midterm

Midterm

- We will release the grade early next week
- Do not worry about the midterm grade, you will be fine
- In a graduate course, grade does not matter
- We are all graduate students
  - Our goal is apply the subject material to our research
  - Everything else is noise

A poem by GPT-4

- Famous quote by Rumi:
  Let go of sorrow,
  for whatever you have is for you.
  Whatever you have lost,
  will come back in another form.
  The wound is the crack,
  where the Light enters your heart.
- This is already very profound.
- I asked ChatGPT (GPT-4) to make it a rhyming poem in the style of Rumi himself.

ChatGPT (GPT-4) output

Can you please write the following quotes by Rumi as a rhyming poem in the style of Rumi himself?

Let go of sorrow, for
whatever you have is for you.
Whatever you have lost,
will come back in another form.
The wound is the crack,
where the Light enters your heart.

Release your sorrows, let them fly,
For all you hold is yours, thereby,
In every loss and tear that's shed,
New forms arise, no need for dread.

The dance of life, forever spinning,
Transforming hearts, new chapters beginning,
Embrace the change, let go of fear,
For love's embrace is ever near.

The wounds that cause your heart to break,
Are sacred paths for light to take,
Through every crack, in every scar,
The light shines bright, it's never far.

In this play of loss and gain,
The heart finds strength, learns to sustain,
For in the wounds, the cracks, the strife,
The light shines through, revealing life.

ChatGPT (GPT-4) output

Release your sorrows, let them fly,
For all you hold is yours, thereby,
In every loss and tear that's shed,
New forms arise, no need for dread.

The dance of life, forever spinning,
Transforming hearts, new chapters beginning,
Embrace the change, let go of fear,
For love's embrace is ever near.

The wounds that cause your heart to break,
Are sacred paths for light to take,
Through every crack, in every scar,
The light shines bright, it's never far.

In this play of loss and gain,
The heart finds strength, learns to sustain,
For in the wounds, the cracks, the strife,
The light shines through, revealing life.

ChatGPT (GPT-4) output

Release your sorrows, let them fly,
For all you hold is yours, thereby,
In every loss and tear that's shed,
New forms arise, no need for dread.

The dance of life, forever spinning,
Transforming hearts, new chapters beginning,
Embrace the change, let go of fear,
For love's embrace is ever near.

The wounds that cause your heart to break,
Are sacred paths for light to take,
Through every crack, in every scar,
The light shines bright, it's never far.

In this play of loss and gain,
The heart finds strength, learns to sustain,
For in the wounds, the cracks, the strife,
The light shines through, revealing life.

- Key takeaway: a first-order method made this poem possible!

15.084/6.7220 Recitation 7: Subgradient Methods for Constrained Optimization

"The more you know who you are and what you want, the less you let things upset you."-Lost in translation (2003)

Shuvomoy Das Gupta

Outline

Project

- Most project proposals are approved (please check the comments)
- Most project proposals are research oriented and a few review based projects
- Please work consistently on the project, the final project constitutes a significant portion of your overall grade (30%)
- Please send me an email any time if you want to chat about the project, I am very happy to help
- The page limit of 10 pages excludes appendix
- You can have an appendix which does not have any page limit

Outline

Recap of subgradient

- Want to solve

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x),$$

but f is not differentiable any more.

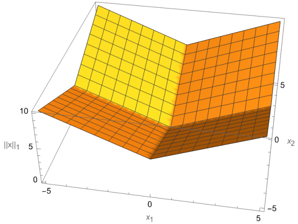


Figure: $\|x\|_1$ for $x \in \mathbb{R}^2$

Recap of subgradient

- Assume f is proper i.e., $\mathbf{dom}\, f = \{x \mid f(x) < \infty\} \neq \emptyset$
- $g \in \mathbb{R}^d$ is a *subgradient* of convex f at x if

$$f(y) \geq f(x) + g^\top (y - x) \text{ for all } y \in \mathbb{R}^d.$$

Recap of subgradient

- Assume f is proper i.e., $\mathbf{dom}\, f = \{x \mid f(x) < \infty\} \neq \emptyset$
- $g \in \mathbb{R}^d$ is a *subgradient* of convex f at x if

$$f(y) \geq f(x) + g^\top (y - x) \text{ for all } y \in \mathbb{R}^d.$$

- The *subdifferential* of convex f at x is

$$\partial f(x) = \{g \in \mathbb{R}^d | f(y) \geq f(x) + g^\top (y - x) \ \text{ for all } y \in \mathbb{R}^d\},$$

  i.e., $\partial f(x) = \{\text{subgradients of } f \text{ at } x\}$.
- Common notation: $f'(x)$ denotes one element of $\partial f(x)$

Recap of subgradient

$$\partial f(x) = \{g \in \mathbb{R}^d | f(y) \geq f(x) + g^\top (y - x) \ \text{ for all } y \in \mathbb{R}^d\},$$

- $\partial f(x)$ is a closed convex set, can be empty
- If $x \notin \mathbf{dom}\, f$ then $\partial f(x) = \emptyset$
- Convex f is differentiable at $x \Leftrightarrow \partial f(x) = \{\nabla f(x)\}$
- $x^\star$ an optimal solution to $\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) \quad \Leftrightarrow 0 \in \partial f(x^\star)$

When would a subgradient exist?

- $\partial f(x) \neq \emptyset$ if $x \in \mathbf{relint}\,\mathbf{dom}\, f$

When would a subgradient exist?

- $\partial f(x) \neq \emptyset$ if $x \in \mathbf{relint}\,\mathbf{dom}\, f$
- Recall
  - $\mathbf{aff}\, C$ is smallest affine set (i.e., translated subspace) that contains the set C
  - $\mathbf{relint}\, C = \{x \in C \mid B(x, r) \bigcap \mathbf{aff}\, C \subseteq C \text{ for some } r > 0\}$

When would a subgradient exist?

- $\partial f(x) \neq \emptyset$ if $x \in \mathbf{relint}\,\mathbf{dom}\, f$
- Recall
  - $\mathbf{aff}\, C$ is smallest affine set (i.e., translated subspace) that contains the set C
  - $\mathbf{relint}\, C = \{x \in C \mid B(x,r) \bigcap \mathbf{aff}\, C \subseteq C \text{ for some } r > 0\}$

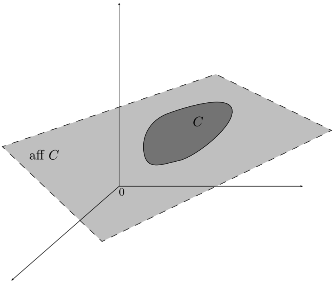


Figure: $\mathbf{relint} C$

Computing one subgradient of pointwise maximum

- Consider $f(x) = \max_{i=1,\ldots,m} f_i(x)$, given a point $x \in \mathbf{relint\,dom}\, f$ how do we compute one subgradient $f'(x)$?

Computing one subgradient of pointwise maximum

- Consider $f(x) = \max_{i=1,\ldots,m} f_i(x)$, given a point $x \in \mathbf{relint}\,\mathbf{dom}\,f$ how do we compute one subgradient $f'(x)$?
- We just need one subgradient $f'(x)$, to that goal we can modify the algorithm before as follows:
  - We have a point x where we want to compute one subgradient

Computing one subgradient of pointwise maximum

- Consider $f(x) = \max_{i=1,\ldots,m} f_i(x)$, given a point $x \in \textbf{relint}\,\textbf{dom}\, f$ how do we compute one subgradient $f'(x)$?
- We just need one subgradient $f'(x)$, to that goal we can modify the algorithm before as follows:
  - We have a point x where we want to compute one subgradient
  - Evaluate the function f at x

Computing one subgradient of pointwise maximum

- Consider $f(x) = \max_{i=1,\ldots,m} f_i(x)$, given a point $x \in \mathbf{relint\,dom}\, f$ how do we compute one subgradient $f'(x)$?
- We just need one subgradient $f'(x)$, to that goal we can modify the algorithm before as follows:
  - We have a point x where we want to compute one subgradient
  - Evaluate the function f at x
  - Find out one function $f_{i^\star}$ that attains the maximum at x, i.e., $f_{i^\star}(x) = \max_{i=1,\ldots,m} f_i(x)$

Computing one subgradient of pointwise maximum

- Consider $f(x) = \max_{i=1,\ldots,m} f_i(x)$, given a point $x \in \textbf{relint dom}\, f$ how do we compute one subgradient $f'(x)$?
- We just need one subgradient $f'(x)$, to that goal we can modify the algorithm before as follows:
  - We have a point x where we want to compute one subgradient
  - Evaluate the function f at x
  - Find out one function $f_{i^\star}$ that attains the maximum at x, i.e., $f_{i^\star}(x) = \max_{i=1,\ldots,m} f_i(x)$
  - Compute one point in the subdifferential of $f_{i^\star}(x)$, i.e, $f'(x) \in \partial f(x)$

Outline

Constrained convex optimization problem

- We want to solve

$$p^\star = \begin{pmatrix} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) \\ \text{subject to} & x \in C \end{pmatrix} \tag{$\mathcal{P}$}$$

where f_0 is a convex function and C is a closed convex set

Constrained convex optimization problem

- We want to solve

$$p^\star = \left(\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) \\ \text{subject to} & x \in C \end{array} \right) \tag{$\mathcal{P}$}$$

  where f_0 is a convex function and C is a closed convex set

- We can write the problem equivalently as

$$p^\star = \left(\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f_0(x) + \delta_C(x) \right),$$

  where $\delta_C(x)$ is equal to 0 if $x \in C$ and equal to ∞ if $x \notin C$.

- $\delta_C(x)$ is called indicator function of C and is a closed convex function if C is a closed and convex set

Indicator function of a convex set is convex

- Let C is closed and convex
- The indicator function δ_C is convex because its epigraph $\mathbf{epi}\,\delta_C = \{(x,t) \mid x \in \mathbf{dom}\,\delta_C,\, \delta_C(x) \leq t\} = \{(x,t) \mid x \in C,\, 0 \leq t\}$ is convex

Subdifferential of indicator function

- Given x we want to find $\partial\delta_C(x)$, where C is closed and convex
- If $x \notin \mathbf{dom}\,\delta_C = C$, then $\partial\delta_C(x) = \emptyset$
- Now consider $x \in \mathbf{dom}\,\delta_C = C$, then we have $\delta_C(x) = 0$. If $g \in \partial\delta_C(x)$ then it will satisfy

$$
\begin{aligned}
&\delta_C(y) \geq \underbrace{\delta_C(x)}_{=0} + g^\top(y - x), \text{ for all } y \in C \\
\Leftrightarrow &0 \geq g^\top(y - x), \text{ for all } y \in C
\end{aligned}
$$

(for $y \notin C$ it is automatically satisfied)

Subdifferential of indicator function

- Combining everything

$$\partial\delta_C(x) = \begin{cases} \{g \mid g^\top(y-x) \le 0 \text{ for all } y \in C\}, & x \in C, \\ \emptyset, & x \notin C. \end{cases}$$

- Subdifferential of indicator function is so important that it has been given a special name: it is called the *normal cone of* C

Necessary and sufficient conditions for optimality

- We want to solve

$$p^\star = \begin{pmatrix} \underset{x\in\mathbb{R}^d}{\text{minimize}} & f_0(x) \\ \text{subject to} & x \in C \end{pmatrix} = \underset{x\in\mathbb{R}^d}{\text{minimize}} \quad (f_0(x) + \delta_C(x)) \tag{$\mathcal{P}$}$$

  where f_0 is a proper $(\mathbf{dom}\, f \neq \infty)$ convex function and C is a closed convex set
- Assumption: $\mathbf{relint}\,\mathbf{dom}\, f \bigcap \mathbf{relint} C \neq \emptyset$

Necessary and sufficient conditions for optimality

- We want to solve

$$p^\star = \left(\begin{array}{ll} \underset{x \in \mathbb{R}^d}{\text{minimize}} & f_0(x) \\ \text{subject to} & x \in C \end{array} \right) = \underset{x \in \mathbb{R}^d}{\text{minimize}} \quad (f_0(x) + \delta_C(x)) \tag{$\mathcal{P}$}$$

  where f_0 is a proper ($\mathbf{dom}\, f \neq \infty$) convex function and C is a closed convex set
- Assumption: $\mathbf{relint}\,\mathbf{dom}\, f \bigcap \mathbf{relint} C \neq \emptyset$
- Then $x^\star \in C$ is an optimal solution to ($\mathcal{P}$) if and only if

$$0 \in \partial f(x^\star) + \partial \delta_C(x^\star)$$

  i.e, there is some $g \in \partial f(x^\star)$ such that $-g \in \partial \delta_C(x^\star)$
- For a proof please see Theorem 3.67 of *Beck, Amir. First-order methods in optimization. Society for Industrial and Applied Mathematics, 2017.*

Outline

Subgradient method for constrained optimization

- Consider the problem

$$p^\star = \begin{pmatrix} \text{minimize}_{x \in \mathbb{R}^d} & f_0(x) & \\ \text{subject to} & f_i(x) \le 0 & i = 1, \dots, m, \end{pmatrix}$$

where f_i is closed ($\mathbf{epi}\, f_i$ is closed set), proper, and convex for $i = 0, \dots, m$. Assume that a finite optimal solution exists.

Subgradient method for constrained optimization

- Consider the problem

$$p^\star = \begin{pmatrix} \underset{x\in\mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m, \end{pmatrix}$$

  where f_i is closed ($\mathbf{epi}\, f_i$ is closed set), proper, and convex for $i = 0, \ldots, m$. Assume that a finite optimal solution exists.

- We can write $f_i(x) \leq 0$ for $i = 1, \ldots, m$ compactly as $h(x) \triangleq \max_{i\in\{1,\ldots,m\}} f_i(x)$, which is convex

Subgradient method for constrained optimization

- Consider the problem

$$p^\star = \begin{pmatrix} \underset{x\in\mathbb{R}^d}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \le 0 & i = 1, \ldots, m, \end{pmatrix}$$

  where f_i is closed ($\mathbf{epi}\, f_i$ is closed set), proper, and convex for $i = 0, \ldots, m$. Assume that a finite optimal solution exists.

- We can write $f_i(x) \le 0$ for $i = 1, \ldots, m$ compactly as $h(x) \triangleq \max_{i\in\{1,\ldots,m\}} f_i(x)$, which is convex

- Then

$$p^\star = \begin{pmatrix} \underset{x\in\mathbb{R}^d}{\text{minimize}} & f_0(x) \\ \text{subject to} & h(x) \le 0. \end{pmatrix}$$

Subgradient method for constrained optimization

- Alternate subgradient algorithm for solving $\min_x\{f_0(x) \mid h(x) \leq 0\}$
- Initialize at some $x_0 \in \mathbb{R}^d$

Subgradient method for constrained optimization

- Alternate subgradient algorithm for solving $\min_x\{f_0(x) \mid h(x) \leq 0\}$
- Initialize at some $x_0 \in \mathbb{R}^d$
- For $k = 0, 1, 2, \ldots$ run

$$x_{k+1} = x_k - \underbrace{s_k}_{>0} g_k,$$

until optimality condition is satisfied, where

$$g_k = \begin{cases} f_0'(x_k), & \text{if } x_k \text{ feasible} \Leftrightarrow h(x_k) \leq 0 \\ h'(x_k), & \text{if } x_k \text{ infeasible} \Leftrightarrow h(x_k) > 0 \end{cases}$$

Subgradient method for constrained optimization

- Alternate subgradient algorithm for solving $\min_x\{f_0(x) \mid h(x) \leq 0\}$
- Initialize at some $x_0 \in \mathbb{R}^d$
- For $k = 0, 1, 2, \ldots$ run

$$x_{k+1} = x_k - \underbrace{s_k}_{>0} g_k,$$

until optimality condition is satisfied, where

$$g_k = \begin{cases} f_0'(x_k), & \text{if } x_k \text{ feasible} \Leftrightarrow h(x_k) \leq 0 \\ h'(x_k), & \text{if } x_k \text{ infeasible} \Leftrightarrow h(x_k) > 0 \end{cases}$$

- Intuition:
  - If the current point x_k is feasible, we use an objective subgradient $f_0'(x_k)$, as if the problem were unconstrained
  - If the current point x_k is infeasible, we choose any violated constraint, and use a subgradient of the associated constraint function
- We will investigate the convergence of this algorithm

Setup and notation

- Assumptions
  - We have a strictly feasible point x_{sf} such that $h(x_{\mathrm{sf}}) < 0$ (Slater's condition) that is suboptimal $f_0(x_f) > p^\star$

Setup and notation

- Assumptions
  - We have a strictly feasible point $x_{\rm sf}$ such that $h(x_{\rm sf}) < 0$ (Slater's condition) that is suboptimal $f_0(x_f) > p^\star$
  - There is one optimal solution $x^\star$ such that $\|x_0 - x_\star\| \leq R$ and $\|x_0 - x_{\rm sf}\| \leq R$

Setup and notation

- Assumptions
  - We have a strictly feasible point x_{sf} such that $h(x_{\mathrm{sf}}) < 0$ (Slater's condition) that is suboptimal $f_0(x_f) > p^\star$
  - There is one optimal solution $x^\star$ such that $\|x_0 - x_\star\| \leq R$ and $\|x_0 - x_{\mathrm{sf}}\| \leq R$
  - The subgradient at the iterates are bounded, i.e., there is some $G > 0$ such that for all $k = 0, 1, \ldots$ we have $\|g_k\| \leq G$

Setup and notation

- Assumptions
  - We have a strictly feasible point x_{sf} such that $h(x_{\mathrm{sf}}) < 0$ (Slater's condition) that is suboptimal $f_0(x_f) > p^\star$
  - There is one optimal solution $x^\star$ such that $\|x_0 - x_\star\| \leq R$ and $\|x_0 - x_{\mathrm{sf}}\| \leq R$
  - The subgradient at the iterates are bounded, i.e., there is some $G > 0$ such that for all $k = 0, 1, \ldots$ we have $\|g_k\| \leq G$
  - The stepsize is *square summable* but *not summable* i.e., $\lim_{k\to\infty} \sum_{i=0}^{k} s_i^2$ is finite but $\lim_{k\to\infty} \sum_{i=0}^{k} s_i = \infty$ e.g, $s_i = 1/(i+1)$

Setup and notation

- Assumptions
  - We have a strictly feasible point $x_{\rm sf}$ such that $h(x_{\rm sf}) < 0$ (Slater's condition) that is suboptimal $f_0(x_f) > p^\star$
  - There is one optimal solution $x^\star$ such that $\|x_0 - x_\star\| \leq R$ and $\|x_0 - x_{\rm sf}\| \leq R$
  - The subgradient at the iterates are bounded, i.e., there is some $G > 0$ such that for all $k = 0, 1, \ldots$ we have $\|g_k\| \leq G$
  - The stepsize is *square summable* but *not summable* i.e., $\lim_{k\to\infty} \sum_{i=0}^{k} s_i^2$ is finite but $\lim_{k\to\infty} \sum_{i=0}^{k} s_i = \infty$ e.g, $s_i = 1/(i+1)$
- We keep track of the best feasible point found so far via $f_{0,k}^\star = \min\{f_0(x_i) \mid x_i \text{ feasible for } i = 0, 1, \ldots, k\}$

Setup and notation

- Assumptions
  - We have a strictly feasible point $x_{\rm sf}$ such that $h(x_{\rm sf}) < 0$ (Slater's condition) that is suboptimal $f_0(x_f) > p^\star$
  - There is one optimal solution $x^\star$ such that $\|x_0 - x_\star\| \leq R$ and $\|x_0 - x_{\rm sf}\| \leq R$
  - The subgradient at the iterates are bounded, i.e., there is some $G > 0$ such that for all $k = 0, 1, \ldots$ we have $\|g_k\| \leq G$
  - The stepsize is *square summable* but *not summable* i.e., $\lim_{k\to\infty} \sum_{i=0}^k s_i^2$ is finite but $\lim_{k\to\infty} \sum_{i=0}^k s_i = \infty$ e.g, $s_i = 1/(i+1)$
- We keep track of the best feasible point found so far via $f_{0,k}^\star = \min\{f_0(x_i) \mid x_i \text{ feasible for } i = 0, 1, \ldots, k\}$
- **Convergence result:** $\lim_{k\to\infty} f_{0,k}^\star = p^\star$

Outline

Project

Recap of subgradient

Solving constrained optimization problems via subgradients

Alternate subgradient method

Proof of alternate subgradient method

Convergence proof

- We will do a proof by contradiction
- Assume $\lim_{k\to\infty} f^\star_{0,k} > p^\star$ (by definition $f^\star_{0,k}$ can not be strictly smaller than $p^\star$)

Convergence proof

- We will do a proof by contradiction
- Assume $\lim_{k\to\infty} f^\star_{0,k} > p^\star$ (by definition $f^\star_{0,k}$ can not be strictly smaller than $p^\star$)
- Now

$$
\begin{aligned}
&\lim_{k\to\infty} f^\star_{0,k} > p^\star \\
\Leftrightarrow & f^\star_{0,k} \geq p^\star + \epsilon \text{ for some } \epsilon > 0 \text{ for all } k \\
\Leftrightarrow & f_0(x_i) \geq p^\star + \epsilon, \text{ for some } \epsilon > 0 \text{ for all feasible } x_i \text{ with } i \in \{0, \ldots, k\} \\
& \hfill \texttt{(divergence)}
\end{aligned}
$$

Convergence proof

- We will do a proof by contradiction
- Assume $\lim_{k\to\infty} f^\star_{0,k} > p^\star$ (by definition $f^\star_{0,k}$ can not be strictly smaller than $p^\star$)
- Now

$$
\begin{aligned}
&\lim_{k\to\infty} f^\star_{0,k} > p^\star \\
\Leftrightarrow & f^\star_{0,k} \geq p^\star + \epsilon \text{ for some } \epsilon > 0 \text{ for all } k \\
\Leftrightarrow & f_0(x_i) \geq p^\star + \epsilon, \text{ for some } \epsilon > 0 \text{ for all feasible } x_i \text{ with } i \in \{0, \dots, k\} \\
& \hfill \texttt{(divergence)}
\end{aligned}
$$

- As a logical statement:
$\exists_{\epsilon>0} \, \forall_{k\in\mathbb{N}} \, \forall_{i\in\{0,1,\dots,k\}} \, \forall_{x_i:\text{feasible}} \; f_0(x_i) - p^\star \geq \epsilon$

Convergence proof

- We will do a proof by contradiction
- Assume $\lim_{k\to\infty} f^\star_{0,k} > p^\star$ (by definition $f^\star_{0,k}$ can not be strictly smaller than $p^\star$)
- Now

$$\begin{aligned}
&\lim_{k\to\infty} f^\star_{0,k} > p^\star \\
\Leftrightarrow\, & f^\star_{0,k} \geq p^\star + \epsilon \text{ for some } \epsilon > 0 \text{ for all } k \\
\Leftrightarrow\, & f_0(x_i) \geq p^\star + \epsilon, \text{ for some } \epsilon > 0 \text{ for all feasible } x_i \text{ with } i \in \{0, \ldots, k\} \\
& \hfill \texttt{(divergence)}
\end{aligned}$$

- As a logical statement:
  $\exists_{\epsilon>0}\, \forall_{k\in\mathbb{N}}\, \forall_{i\in\{0,1,\ldots,k\}}\, \forall_{x_i:\text{feasible}}\, f_0(x_i) - p^\star \geq \epsilon$
- We will show that something bad will happen we assume `(divergence)`.

Convex combination of $x^\star$ and x_{sf}

- Consider the point $\tilde{x} = (1-\theta)x^\star + \theta x_{\mathrm{sf}}$ where $\theta \in [0, 1]$

Convex combination of $x^\star$ and $x_{\rm sf}$

- Consider the point $\tilde{x} = (1-\theta)x^\star + \theta x_{\rm sf}$ where $\theta \in [0,1]$
- We have

$$\begin{aligned} f_0(\tilde{x}) &= f_0\left((1-\theta)x^\star + \theta x_{\rm sf}\right) \\ &\leq (1-\theta) f_0(x^\star) + \theta f_0(x_{\rm sf}) \\ &= \underbrace{f_0(x^\star)}_{=p^\star} + \theta \underbrace{\left(f_0(x_{\rm sf}) - f_0(x^\star)\right)}_{>0} \end{aligned}$$

Convex combination of $x^\star$ and x_{sf}

- Consider the point $\tilde{x} = (1-\theta)x^\star + \theta x_{\text{sf}}$ where $\theta \in [0,1]$
- We have

$$\begin{aligned} f_0(\tilde{x}) &= f_0\left((1-\theta)x^\star + \theta x_{\text{sf}}\right) \\ &\le (1-\theta) f_0(x^\star) + \theta f_0(x_{\text{sf}}) \\ &= \underbrace{f_0(x^\star)}_{=p^\star} + \theta \underbrace{\left(f_0(x_{\text{sf}}) - f_0(x^\star)\right)}_{>0} \end{aligned}$$

- Set $\theta := \min\left\{1, \frac{\epsilon}{2}\frac{1}{f_0(x_{\text{sf}})-p^\star}\right\}$: very cleverly chosen

$$\begin{aligned} f_0(\tilde{x}) &\le p^\star + \min\left\{1, \frac{\epsilon}{2}\frac{1}{f_0(x_{\text{sf}})-p^\star}\right\}\left(f_0(x_{\text{sf}})-p^\star\right) \\ &\le p^\star + \frac{\epsilon}{2}\cancel{\frac{1}{(f_0(x_{\text{sf}})-p^\star)}\left(f_0(x_{\text{sf}})-p^\star\right)} \\ &\le p^\star + \frac{\epsilon}{2} \end{aligned}$$

Function value of $\tilde{x}$

- So we have

$$0 \leq f_0(\tilde{x}) - p^\star \leq \frac{\epsilon}{2} \qquad \texttt{(subopt\_xtilde)}$$

- $\tilde{x}$ is $\frac{\epsilon}{2}$-suboptimal

Feasibility of $\tilde{x}$

- We have $\tilde{x} = (1-\theta)x^\star + \theta x_{\rm sf}$ where $\theta \coloneqq \min\left\{1, \frac{\epsilon}{2}\frac{1}{f_0(x_{\rm sf}) - f(x^\star)}\right\} \in (0, 1]$
- $\tilde{x}$ is a convex combination of $x^\star \in C$ and $x_{\rm sf} \in C$, so $\tilde{x} \in C$

Feasibility of $\tilde{x}$

- We have $\tilde{x} = (1-\theta)x^\star + \theta x_{\text{sf}}$ where $\theta := \min\left\{1, \frac{\epsilon}{2}\frac{1}{f_0(x_{\text{sf}}) - f(x^\star)}\right\} \in (0, 1]$
- $\tilde{x}$ is a convex combination of $x^\star \in C$ and $x_{\text{sf}} \in C$, so $\tilde{x} \in C$
- Now $h(\tilde{x}) = h\left((1-\theta)x^\star + \theta x_{\text{sf}}\right) \le (1-\theta)\underbrace{h(x^\star)}_{\le 0} + \theta\underbrace{h(x_{\text{sf}})}_{<0} < 0$

  because $\theta \in (0, 1]$

Feasibility of $\tilde{x}$

- We have $\tilde{x} = (1-\theta)x^\star + \theta x_{\text{sf}}$ where $\theta := \min\left\{1, \frac{\epsilon}{2}\frac{1}{f_0(x_{\text{sf}})-f(x^\star)}\right\} \in (0,1]$
- $\tilde{x}$ is a convex combination of $x^\star \in C$ and $x_{\text{sf}} \in C$, so $\tilde{x} \in C$
- Now $h(\tilde{x}) = h\left((1-\theta)x^\star + \theta x_{\text{sf}}\right) \leq (1-\theta)\underbrace{h(x^\star)}_{\leq 0} + \theta \underbrace{h(x_{\text{sf}})}_{<0} < 0$

  because $\theta \in (0,1]$
- So there is some $\gamma > 0$ such that

$$h(\tilde{x}) \leq -\gamma \qquad (\texttt{strict\_fsblt\_x\_tilde})$$

Feasibility of $\tilde{x}$

- We have $\tilde{x} = (1-\theta)x^\star + \theta x_{\mathrm{sf}}$ where $\theta := \min\left\{1, \frac{\epsilon}{2}\frac{1}{f_0(x_{\mathrm{sf}}) - f(x^\star)}\right\} \in (0,1]$
- $\tilde{x}$ is a convex combination of $x^\star \in C$ and $x_{\mathrm{sf}} \in C$, so $\tilde{x} \in C$
- Now $h(\tilde{x}) = h\left((1-\theta)x^\star + \theta x_{\mathrm{sf}}\right) \le (1-\theta)\underbrace{h(x^\star)}_{\le 0} + \theta\underbrace{h(x_{\mathrm{sf}})}_{<0} < 0$

  because $\theta \in (0,1]$
- So there is some $\gamma > 0$ such that

$$h(\tilde{x}) \le -\gamma \qquad (\texttt{strict\_fsblt\_x\_tilde})$$

- Next, we will show every iterate satisfies

$$\|x_{i+1} - \tilde{x}\|^2 \le \|x_i - \tilde{x}\|^2 + s_i^2\|g_i\|^2 - s_i\underbrace{\beta}_{>0}$$

Case 1: x_i feasible

- Note that
  - From (`divergence`) we have $-f_0(x_i) + p^\star \le -\epsilon$ and
  - From (`subopt_xtilde`) we have $f_0(\tilde{x}) - p^\star \le \frac{\epsilon}{2}$
  - Adding the last two together $f_0(\tilde{x}) - f_0(x_i) \le -\frac{\epsilon}{2}$

Case 1: x_i feasible

- Note that
  - From (`divergence`) we have $-f_0(x_i) + p^\star \leq -\epsilon$ and
  - From (`subopt_xtilde`) we have $f_0(\tilde{x}) - p^\star \leq \frac{\epsilon}{2}$
  - Adding the last two together $f_0(\tilde{x}) - f_0(x_i) \leq -\frac{\epsilon}{2}$
- x_i feasible means $h(x_i) \leq 0$
  - So here we pick $g_i \in \partial f_0(x_i)$, so $f_0(y) \geq f_0(x_i) + g_i^\top (y - x_i)$ for any y

Case 1: x_i feasible

- Note that
  - From (`divergence`) we have $-f_0(x_i) + p^\star \le -\epsilon$ and
  - From (`subopt_xtilde`) we have $f_0(\tilde{x}) - p^\star \le \frac{\epsilon}{2}$
  - Adding the last two together $f_0(\tilde{x}) - f_0(x_i) \le -\frac{\epsilon}{2}$
- x_i feasible means $h(x_i) \le 0$
  - So here we pick $g_i \in \partial f_0(x_i)$, so $f_0(y) \ge f_0(x_i) + g_i^\top (y - x_i)$ for any y
  - Set $y := \tilde{x}$ yields $f_0(\tilde{x}) - f_0(x_i) \ge g_i^\top (\tilde{x} - x_i)$
  - Note that

$$
\begin{aligned}
\|x_{i+1} - \tilde{x}\|^2 = \|x_i - s_i g_i - \tilde{x}\|^2 &= \|(x_i - \tilde{x}) - s_i g_i\|^2 \\
&= \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 - 2 s_i g_i^\top (x_i - \tilde{x}) \\
&= \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + 2 s_i \underbrace{g_i^\top (\tilde{x} - x_i)}_{\le f_0(\tilde{x}) - f_0(x_i)} \\
&\le \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + 2 s_i \underbrace{(f_0(\tilde{x}) - f_0(x_i))}_{\le -\frac{\epsilon}{2}} \\
&\le \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + s_i(-\epsilon)
\end{aligned}
$$

Case 2: x_i infeasible

- Observe
  - x_i infeasible means $h(x_i) > 0 \Leftrightarrow -h(x_i) < 0$
  - From (`strict_fsblt_x_tilde`), we have $h(\tilde{x}) \leq -\gamma$
  - Adding the last two we have $h(\tilde{x}) - h(x_i) \leq -\gamma$

Case 2: x_i infeasible

- Observe
  - x_i infeasible means $h(x_i) > 0 \Leftrightarrow -h(x_i) < 0$
  - From (`strict_fsblt_x_tilde`), we have $h(\tilde{x}) \leq -\gamma$
  - Adding the last two we have $h(\tilde{x}) - h(x_i) \leq -\gamma$
- Also, when x_i infeasible, we pick
  - $g_i \in \partial h(x_i)$, so $h(y) \geq h(x_i) + g_i^\top (y - x_i)$ for any y
  - Set $y := \tilde{x}$ yields $h(\tilde{x}) - h(x_i) \geq g_i^\top (\tilde{x} - x_i)$

Case 2: x_i infeasible

- Observe
  - x_i infeasible means $h(x_i) > 0 \Leftrightarrow -h(x_i) < 0$
  - From (`strict_fsblt_x_tilde`), we have $h(\tilde{x}) \leq -\gamma$
  - Adding the last two we have $h(\tilde{x}) - h(x_i) \leq -\gamma$
- Also, when x_i infeasible, we pick
  - $g_i \in \partial h(x_i)$, so $h(y) \geq h(x_i) + g_i^\top (y - x_i)$ for any y
  - Set $y := \tilde{x}$ yields $h(\tilde{x}) - h(x_i) \geq g_i^\top (\tilde{x} - x_i)$
  - Note that

$$
\begin{aligned}
\|x_{i+1} - \tilde{x}\|^2 = \|x_i - s_i g_i - \tilde{x}\|^2 &= \|(x_i - \tilde{x}) - s_i g_i\|^2 \\
&= \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 - 2 s_i g_i^\top (x_i - \tilde{x}) \\
&= \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + 2 s_i \underbrace{g_i^\top (\tilde{x} - x_i)}_{\leq h(\tilde{x}) - h(x_i)} \\
&\leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + 2 s_i \underbrace{(h(\tilde{x}) - h(x_i))}_{\leq -\gamma} \\
&\leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + s_i (-2\gamma)
\end{aligned}
$$

Combine the last two steps

- When $h(x_i) \leq 0$ we have

$$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + s_i(-\epsilon)$$

- When $h(x_i) > 0$ we have

$$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 + s_i \left(-2\gamma\right)$$

- define $-\beta = \max(-\epsilon, -2\gamma)$, clearly $\beta > 0$
- So, no matter x_i is feasible or infeasible, we have for all $i = 0, 1, \ldots$

$$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 - s_i \beta$$

A telescoping sum

- So, no matter x_i is feasible or infeasible, we have for all $i = 0, 1, \ldots$

$$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 - s_i \beta$$

A telescoping sum

- So, no matter x_i is feasible or infeasible, we have for all $i = 0, 1, \ldots$

$$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 - s_i \beta$$

- Lets do a telescoping sum ranging over $i = 0, 1, \ldots, k$

$$\begin{aligned}
\|x_1 - \tilde{x}\|^2 - \|x_0 - \tilde{x}\|^2 &\leq s_0^2 \|g_0\|^2 - s_0 \beta \\
\|x_2 - \tilde{x}\|^2 - \|x_1 - \tilde{x}\|^2 &\leq s_1^2 \|g_1\|^2 - s_1 \beta \\
&\vdots \\
\|x_k - \tilde{x}\|^2 - \|x_{k-1} - \tilde{x}\|^2 &\leq s_{k-1}^2 \|g_{k-1}\|^2 - s_{k-1} \beta \\
\|x_{k+1} - \tilde{x}\|^2 - \|x_k - \tilde{x}\|^2 &\leq s_k^2 \|g_k\|^2 - s_k \beta
\end{aligned}$$

A telescoping sum

- So, no matter x_i is feasible or infeasible, we have for all $i = 0, 1, \ldots$

$$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2 \|g_i\|^2 - s_i \beta$$

- Lets do a telescoping sum ranging over $i = 0, 1, \ldots, k$

$$\begin{aligned}
\|x_1 - \tilde{x}\|^2 - \|x_0 - \tilde{x}\|^2 &\leq s_0^2 \|g_0\|^2 - s_0 \beta \\
\|x_2 - \tilde{x}\|^2 - \|x_1 - \tilde{x}\|^2 &\leq s_1^2 \|g_1\|^2 - s_1 \beta \\
&\vdots \\
\|x_k - \tilde{x}\|^2 - \|x_{k-1} - \tilde{x}\|^2 &\leq s_{k-1}^2 \|g_{k-1}\|^2 - s_{k-1} \beta \\
\|x_{k+1} - \tilde{x}\|^2 - \|x_k - \tilde{x}\|^2 &\leq s_k^2 \|g_k\|^2 - s_k \beta
\end{aligned}$$

- Adding the inequalities above yields

$$\|x_{k+1} - \tilde{x}\|^2 - \|x_0 - \tilde{x}\|^2 \leq \sum_{i=0}^{k} s_i^2 \|g_i\|^2 - \beta \sum_{i=0}^{k} s_i$$

Arriving at the contradiction

- Adding the inequalities above yields

$$
\begin{aligned}
\|x_{k+1} - \tilde{x}\|^2 - \|x_0 - \tilde{x}\|^2 &\leq \sum_{i=0}^{k} s_i^2 \|g_i\|^2 - \beta \sum_{i=0}^{k} s_i \\
\Leftrightarrow \|x_{k+1} - \tilde{x}\|^2 &\leq \underbrace{\|x_0 - \tilde{x}\|^2}_{\leq R^2} + \sum_{i=0}^{k} s_i^2 \underbrace{\|g_i\|^2}_{\leq G^2} - \beta \sum_{i=0}^{k} s_i \\
&\leq R^2 + G^2 \sum_{i=0}^{k} s_i^2 - \beta \sum_{i=0}^{k} s_i
\end{aligned}
$$

Arriving at the contradiction

- We have shown

$$
\begin{aligned}
&\|x_{k+1} - \tilde{x}\|^2 \leq R^2 + G^2 \sum_{i=0}^{k} s_i^2 - \beta \sum_{i=0}^{k} s_i \\
\Rightarrow &0 \leq R^2 + G^2 \sum_{i=0}^{k} s_i^2 - \beta \sum_{i=0}^{k} s_i \\
\Leftrightarrow &\beta \sum_{i=0}^{k} s_i \leq R^2 + G^2 \sum_{i=0}^{k} s_i^2 \\
\Rightarrow &\underbrace{\beta}_{\text{finite},>0} \underbrace{\lim_{k\to\infty} \sum_{i=0}^{k} s_i}_{\infty} \leq R^2 + G^2 \underbrace{\lim_{k\to\infty} \sum_{i=0}^{k} s_i^2}_{\text{finite}}
\end{aligned}
$$

- But this leads to contradiction because as $k \to \infty$, the LHS will blow up, but RHS will converge to a finite number
- So, our initial assumption $\lim_{k\to\infty} f_{0,k}^\star > p^\star$ cannot be correct
- Only possibility is: $\lim_{k\to\infty} f_{0,k}^\star = p^\star$

Summary of the proof structure

- This type of proof structure is extremely common in optimization
- We assumed opposite of what we wanted to prove, goal is a proof by contradiction
- Created $\tilde{x}$ a convex combination of $x^\star$ and x_{sf}
- Showed that $\tilde{x}$ is $\epsilon/2$ suboptimal and $-\gamma$ strictly feasible
- Then showed that

  $$\|x_{i+1} - \tilde{x}\|^2 \leq \|x_i - \tilde{x}\|^2 + s_i^2\|g_i\|^2 - s_i\beta$$

  where $-\beta = \max(-\epsilon, -2\gamma)$
- Did a telescoping sum which gave us

  $$\beta \sum_{i=0}^{k} s_i \leq R^2 + G^2 \sum_{i=0}^{k} s_i^2,$$

  leading to contradiction

Performance of subgradient methods in practice

- In practice the stepsizes $s_k$are often chosen based on heuristic - the homework gives one such heuristic
- The subgradient method can be (and often is) slow in practice compared to second-order methods

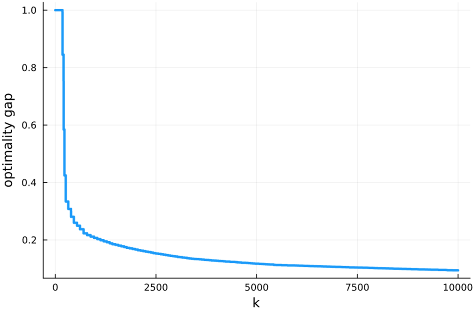


Figure: Typical convergence behavior of subgradient-based methods on solving linear programs

Performance of subgradient methods in practice

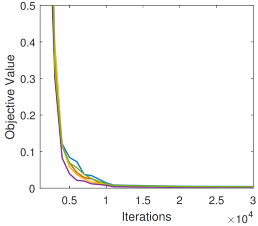


Figure: Typical convergence behavior of subgradient-based methods on training neural networks

- This is one of the painful observation in training neural networks, but this is nothing to be upset about

Performance of subgradient methods in practice

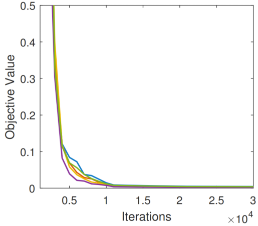


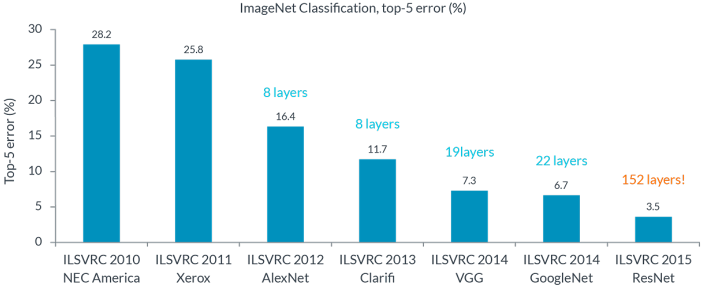
Figure: Typical convergence behavior of subgradient-based methods on training neural networks

- This is one of the painful observation in training neural networks, but this is nothing to be upset about
- "The more you know who you are and what you want, the less you let things upset you."-Lost in translation
- What do you expect from an algorithm that is just a
  - few lines of code, has no line search, uses only subgradient?

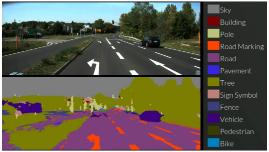
The Hidden Convex Optimization Landscape of Deep Neural Networks

Tolga Ergen
Stanford University

Deep Learning Revolution



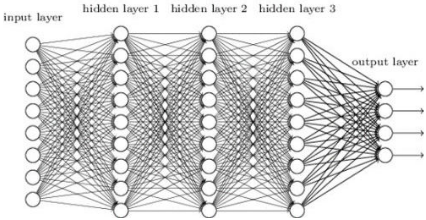
Impact of Deep Learning



Y. LeCun, Y. Bengio, G. Hinton (2015)

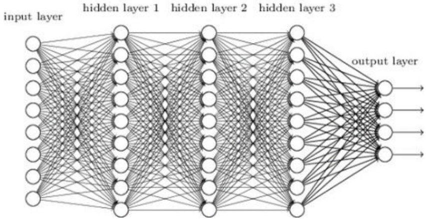
Table of Contents

What are the challenges?



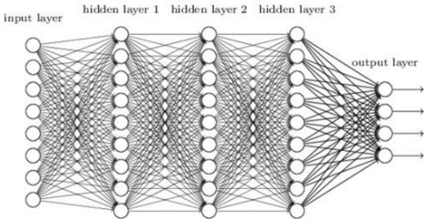
- extremely high dimensional training problem

What are the challenges?



- extremely high dimensional training problem
  - 152 layer ResNet-152: 60.2 Million parameters (2015)

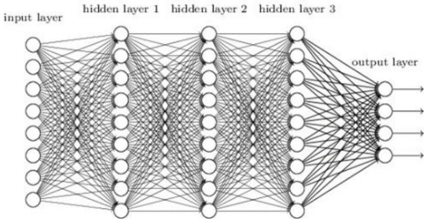
What are the challenges?



- extremely high dimensional training problem
  - 152 layer ResNet-152: 60.2 Million parameters (2015)
  - GPT[1]-3 language model: 175 Billion parameters (May 2020)

[1] OpenAI General Purpose Transformer

What are the challenges?

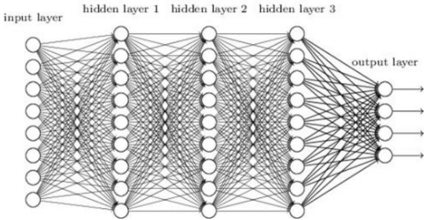


- extremely high dimensional training problem
  - 152 layer ResNet-152: 60.2 Million parameters (2015)
  - GPT[1]-3 language model: 175 Billion parameters (May 2020)
  - BAAI[2] multi-modal model: 1.75 Trillion parameters (June 2021)

---

[1] OpenAI General Purpose Transformer
[2] The Beijing Academy of Artificial Intelligence

What are the challenges?

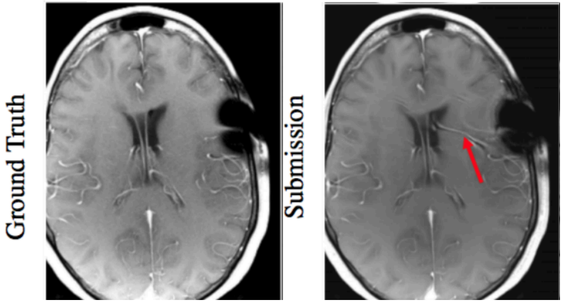


- extremely high dimensional training problem
  - 152 layer ResNet-152: 60.2 Million parameters (2015)
  - GPT[1]-3 language model: 175 Billion parameters (May 2020)
  - BAAI[2] multi-modal model: 1.75 Trillion parameters (June 2021)
- complex black-box systems based on non-convex optimization
  - hard to interpret what the model is actually learning

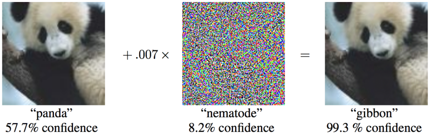
[1]OpenAI General Purpose Transformer
[2]The Beijing Academy of Artificial Intelligence

Interpretability is important

Example: Deep networks for MR image reconstruction (Fast MRI Challenge, 2020)

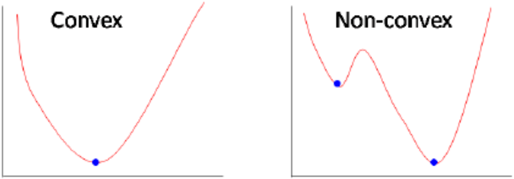


Adversarial examples



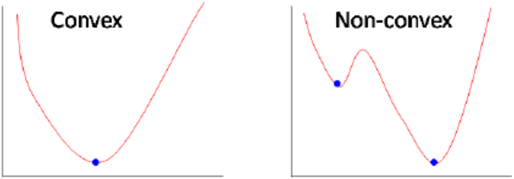
- adversarial examples, Szegedy et al., 2014, Goodfellow et al., 2015
- stop sign recognized as speed limit sign, Evtimov et al, 2017

Convex vs Non-convex



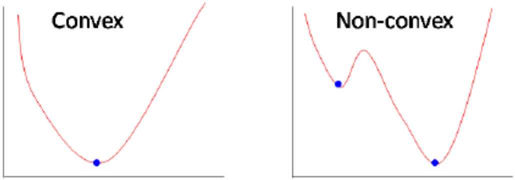
- **Convex:** least-squares, logistic regression, SVMs etc.

Convex vs Non-convex



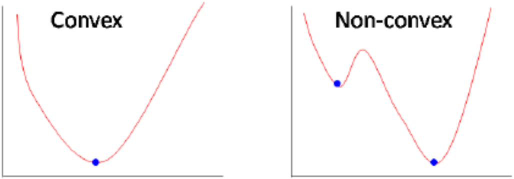
- **Convex:** least-squares, logistic regression, SVMs etc.
  - are extremely well understood

Convex vs Non-convex



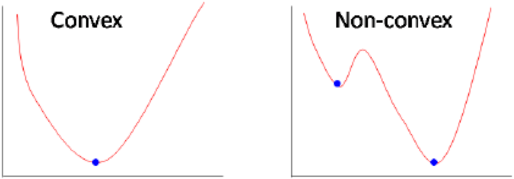
- **Convex:** least-squares, logistic regression, SVMs etc.
  - are extremely well understood
  - the choice of the solver, initialization, learning rate schedule do not matter

Convex vs Non-convex



- **Convex:** least-squares, logistic regression, SVMs etc.
  - are extremely well understood
  - the choice of the solver, initialization, learning rate schedule do not matter
  - Interpretable and insightful theorems

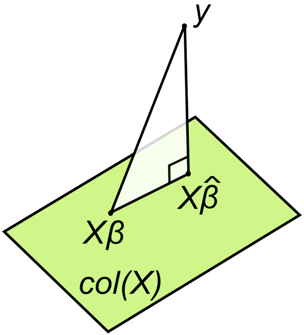
Convex vs Non-convex



- **Convex:** least-squares, logistic regression, SVMs etc.
  - are extremely well understood
  - the choice of the solver, initialization, learning rate schedule do not matter
  - Interpretable and insightful theorems
- **Non-convex:** neural networks
  - ???

Least Squares

$$\min_{\beta} \|X\beta - y\|_2^2$$



- convex optimality condition: $X^T X\beta = X^T y$
- efficient solvers: conjugate gradient (CG), preconditioned CG, QR, Cholesky...

Least Squares with L1 Regularization (Lasso)

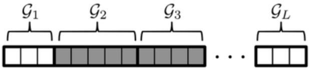
$$\min_{\boldsymbol{\beta}} \|\mathsf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- L1 norm $\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^{d} |\beta_i|$ encourages sparsity

Tibshirani (1996), Candes & Tao (2005), Donoho (2006)

Least Squares with Group L1 regularization (Group Lasso)

$$\min_{\boldsymbol{\beta}} \left\| \sum_{i=1}^{k} \mathrm{X}_i \boldsymbol{\beta}_i - \mathrm{y} \right\|_2^2 + \lambda \sum_{i=1}^{k} \|\boldsymbol{\beta}_i\|_2$$



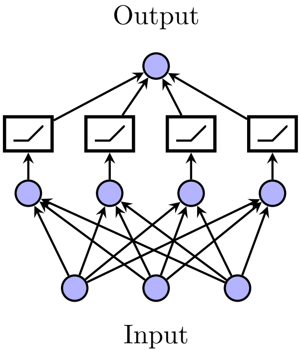
- encourages group sparsity in the solution $[\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_k]$, i.e., most blocks are zero
- convex optimization and convex regularization methods are well understood

Yuan & Lin (2007)

Table of Contents

Two-layer Neural Networks with ReLU Activation

Model:



Notation:

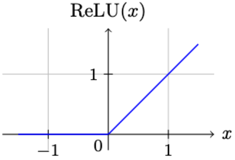
$X \in \mathbb{R}^{n \times d}$: Data matrix

$y \in \mathbb{R}^{n}$: Label vector

$\mathcal{L}(\cdot, \cdot)$: Arbitrary convex loss function

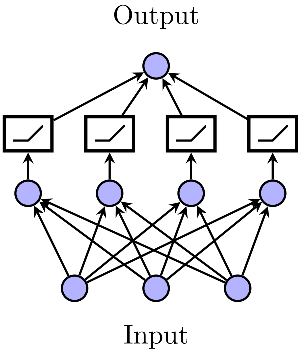
$\beta > 0$: Regularization coefficient

$W_1 \in \mathbb{R}^{d \times m}, w_2 \in \mathbb{R}^{m}$: Layer weights



Two-layer Neural Networks with ReLU Activation

Model:



Notation:

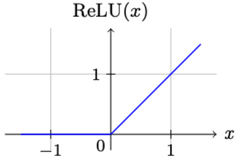
$X \in \mathbb{R}^{n \times d}$: Data matrix

$y \in \mathbb{R}^{n}$: Label vector

$\mathcal{L}(\cdot, \cdot)$: Arbitrary convex loss function

$\beta > 0$: Regularization coefficient

$W_1 \in \mathbb{R}^{d \times m}, w_2 \in \mathbb{R}^{m}$: Layer weights



Optimization problem:

$$p_{non-convex} := \min_{W_1, w_2} \mathcal{L}(\phi(XW_1)w_2, y) + \frac{\beta}{2}(\|W_1\|_F^2 + \|w_2\|_2^2)$$

where $\phi(x) = \mathrm{ReLU}(x) = (x)_+$ and $\mathcal{L}(\cdot, \cdot)$ is arbitrary convex loss function

Neural Networks are Convex Regularizers

Non-convex optimization problem:

$$p_{non-convex} = \min_{W_1 \in \mathbb{R}^{d \times m}, w_2 \in \mathbb{R}^m} \mathcal{L}(\phi(XW_1)w_2, y) + \frac{\beta}{2}(\|W_1\|_F^2 + \|w_2\|_2^2)$$

Convex optimization problem:

$$p_{convex} := \min_{u_i, v_i \in \mathcal{C}} \mathcal{L}\left(\sum_{i=1}^{P} D_i X(u_i - v_i), y\right) + \beta \sum_{i=1}^{P}(\|u_i\|_2 + \|v_i\|_2)$$

where $D_1, \ldots, D_P$ are fixed diagonal matrices

Neural Networks are Convex Regularizers

Non-convex optimization problem:

$$p_{non-convex} = \min_{\mathbf{W}_1 \in \mathbb{R}^{d\times m}, \mathbf{w}_2 \in \mathbb{R}^m} \mathcal{L}(\phi(\mathbf{X}\mathbf{W}_1)\mathbf{w}_2, \mathbf{y}) + \frac{\beta}{2}(\|\mathbf{W}_1\|_F^2 + \|\mathbf{w}_2\|_2^2)$$

Convex optimization problem:

$$p_{convex} := \min_{\mathbf{u}_i, \mathbf{v}_i \in \mathcal{C}} \mathcal{L}\left(\sum_{i=1}^{P} \mathbf{D}_i \mathbf{X}(\mathbf{u}_i - \mathbf{v}_i), \mathbf{y}\right) + \beta \sum_{i=1}^{P}(\|\mathbf{u}_i\|_2 + \|\mathbf{v}_i\|_2)$$

where $\mathbf{D}_1, \ldots, \mathbf{D}_P$ are fixed diagonal matrices

Theorem ([3])

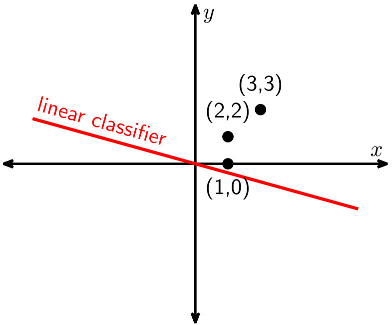
$p_{non-convex} = p_{convex}$ *and an optimal solution to* $p_{non-convex}$ *can be recovered from optimal non-zero* $\{\mathbf{u}_i^*, \mathbf{v}_i^*\}_{i=1}^{P}$ *as follows*

$$\mathbf{w}_{1i}^* = \frac{\mathbf{u}_i^*}{\sqrt{\|\mathbf{u}_i^*\|_2}}, w_{2i}{}^* = \sqrt{\|\mathbf{u}_i\|_2} \textit{ or } \mathbf{w}_{1i}^* = \frac{\mathbf{v}_i^*}{\sqrt{\|\mathbf{v}_i^*\|_2}}, w_{2i}{}^* = -\sqrt{\|\mathbf{v}_i\|_2}.$$

[3] M. Pilanci, T. Ergen, **"Neural Networks are Convex Regularizers...", ICML 2020**

Hyperplane Arrangements (D_i)

$$n = 3 \text{ samples in } \mathbb{R}^d,\ d = 2,\ X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix},\ y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

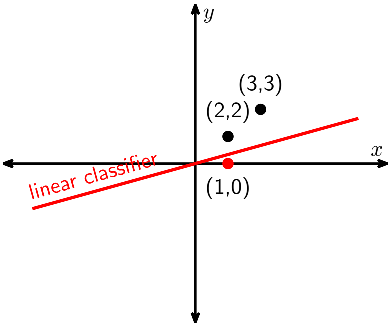


$$D_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ D_1 X = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix}$$

$$\implies (Xw_1)_+ = D_1 X w_1$$

Hyperplane Arrangements (D_i)

$$n = 3 \text{ samples in } \mathbb{R}^d,\ d = 2,\ \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix},\ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

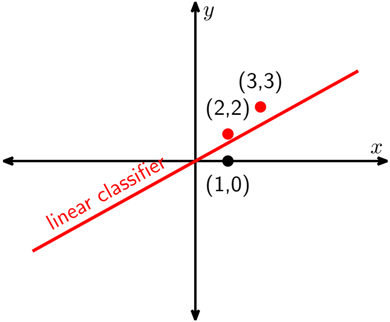


$$\mathbf{D}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & {\color{red}0} \end{bmatrix},\ \mathbf{D}_2\mathbf{X} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ {\color{red}0} & {\color{red}0} \end{bmatrix}$$

$$\implies (\mathbf{X}\mathbf{w}_2)_+ = \mathbf{D}_2\mathbf{X}\mathbf{w}_2$$

Hyperplane Arrangements (D_i)

$$n = 3 \text{ samples in } \mathbb{R}^d,\ d = 2,\ \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix},\ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

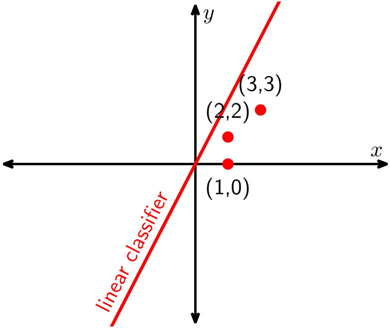


$$\mathbf{D}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \mathbf{D}_3\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\implies (\mathbf{X}\mathbf{w}_3)_+ = \mathbf{D}_3\mathbf{X}\mathbf{w}_3$$

Hyperplane Arrangements ($\mathbf{D}_i$)

$$n = 3 \text{ samples in } \mathbb{R}^d,\ d = 2,\ \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix},\ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$



$$\mathbf{D}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \mathbf{D}_4\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\implies (\mathbf{X}\mathbf{w}_4)_+ = \mathbf{D}_4\mathbf{X}\mathbf{w}_4$$

Example: Convex Program for $n = 3$, $d = 2$

$$n = 3 \text{ samples in } \mathbb{R}^d,\ d = 2,\ \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix},\ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$\min_{\{\mathbf{u}_i, \mathbf{v}_i\}_{i=1}^3} \frac{1}{2} \left\| \mathbf{D}_1 \mathbf{X}(\mathbf{u}_1 - \mathbf{v}_1) + \mathbf{D}_2 \mathbf{X}(\mathbf{u}_2 - \mathbf{v}_2) + \mathbf{D}_3 \mathbf{X}(\mathbf{u}_3 - \mathbf{v}_3) - \mathbf{y} \right\|_2^2 + \beta \sum_{i=1}^{3} (\|\mathbf{u}_i\|_2 + \|\mathbf{v}_i\|_2)$$

subject to

$$\mathbf{D}_1 \mathbf{X}[\mathbf{u}_1\ \mathbf{v}_1] \geq 0,\ (\mathbf{I}_n - \mathbf{D}_1)\mathbf{X}[\mathbf{u}_1\ \mathbf{v}_1] \leq 0$$

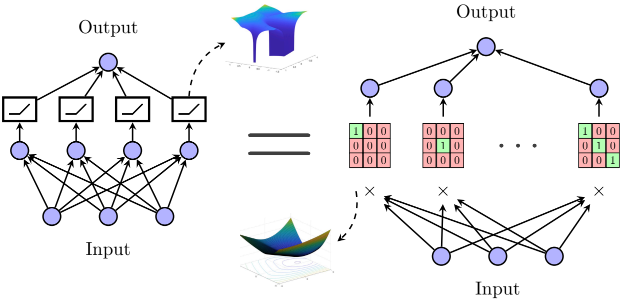
$$\mathbf{D}_2 \mathbf{X}[\mathbf{u}_2\ \mathbf{v}_2] \geq 0,\ (\mathbf{I}_n - \mathbf{D}_2)\mathbf{X}[\mathbf{u}_2\ \mathbf{v}_2] \leq 0$$

$$\mathbf{D}_3 \mathbf{X}[\mathbf{u}_3\ \mathbf{v}_3] \geq 0,\ (\mathbf{I}_n - \mathbf{D}_3)\mathbf{X}[\mathbf{u}_3\ \mathbf{v}_3] \leq 0$$

equivalent to the two-layer ReLU neural network!

Neural Networks as High-dimensional Variable Selectors

Non-convex **Convex**



$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times d} \xRightarrow[\text{ReLU Network}]{} \tilde{X} = \begin{bmatrix} D_1 X & \ldots & D_P X \end{bmatrix} \in \mathbb{R}^{n \times dP}$$

$\boxed{\text{ReLU networks} \equiv \text{convex model selection applied to } \tilde{X}}$

Training Complexity

Given the data $\mathbf{X} \in \mathbb{R}^{n \times d}$, learning two-layer ReLU neural networks with m neurons: $f(\mathbf{X}) = \sum_{j=1}^{m} \left(\mathbf{X}\mathbf{w}_{1j}\right)_+ w_{2j}$

- Previous result: $\mathcal{O}(2^m n^{dm})$ (Arora et al., ICLR 2018)

Training Complexity

Given the data $\mathrm{X} \in \mathbb{R}^{n \times d}$, learning two-layer ReLU neural networks with m neurons: $f(\mathrm{X}) = \sum_{j=1}^{m} \left(\mathrm{X}\mathrm{w}_{1j}\right)_{+} w_{2j}$

- Previous result: $\mathcal{O}(2^{m} n^{dm})$ (Arora et al., ICLR 2018)
- Our convex program: $\boxed{\mathcal{O}\left(\left(\frac{n}{r}\right)^{r}\right), \textbf{ where } r := \mathrm{rank}(\mathrm{X})}$

n : # of samples, d : # of features

Training Complexity

Given the data $\mathsf{X} \in \mathbb{R}^{n \times d}$, learning two-layer ReLU neural networks with m neurons: $f(\mathsf{X}) = \sum_{j=1}^{m} \left(\mathsf{X}\mathsf{w}_{1j}\right)_+ w_{2j}$

- Previous result: $\mathcal{O}(2^m n^{dm})$ (Arora et al., ICLR 2018)
- Our convex program: $\boxed{\mathcal{O}\left(\left(\frac{n}{r}\right)^r\right), \textbf{ where } r := \mathrm{rank}(\mathsf{X})}$

n : # of samples, d : # of features

  - polynomial in n, d, and m for fixed rank r

Training Complexity

Given the data $\mathsf{X} \in \mathbb{R}^{n \times d}$, learning two-layer ReLU neural networks with m neurons: $f(\mathsf{X}) = \sum_{j=1}^{m} \left(\mathsf{X}\mathsf{w}_{1j}\right)_{+} w_{2j}$

- Previous result: $\mathcal{O}(2^m n^{dm})$ (Arora et al., ICLR 2018)
- Our convex program: $\boxed{\mathcal{O}\left(\left(\frac{n}{r}\right)^r\right)\text{, where } r := \mathrm{rank}(\mathsf{X})}$

n : # of samples, d : # of features

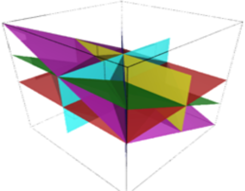
  - polynomial in n, d, and m for fixed rank r
  - exponential in d for full rank data $r = d$. This can not be improved unless $P = NP$ even for $m = 1$.

Table of Contents

Convolutional Hyperplane Arrangements

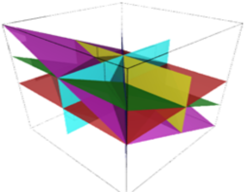
Fully Connected(FC) Arrangements: Let $\mathsf{X} \in \mathbb{R}^{n \times d}$ and $r = \mathrm{rank}(\mathsf{X})$

$$|\{\mathsf{D}_i\}| := \left|\{\mathrm{sign}(\mathsf{X}\mathsf{w}) : \mathsf{w} \in \mathbb{R}^d\}\right| \leq \mathcal{O}\left(\left(\frac{n}{r}\right)^r\right)$$



Fully Connected(FC) Arrangements: Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $r = \mathrm{rank}(\mathbf{X})$

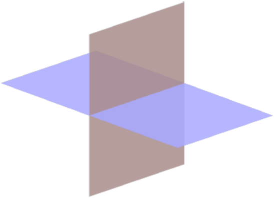
$$|\{\mathbf{D}_i\}| := \left|\{\mathrm{sign}(\mathbf{X}\mathbf{w}) : \mathbf{w} \in \mathbb{R}^d\}\right| \leq \mathcal{O}\left(\left(\frac{n}{r}\right)^r\right)$$



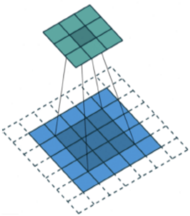
Convolutional Arrangements: Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be partitioned into K patch matrices as $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots & \mathbf{X}_K \end{bmatrix}$, where $\mathbf{X}_k \in \mathbb{R}^{n \times h}$

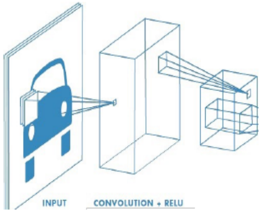
$$\left|\{\mathbf{D}_i^k\}\right| := \left|\{\mathrm{sign}(\mathbf{X}_k\mathbf{w}) : \mathbf{w} \in \mathbb{R}^h\}\right| \leq \mathcal{O}\left(\left(\frac{nK}{h}\right)^h\right)$$

$h(\ll r)$: filter size, K : # of patches



CNNs can be optimized in fully polynomial time





Given the data $\mathbf{X} \in \mathbb{R}^{n \times d}$, learning two-layer convolutional ReLU neural networks with m filters: $f(\mathbf{X}) = \sum_{k=1}^{K} \sum_{j=1}^{m} \left(\mathbf{X}_k \mathbf{w}_{1j}\right)_+ w_{2jk}$

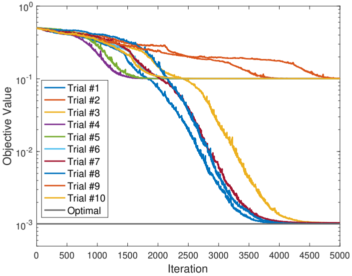
Convex program can be globally optimized with complexity $\mathcal{O}\left(\left(\frac{nK}{h}\right)^h\right)$, where $h \ll r \leq \min\{n, d\}$[4]

n : # of data samples, h : filter size, K : # of patches

[4]T. Ergen, M. Pilanci **“Implicit Convex Regularizers of CNN Architectures ...”, ICLR 2021**

Numerical Experiments: Two-layer Fully Connected ReLU Network

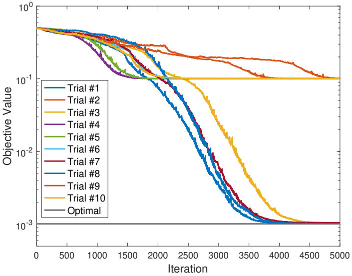
Training cost of a two-layer ReLU network trained with SGD (10 initialization trials) on a toy dataset ($d = 2$)



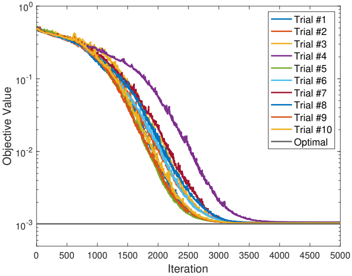
(a) $m = 8$

Numerical Experiments: Two-layer Fully Connected ReLU Network

Training cost of a two-layer ReLU network trained with SGD (10 initialization trials) on a toy dataset ($d = 2$)



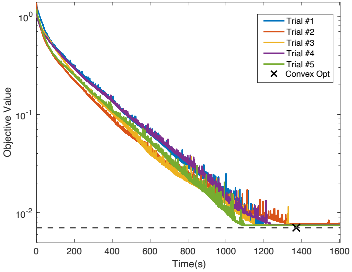
(a) $m = 8$



(b) $m = 50$

Numerical Experiments: Two-layer Convolutional Network on CIFAR

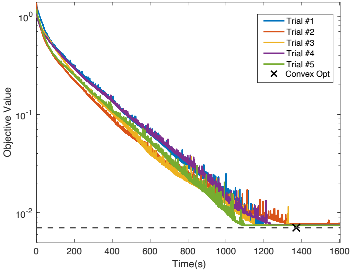
Binary classification on a subset of the CIFAR dataset



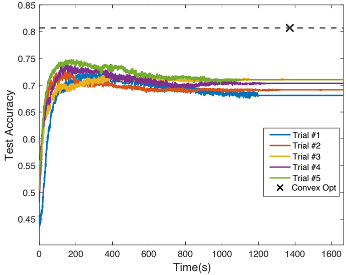
(a) Objective value

Numerical Experiments: Two-layer Convolutional Network on CIFAR

Binary classification on a subset of the CIFAR dataset



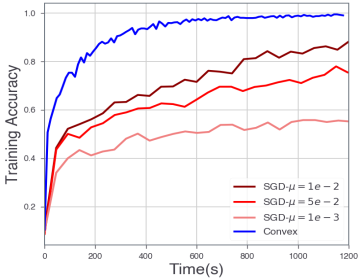
(a) Objective value



(b) Test accuracy

SGD for the Convex Program vs SGD for the Non-convex Problem

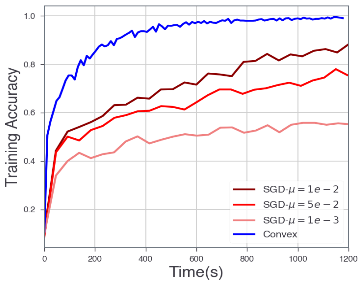
10-class classification on the CIFAR-10 dataset (n = 50000, d = 3072)



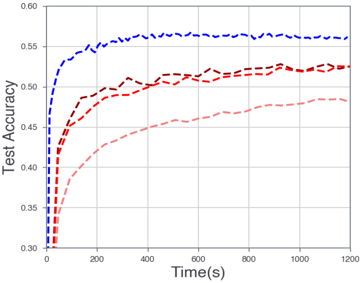
(a) Training accuracy

SGD for the Convex Program vs SGD for the Non-convex Problem

10-class classification on the CIFAR-10 dataset (n = 50000, d = 3072)



(a) Training accuracy



(b) Test accuracy)

Table of Contents

Vector Output Networks: Nuclear Norm Regularization

Non-convex optimization problem with C outputs/classes:

$$p_{non-convex} = \min_{W_1 \in \mathbb{R}^{d \times m}, W_2 \in \mathbb{R}^{m \times C}} \mathcal{L}(\phi(XW_1)W_2, Y) + \frac{\beta}{2}(\|W_1\|_F^2 + \|W_2\|_F^2)$$

Vector Output Networks: Nuclear Norm Regularization

Non-convex optimization problem with C outputs/classes:

$$p_{non-convex} = \min_{W_1 \in \mathbb{R}^{d \times m}, W_2 \in \mathbb{R}^{m \times C}} \mathcal{L}(\phi(XW_1)W_2, Y) + \frac{\beta}{2}(\|W_1\|_F^2 + \|W_2\|_F^2)$$

Convex optimization problem:

$$p_{convex} = \min_{U_i \in \mathcal{C}} \mathcal{L}\left(\sum_{i=1}^{P} D_i X U_i, Y\right) + \beta \sum_{i=1}^{P} \|U_i\|_*$$

Theorem ([5])

$p_{non-convex} = p_{convex}$ and an optimal solution to $p_{non-convex}$ can be recovered from optimal non-zero $\{U_i^*\}_{i=1}^P$.

[5] A. Sahiner, T. Ergen, J. Pauly, M. Pilanci, **"Vector-output ReLU Neural Network Problems are Copositive Programs ..."**, ICLR 2021

Table of Contents

ReLU Networks with Batch Normalization (BN)

BN transforms a batch of data to zero mean and standard deviation one, and has two trainable parameters α, γ:

$$\mathsf{BN}_{\gamma,\alpha}(\mathsf{x}) = \frac{(\mathsf{I_d} - \frac{1}{d}\mathsf{1}\mathsf{1}^T)\mathsf{x}}{\|(\mathsf{I_d} - \frac{1}{d}\mathsf{1}\mathsf{1}^T)\mathsf{x}\|_2}\gamma + \alpha$$

ReLU Networks with Batch Normalization (BN)

BN transforms a batch of data to zero mean and standard deviation one, and has two trainable parameters α, γ:

$$\mathrm{BN}_{\gamma,\alpha}(\mathbf{x}) = \frac{(\mathbf{I_d} - \frac{1}{d}\mathbf{1}\mathbf{1}^T)\mathbf{x}}{\|(\mathbf{I_d} - \frac{1}{d}\mathbf{1}\mathbf{1}^T)\mathbf{x}\|_2}\gamma + \alpha$$

Non-convex optimization problem:

$$p_{non-convex} = \min_{\mathbf{W}_1,\mathbf{w}_2,\gamma,\alpha} \mathcal{L}(\mathrm{BN}_{\gamma,\alpha}(\phi(\mathbf{X}\mathbf{W}_1))\mathbf{w}_2, \mathbf{y}) + \frac{\beta}{2}(\|\mathbf{W}_1\|_F^2 + \|\mathbf{w}_2\|_2^2)$$

ReLU Networks with Batch Normalization (BN)

BN transforms a batch of data to zero mean and standard deviation one, and has two trainable parameters α, γ:

$$\mathrm{BN}_{\gamma,\alpha}(\mathbf{x}) = \frac{(\mathbf{I_d} - \frac{1}{d}\mathbf{1}\mathbf{1}^T)\mathbf{x}}{\|(\mathbf{I_d} - \frac{1}{d}\mathbf{1}\mathbf{1}^T)\mathbf{x}\|_2}\gamma + \alpha$$

Non-convex optimization problem:

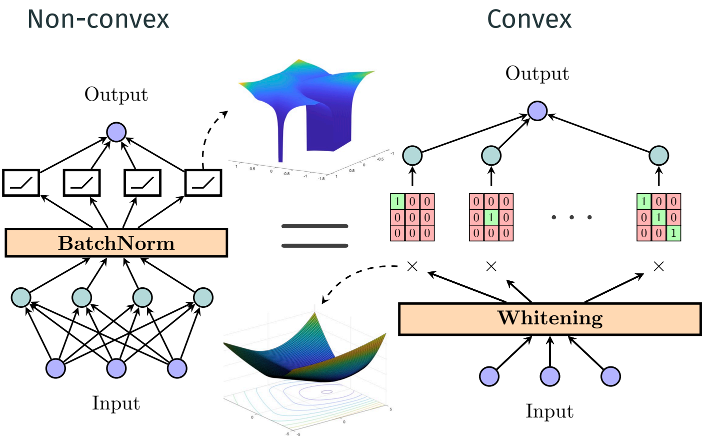
$$p_{non-convex} = \min_{\mathbf{W}_1, \mathbf{w}_2, \gamma, \alpha} \mathcal{L}(\mathrm{BN}_{\gamma,\alpha}(\phi(\mathbf{X}\mathbf{W}_1))\mathbf{w}_2, \mathbf{y}) + \frac{\beta}{2}(\|\mathbf{W}_1\|_F^2 + \|\mathbf{w}_2\|_2^2)$$

Convex optimization problem:[6]

$$p_{convex} = \min_{\mathbf{w}_i, \mathbf{v}_i \in \mathcal{C}} \mathcal{L}\left(\sum_{i=1}^{P} \mathbf{U}_i(\mathbf{w}_i - \mathbf{v}_i), \mathbf{y}\right) + \beta \sum_{i=1}^{P}(\|\mathbf{w}_i\|_2 + \|\mathbf{v}_i\|_2)$$

where $\mathbf{D}_i\mathbf{X} = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T$ is the compact SVD of $\mathbf{D}_i\mathbf{X}$, i.e., BatchNorm whitens local data

[6] T. Ergen*, A. Sahiner* et al, **"Demystifying Batch Normalization in ReLU Networks ...", ICLR 2022**



$$X \in \mathbb{R}^{n\times d} \xRightarrow[\text{ReLU+BN}]{} \tilde{X} = \begin{bmatrix} D_1 U & \dots & D_P U \end{bmatrix} \in \mathbb{R}^{n\times dP}$$

$\boxed{\text{ReLU+BN} \equiv \text{Sparse convex model applied to whitened data } \tilde{X}}$

Deep ReLU Networks with BN

Model: $f_{\theta,L}(\mathsf{X}) := \mathsf{A}^{(L-1)}\mathsf{W}^{(L)}, \quad \text{where } \mathsf{A}^{(l)} := \left(\mathrm{BN}_{\gamma,\alpha}\left(\mathsf{A}^{(l-1)}\mathsf{W}^{(l)}\right)\right)_+$

Theorem

Assume the network is overparameterized s.t. $\mathrm{Range}(\mathsf{A}^{(L-2)}) = \mathbb{R}^n$*, then optimal solution in closed-form is as follows*

$$\begin{aligned}
\left(\mathbf{w}_j^{(L-1)^*}, \mathbf{w}_j^{(L)^*}\right) &= \left(\mathsf{A}^{(L-2)^\dagger}\mathbf{y}_j, \left(\|\mathbf{y}_j\|_2 - \beta\right)_+ \mathbf{e}_j\right) \\
\left(\gamma_j^{(L-1)^*}, \alpha_j^{(L-1)^*}\right) &= \left(\frac{\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{y}_j\|_2}{\|\mathbf{y}_j\|_2}, \frac{\mathbf{1}^T\mathbf{y}_j}{\sqrt{n}\|\mathbf{y}_j\|_2}\right)
\end{aligned}, \quad \forall j \in [C]$$

where C is the number of classes/outputs and $\mathbf{e}_j$ is the jth ordinary basis vector.

Deep ReLU Networks with BN

Model: $f_{\theta,L}(\mathbf{X}) := \mathbf{A}^{(L-1)}\mathbf{W}^{(L)}$, where $\mathbf{A}^{(l)} := \left(\mathrm{BN}_{\gamma,\alpha}\left(\mathbf{A}^{(l-1)}\mathbf{W}^{(l)}\right)\right)_+$

Theorem

Assume the network is overparameterized s.t. $\mathrm{Range}(\mathbf{A}^{(L-2)}) = \mathbb{R}^n$, *then optimal solution in closed-form is as follows*

$$\left(\mathbf{w}_j^{(L-1)*}, \mathbf{w}_j^{(L)*}\right) = \left(\mathbf{A}^{(L-2)\dagger}\mathbf{y}_j, \left(\|\mathbf{y}_j\|_2 - \beta\right)_+ \mathbf{e}_j\right)$$
$$\left(\gamma_j^{(L-1)*}, \alpha_j^{(L-1)*}\right) = \left(\frac{\|\mathbf{y}_j - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{y}_j\|_2}{\|\mathbf{y}_j\|_2}, \frac{\mathbf{1}^T\mathbf{y}_j}{\sqrt{n}\|\mathbf{y}_j\|_2}\right), \ \forall j \in [C]$$

where C is the number of classes/outputs and $\mathbf{e}_j$ is the jth ordinary basis vector.

This also explains **Neural Collapse** in (Papyan et al., 2020)

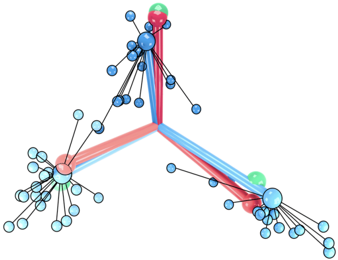
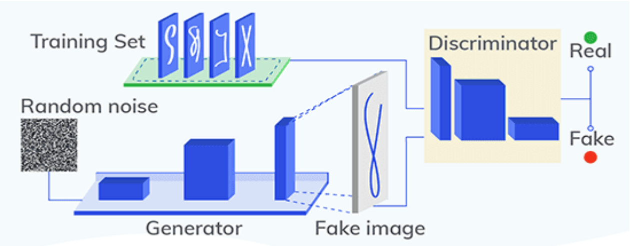


Table of Contents

Convex Generative Adversarial Networks (GANs)



Wasserstein GAN parameterized with neural networks:

$$p_{non-convex} = \min_{\theta_g} \max_{\theta_d} \mathbb{E}_{\mathbf{x} \sim p_x}[D_{\theta_d}(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_z}[D_{\theta_d}(G_{\theta_g}(\mathbf{z}))],$$

Theorem ([7])

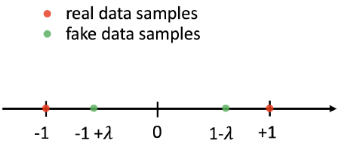
Two layer generator two layer discriminator WGAN problems are convex-concave games.

[7] A. Sahiner*, T. Ergen* et al, **"Hidden Convexity of Wasserstein GANs ...", ICLR 2022**

Two-layer ReLU discriminator/generator WGANs for 1D data

optimal solution can be found in closed form using convex optimality conditions

λ is the weight decay regularization parameter of the discriminator



$\lambda \leq 1$

no mode collapse

Two-layer ReLU discriminator/generator WGANs for 1D data

optimal solution can be found in closed form using convex optimality conditions

λ is the weight decay regularization parameter of the discriminator

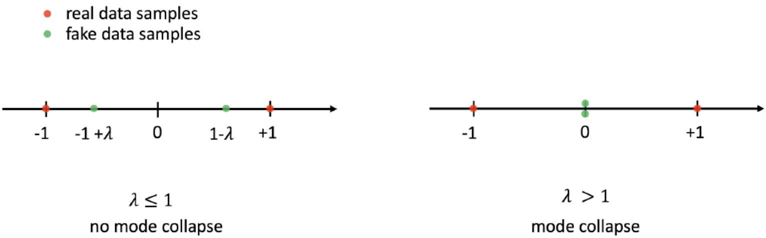
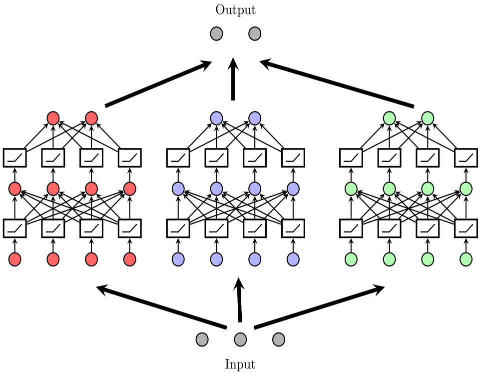


Table of Contents

Three-layer ReLU Networks with parallel architecture

Model:



Notation:

$\mathbf{X} \in \mathbb{R}^{n\times d}$: Data matrix

$\mathbf{y} \in \mathbb{R}^{n}$: Label vector

$\mathcal{L}(\cdot,\cdot)$: Convex loss function

$\beta > 0$: Regularization coefficient

θ : All parameters

l and k : Layer and sub-network indices

$\mathbf{W}_{lk} \in \mathbb{R}^{m_{l-1}\times m_l}$: Weights

Optimization problem:

$$p_{non-convex} = \min_{\theta} \mathcal{L}\left(\sum_{k=1}^{K} \left((\mathbf{X}\mathbf{W}_{1k})_+ \mathbf{w}_{2k}\right)_+ w_{3k}, \mathbf{y}\right) + \frac{\beta}{2}\sum_{k=1}^{K}\sum_{l=1}^{3} \|\mathbf{W}_{lk}\|_F^2$$

Convex Program for Three-layer Neural Networks

Non-convex optimization problem:

$$p_{non-convex} = \min_{\theta} \mathcal{L}\left(\sum_{k=1}^{K} \left(\left(\mathbf{X}\mathbf{W}_{1k}\right)_{+} \mathbf{w}_{2k}\right)_{+} w_{3k}, \mathbf{y}\right) + \frac{\beta}{2} \sum_{k=1}^{K} \sum_{l=1}^{3} \|\mathbf{W}_{lk}\|_F^2$$

Theorem ([8][9])

The non-convex training problem can be equivalently stated as

$$\min_{\mathbf{w},\mathbf{w}' \in \mathcal{C}} \frac{1}{2} \left\| \tilde{\mathbf{X}} \left(\mathbf{w}' - \mathbf{w}\right) - \mathbf{y} \right\|_2^2 + \beta \left(\|\mathbf{w}\|_{2,1} + \|\mathbf{w}'\|_{2,1}\right)$$

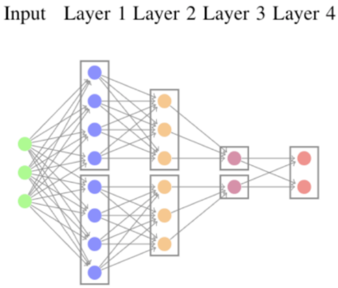
where $\|\cdot\|_{2,1}$ *is d dimensional group norm:* $\|\mathbf{w}\|_{2,1} := \sum_{i=1}^{P} \|\mathbf{w}_i\|_2$

$$\tilde{\mathbf{X}} := \begin{bmatrix} \tilde{\mathbf{X}}_s & 0 \\ 0 & \tilde{\mathbf{X}}_s \end{bmatrix}, \quad \tilde{\mathbf{X}}_s := \begin{bmatrix} \mathbf{D}_1^{(1)} \mathbf{D}_1^{(2)} \mathbf{X} & \dots & \mathbf{D}_i^{(1)} \mathbf{D}_j^{(2)} \mathbf{X} & \dots & \mathbf{D}_{P_1}^{(1)} \mathbf{D}_{P_2}^{(2)} \mathbf{X} \end{bmatrix}.$$

[8] T. Ergen, M. Pilanci **"Global Optimality Beyond Two Layers: Training Deep ReLU Networks via Convex Programs"**, ICML 2021

[9] T. Ergen, M. Pilanci **"Path Regularization: A Convexity and Sparsity Inducing Regularization for Parallel ReLU Networks"**, arXiv:2110.09548

Deep ReLU Networks (Depth $L > 3$)



arbitrarily deep ReLU neural networks with parallel architecture

Theorem ([10][11])

There is a convex program for arbitrarily deep linear and ReLU networks such that $p_{non-convex} = p_{convex}$

[10] T. Ergen, M. Pilanci, **"Revealing the Structure of Deep Neural Networks via Convex Duality", ICML 2021**

[11] Y. Wang, T. Ergen, M. Pilanci, **"Parallel Deep Neural Networks Have Zero Duality Gap", ICLR 2023**

Plan for the rest of the talk

How to make neural network training and inference more energy/memory/data efficient?

- optimal quantization of network activations
- layerwise learning of deep neural network models
- Transfer learning with pretrained models

Quantizing Network Activations

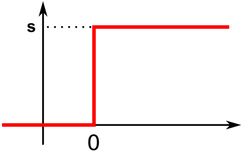
discrete valued activation $\sigma(\cdot)$

- computational efficiency
- intermediate feature vectors are discrete (e.g., 0-1 valued)
- enables efficient storage of activation patterns

Threshold Activation Networks

- **Threshold activations with a trainable amplitude:**

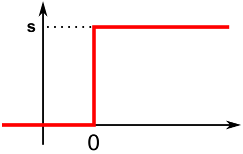
$$\sigma_s(x) := s\mathbb{1}\{x \geq 0\} = \begin{cases} s & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



- **Model:** $f(\mathrm{X}) = \sigma_{s_L}(\ldots\sigma_{s_1}(\mathrm{X}\mathrm{W}^{(1)})\mathrm{W}^{(2)}\ldots)\mathrm{W}^{(L)}$

- Threshold activations with a trainable amplitude:

$$\sigma_s(x) := s\mathbb{1}\{x \geq 0\} = \begin{cases} s & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



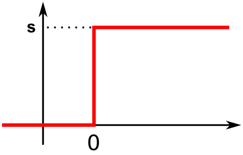
- Model: $f(\mathbf{X}) = \sigma_{\mathbf{s}_L}(\ldots\sigma_{\mathbf{s}_1}(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}\ldots)\mathbf{W}^{(L)}$
- Non-convex optimization problem:

$$p_{\text{nonconvex}} = \min_{\mathbf{W}^{(i)}, \mathbf{s}_i\, \forall i} \frac{1}{2}\|f(\mathbf{X}) - \mathbf{y}\|_2^2 + \frac{\beta}{2}\sum_{i=1}^{L}\left(\|\mathbf{W}^{(i)}\|_F^2 + \|\mathbf{s}_i\|_2^2\right)$$

Threshold Activation Networks

- Threshold activations with a trainable amplitude:

$$\sigma_s(x) := s\mathbb{1}\{x \geq 0\} = \begin{cases} s & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



- Model: $f(\mathbf{X}) = \sigma_{s_L}(\ldots\sigma_{s_1}(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}\ldots)\mathbf{W}^{(L)}$
- Non-convex optimization problem:

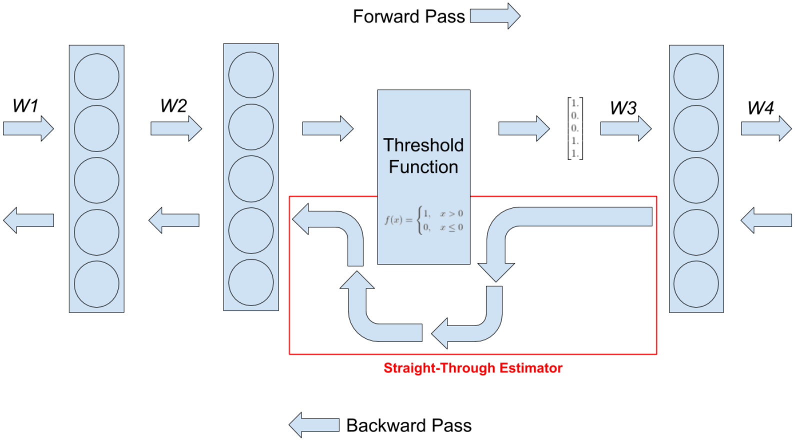
$$p_{\text{nonconvex}} = \min_{\mathbf{W}^{(i)}, \mathbf{s}_i\ \forall i} \frac{1}{2}\|f(\mathbf{X}) - \mathbf{y}\|_2^2 + \frac{\beta}{2}\sum_{i=1}^{L}\left(\|\mathbf{W}^{(i)}\|_F^2 + \|\mathbf{s}_i\|_2^2\right)$$

- Convex optimization problem: Enumerate the patterns $D_1, \ldots, D_P$ as columns of an $n \times P$ 0-1 valued matrix $\mathbf{D} \in \{0,1\}^{n \times P}$. Then we have[12]

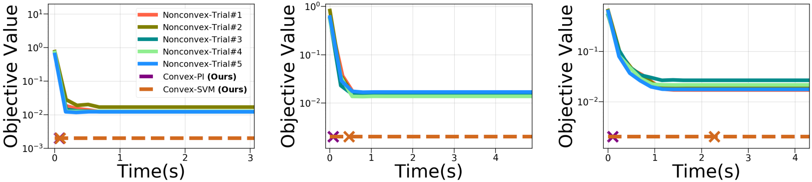
$$p_{\text{nonconvex}} = p_{\text{convex}} = \min_{\mathbf{w} \in \mathbb{R}^P} \frac{1}{2}\|\mathbf{D}\mathbf{w} - \mathbf{y}\|_2^2 + \beta\|\mathbf{w}\|_1$$

[12] T. Ergen, et al, **"Globally Optimal Training of Neural Networks with Threshold Activation Functions", ICLR 2023**

Standard heuristic: Straight-Through Estimator (STE)



Numerical Results for Threshold Activation Networks



(a) $(n, d) = (20, 100)$ **(b)** $(n, d) = (50, 50)$ **(c)** $(n, d) = (100, 20)$

Training comparison of our convex programs with the standard non-convex training heuristic (**Straight-Through Estimator (STE)** and SGD). In each case, our convex training algorithms achieve lower training objective. **STE does not yield optimal quantized networks.**

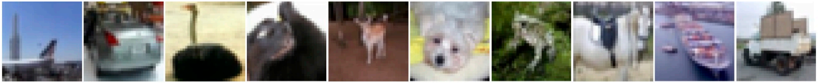
Numerical Results for Threshold Activation Networks

Performance comparison on CIFAR-10, MNIST, and UCI Machine Learning Repository datasets (Accuracy ↑, Time↓)

Dataset	Convex-Lasso (Ours)		Nonconvex-STE		Nonconvex-ReLU		Nonconvex-LReLU		Nonconvex-CReLU	
	Accuracy	Time(s)	Accuracy	Time(s)	Accuracy	Time(s)	Accuracy	Time(s)	Accuracy	Time(s)
CIFAR-10	0.816	8.9	0.81	83.5	0.803	85.8	0.798	92.1	0.808	87.1
MNIST	0.9991	39.4	0.9986	61.3	0.9984	63.4	0.9985	75.5	0.9985	64.9
bank	0.895	7.72	0.892	5.83	0.900	5.96	0.899	8.41	0.897	6.35
chess-krvkp	0.945	5.34	0.937	6.78	0.934	6.17	0.945	7.44	0.941	6.15
mammographic	0.818	2.64	0.808	5.40	0.803	6.51	0.801	5.76	0.817	5.29
oocytes-4d	0.787	2.23	0.787	5.61	0.756	7.09	0.723	6.22	0.732	5.79
oocytes-2f	0.799	1.99	0.776	5.24	0.774	6.97	0.775	5.89	0.783	5.46
ozone	0.967	3.65	0.967	6.30	0.967	6.89	0.967	7.86	0.967	6.20
pima	0.719	1.67	0.727	5.20	0.730	6.54	0.734	5.72	0.729	5.23
spambase	0.919	6.91	0.924	7.41	0.925	6.17	0.921	8.78	0.926	6.61
statlog-german	0.761	2.22	0.755	5.84	0.756	6.39	0.753	5.89	0.758	5.48
tic-tac-toe	0.980	1.89	0.954	4.97	0.932	6.63	0.926	5.61	0.951	5.18
titanic	0.778	0.35	0.790	5.06	0.784	6.30	0.796	6.24	0.784	5.19
Accuracy/Time	9/13	11/13	2/13	1/13	2/13	1/13	4/13	0/13	2/13	0/13

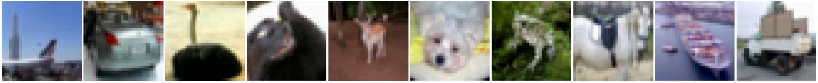
Numerical Results for Threshold Activation Networks

Original images



Numerical Results for Threshold Activation Networks

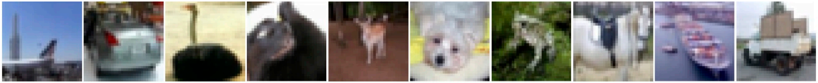
Original images



Nonconvex training

Numerical Results for Threshold Activation Networks

Original images



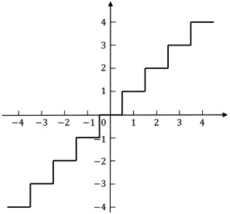
Nonconvex training

Convex training

Quantized (multi-step) Activation Networks

- **Threshold activations with a trainable amplitude:**

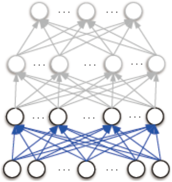
$$\sigma_s(x) = \begin{cases} \vdots & \\ -2s & \text{if } -2 \leq x < -1 \\ -s & \text{if } -1 \leq x < 0 \\ 0 & \text{if } 0 \leq x < 1 \\ s & \text{if } 1 \leq x < 2 \\ \vdots & \end{cases}$$



- **Convex optimization problem:** There is a fixed matrix $\mathbf{D} \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}^{n \times P}$ such that

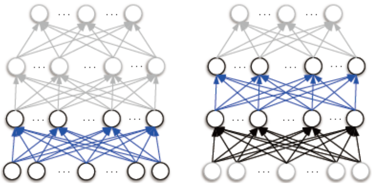
$$p_{\text{nonconvex}} = p_{\text{convex}} = \min_{\mathbf{w} \in \mathbb{R}^P} \frac{1}{2} \|\mathbf{D}\mathbf{w} - \mathbf{y}\|_2^2 + \beta \|\mathbf{w}\|_1$$

Convex Layer-Wise Training of Deep Networks



(i) train a two-layer network using convex optimization

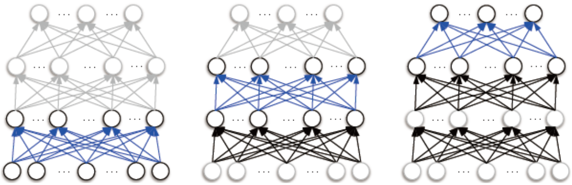
Convex Layer-Wise Training of Deep Networks



(i) train a two-layer network using convex optimization

(ii) fix the hidden layer to use as feature embedding

Convex Layer-Wise Training of Deep Networks

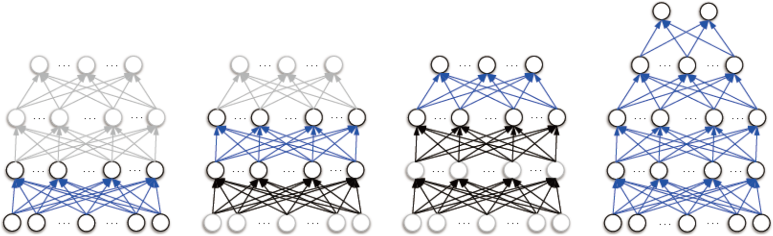


(i) train a two-layer network using convex optimization

(ii) fix the hidden layer to use as feature embedding

(iii) repeat two-layer network training on these features

Convex Layer-Wise Training of Deep Networks



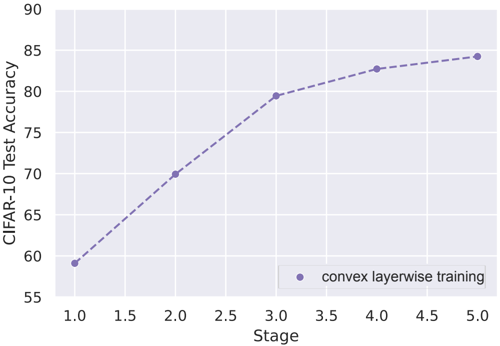
(i) train a two-layer network using convex optimization

(ii) fix the hidden layer to use as feature embedding

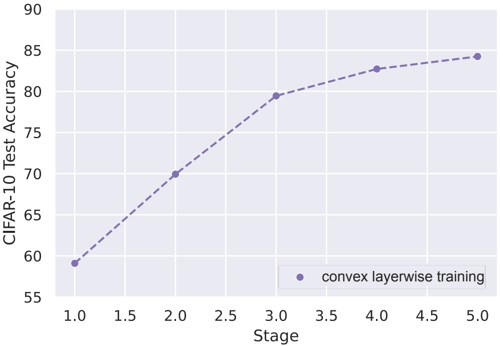
(iii) repeat two-layer network training on these features

- low memory consumption
- modular models: networks can keep evolving
- each convex model is trained to global optimality efficiently with no hyperparameter tuning

Numerical results for layer-wise convex learning on CIFAR-10

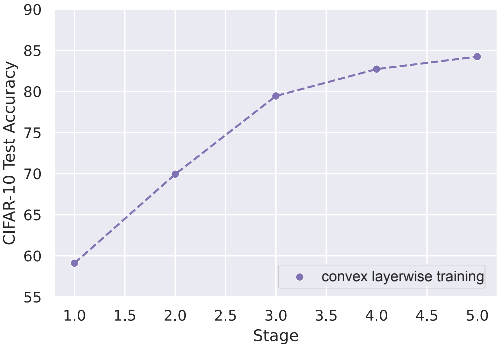


Numerical results for layer-wise convex learning on CIFAR-10



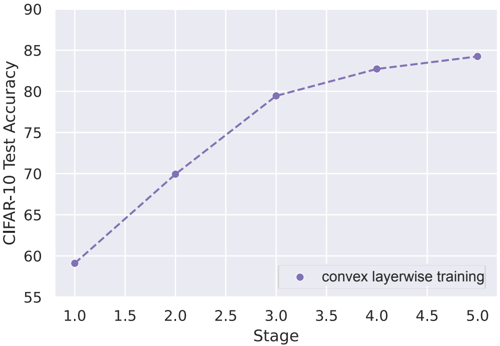
	Accuracy	Layers
End-to-end CNN	81.6%	6

Numerical results for layer-wise convex learning on CIFAR-10



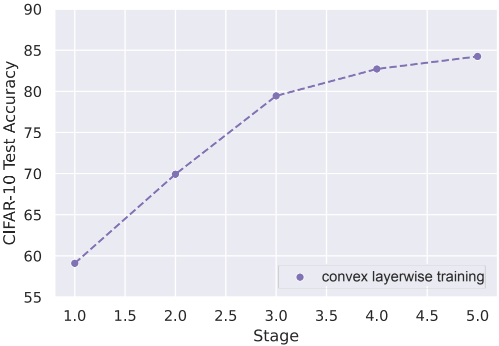
	Accuracy	Layers
End-to-end CNN	81.6%	6
AlexNet	82%	8

Numerical results for layer-wise convex learning on CIFAR-10



	Accuracy	Layers
End-to-end CNN	81.6%	6
AlexNet	82%	8
ResNet	83%	18

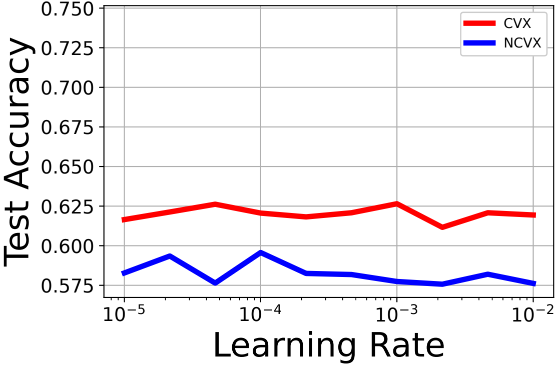
Numerical results for layer-wise convex learning on CIFAR-10



	Accuracy	Layers
End-to-end CNN	81.6%	6
AlexNet	82%	8
ResNet	83%	18
VGG	89%	16

Transfer Learning: Person Detection on the COCO dataset

Binary classification on the COCO dataset (n = 20k, 256×256 images)

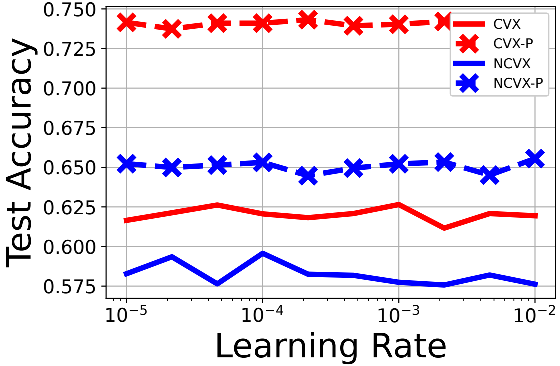


CVX: Convex CNN

NCVX: Nonconvex CNN

Transfer Learning: Person Detection on the COCO dataset

Binary classification on the COCO dataset (n = 20k, 256 × 256 images)



CVX: Convex CNN

NCVX: Nonconvex CNN

CVX-P: Convex CNN trained on MobileNetV3 features

NCVX-P: Nonconvex CNN trained on MobileNetV3 features

Takeaways and Open Problems

- ReLU network training is convex in high dimensions

Takeaways and Open Problems

- ReLU network training is convex in high dimensions
  - convex optimization theory and solvers can be applied

Takeaways and Open Problems

- ReLU network training is convex in high dimensions
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization

Takeaways and Open Problems

- ReLU network training is convex in high dimensions
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- Training complexity is polynomial-time w.r.t. the number of samples n and the feature dimension d

Takeaways and Open Problems

- ReLU network training is convex in high dimensions
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- Training complexity is polynomial-time w.r.t. the number of samples n and the feature dimension d
- **architecture search = regularizer search** (group ℓ_1, nuclear norm,...)

Takeaways and Open Problems

- ReLU network training is convex in high dimensions
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- Training complexity is polynomial-time w.r.t. the number of samples n and the feature dimension d
- **architecture search = regularizer search** (group ℓ_1, nuclear norm,...)

Future research directions:

- faster algorithms to solve high-dimensional convex programs

Takeaways and Open Problems

- ReLU network training is convex in high dimensions
  - convex optimization theory and solvers can be applied
  - don't need heuristics or hyperparameter search, e.g., learning rate and initialization
- Training complexity is polynomial-time w.r.t. the number of samples n and the feature dimension d
- **architecture search = regularizer search** (group ℓ_1, nuclear norm,...)

Future research directions:

- faster algorithms to solve high-dimensional convex programs
- Other NN architectures: Transformers, diffusion models ...

@tolgaergen_

stanford.edu/~ergen/

tolgaergen

References

stanford.edu/~ergen

- M. Pilanci, T. Ergen, “Neural Networks are Convex Reg ...”, ICML 2020
- T. Ergen, M. Pilanci “Implicit Convex Regularizers of CNN Architectures ...”, ICLR 2021
- A. Sahiner, T. Ergen et al “Vector-output ReLU Neural Network Problems are Copositive Programs ...”, ICLR 2021
- T. Ergen*, A. Sahiner* et al, “Demystifying Batch Normalization in ReLU Networks ...”, ICLR 2022
- A. Sahiner*, T. Ergen* et al, “Hidden Convexity of Wasserstein GANs ...”, ICLR 2022
- T. Ergen, M. Pilanci, “Global Optimality Beyond Two Layers ...”, ICML 2021
- T. Ergen, M. Pilanci, “Revealing the Structure of Deep Neural Networks via Convex Duality”, ICML 2021
- T. Ergen, M. Pilanci “ Convex Geometry and Duality of Over-parameterized Neural Networks”, JMLR
- T. Ergen, M. Pilanci, “Path Regularization: A Convexity and Sparsity Inducing Regularization for Parallel ReLU Networks”, arXiv:2110.09548

15.084/6.7220 Recitation 9: Projected Stochastic Gradient Descent and its Convergence

"It does not matter how slowly you go as long as you do not stop."-Confucius

Shuvomoy Das Gupta

Outline

HW4 and project

Stochastic gradient descent for nonsmooth convex setup

Minibatch SGD and momentum SGD

HW4

- HW4 will be uploaded on Monday, I am still working on the questions
- It will contain one question from the guest recitation, please watch the video
- HW3 will be grade will be uploaded this weekend

Project

- Hope project going well
- Please contact me if you face any issue

Outline

HW4 and project

Stochastic gradient descent for nonsmooth convex setup

Minibatch SGD and momentum SGD

Problem setup

- We are interested in solving the problem

$$p^\star = \left(\begin{array}{ll} \text{minimize}_{x \in \mathbb{R}^d} & f(x) \\ \text{subject to} & x \in C, \end{array} \right) \qquad (\mathcal{P})$$

  where we have the following assumptions regarding the nature of the problem.
  We assume:
  - $f : \mathbb{R}^d \to (-\infty, \infty]$ is a closed (epigraph closed), proper $(\mathbf{dom}\, f \neq \emptyset)$, and subdifferentiable convex function
  - C is a nonempty, closed, convex set, with $C \subseteq \mathbf{relint}\,\mathbf{dom}\, f$
  - $(\mathcal{P})$ has a finite optimal solution

Notation:

- all norms are Euclidean norm
- Π_C is projection onto the set C, will satisfy $\|\Pi_C(x) - \Pi_C(y)\| \leq \|x - y\|$

Stochastic oracle

We assume that given an iterate x_k, the stochastic oracle is capable of producing a random vector g_k with the following properties:

- (unbiased) $\forall_{k\geq 0}\ \mathbf{E}\left[g_k \mid x_k\right] \in \partial f(x_k)$, and
- (bounded variance) $\exists_{G>0}\ \forall_{k\geq 0}\ \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] \leq G^2$.

Stochastic gradient descent

1. initialization:
pick $x_0 \in C$ arbitrarily
2. main iteration:
for $k = 0, 1, 2, \ldots, K-1$
(i) pick stepsizes $\alpha_k > 0$ and random $g_k \in \mathbb{R}^d$ satisfying
$\mathbf{E}\left[g_k \mid x_k\right] \in \partial f(x_k)$ and $\mathbf{E}\left[\|g_k\|^2 \mid x_k\right] \leq G^2$
(ii) compute $x_{k+1} = \Pi_C\left(x_k - \alpha_k g_k\right)$
end for
3. return x_K

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|\Pi_C(x_k - \alpha_k g_k) - \Pi_C(x_\star)\|^2 \mid x_k\right] \quad \triangleright \text{ using } x_{k+1} = \Pi_C(x_k - \alpha_k g_k)
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|\Pi_C(x_k - \alpha_k g_k) - \Pi_C(x_\star)\|^2 \mid x_k\right] \ \triangleright \text{ using } x_{k+1} = \Pi_C(x_k - \alpha_k g_k) \\
&\leq \mathbf{E}\left[\|x_k - \alpha_k g_k - x_\star\|^2 \mid x_k\right] \ \triangleright \text{ using } \|\Pi_C(x) - \Pi_C(y)\| \leq \|x - y\|
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|\Pi_C(x_k - \alpha_k g_k) - \Pi_C(x_\star)\|^2 \mid x_k\right] \;\triangleright \text{ using } x_{k+1} = \Pi_C(x_k - \alpha_k g_k) \\
&\leq \mathbf{E}\left[\|x_k - \alpha_k g_k - x_\star\|^2 \mid x_k\right] \;\triangleright \text{ using } \|\Pi_C(x) - \Pi_C(y)\| \leq \|x - y\| \\
&= \mathbf{E}\left[\|(x_k - x_\star) - \alpha_k g_k\|^2 \mid x_k\right]
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|\Pi_C(x_k - \alpha_k g_k) - \Pi_C(x_\star)\|^2 \mid x_k\right] \;\; \triangleright \text{ using } x_{k+1} = \Pi_C(x_k - \alpha_k g_k) \\
&\leq \mathbf{E}\left[\|x_k - \alpha_k g_k - x_\star\|^2 \mid x_k\right] \;\; \triangleright \text{ using } \|\Pi_C(x) - \Pi_C(y)\| \leq \|x - y\| \\
&= \mathbf{E}\left[\|(x_k - x_\star) - \alpha_k g_k\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 \|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 \|g_k\|^2 - 2\alpha_k \left\langle x_k - x_\star;\, g_k \right\rangle \mid x_k\right]
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$$
$$= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 \|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]$$

- $\triangleright$ use linearity of expectation

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$$

$$= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2\|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]$$

▷ use linearity of expectation

$$= \mathbf{E}\left[\|x_k - x_\star\|^2 \mid x_k\right] + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \mathbf{E}\left[\langle x_k - x_\star;\, g_k\rangle \mid x_k\right]$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2\|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]
\end{aligned}$$

▷ use linearity of expectation

$$= \mathbf{E}\left[\|x_k - x_\star\|^2 \mid x_k\right] + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \mathbf{E}\left[\langle x_k - x_\star;\, g_k\rangle \mid x_k\right]$$

▷ use $\mathbf{E}\left[h(X)Y \mid X\right] = h(X)\mathbf{E}\left[Y \mid X\right]$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 \|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]
\end{aligned}
$$

▷ use linearity of expectation

$$
= \mathbf{E}\left[\|x_k - x_\star\|^2 \mid x_k\right] + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \mathbf{E}\left[\langle x_k - x_\star;\, g_k\rangle \mid x_k\right]
$$

▷ use $\mathbf{E}\left[h(X)Y \mid X\right] = h(X)\mathbf{E}\left[Y \mid X\right]$

$$
= \|x_k - x_\star\|^2 + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \langle x_k - x_\star;\, \mathbf{E}\left[g_k \mid x_k\right]\rangle
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$
\begin{aligned}
&\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \\
&= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 \|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]
\end{aligned}
$$

▷ use linearity of expectation

$$
= \mathbf{E}\left[\|x_k - x_\star\|^2 \mid x_k\right] + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \mathbf{E}\left[\langle x_k - x_\star;\, g_k\rangle \mid x_k\right]
$$

▷ use $\mathbf{E}\left[h(X)Y \mid X\right] = h(X)\mathbf{E}\left[Y \mid X\right]$

$$
= \|x_k - x_\star\|^2 + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \langle x_k - x_\star;\, \mathbf{E}\left[g_k \mid x_k\right]\rangle
$$

▷ use $\mathbf{E}\left[\|g_k\|^2 \mid x_k\right] \leq G^2$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$$

$$= \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 \|g_k\|^2 - 2\alpha_k \langle x_k - x_\star;\, g_k\rangle \mid x_k\right]$$

▷ use linearity of expectation

$$= \mathbf{E}\left[\|x_k - x_\star\|^2 \mid x_k\right] + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \mathbf{E}\left[\langle x_k - x_\star;\, g_k\rangle \mid x_k\right]$$

▷ use $\mathbf{E}\left[h(X)Y \mid X\right] = h(X)\mathbf{E}\left[Y \mid X\right]$

$$= \|x_k - x_\star\|^2 + \alpha_k^2 \mathbf{E}\left[\|g_k\|^2 \mid x_k\right] - 2\alpha_k \langle x_k - x_\star;\, \mathbf{E}\left[g_k \mid x_k\right]\rangle$$

▷ use $\mathbf{E}\left[\|g_k\|^2 \mid x_k\right] \leq G^2$

$$\leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k \langle x_k - x_\star;\, \mathbf{E}\left[g_k \mid x_k\right]\rangle$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$f(y) \geq f(x) + \langle f'(x);\, y - x \rangle$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$
\begin{aligned}
& f(y) \geq f(x) + \langle f'(x);\, y - x\rangle \\
\overset{y \leftarrow x_\star,\, x \leftarrow x_k}{\Rightarrow}\ & f(x_\star) \geq f(x_k) + \langle f'(x_k);\, x_\star - x_k\rangle
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$
\begin{aligned}
& f(y) \geq f(x) + \langle f'(x);\, y - x\rangle \\
\overset{y \leftarrow x_\star,\, x \leftarrow x_k}{\Rightarrow}\ & f(x_\star) \geq f(x_k) + \langle f'(x_k);\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) + \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_\star - x_k\rangle
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$
\begin{aligned}
& f(y) \geq f(x) + \langle f'(x);\, y - x\rangle \\
\overset{y \leftarrow x_\star,\, x \leftarrow x_k}{\Rightarrow}\ & f(x_\star) \geq f(x_k) + \langle f'(x_k);\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) + \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) - \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star\rangle
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$\begin{aligned}
& f(y) \geq f(x) + \langle f'(x);\, y - x\rangle \\
\overset{y\leftarrow x_\star,\, x\leftarrow x_k}{\Rightarrow}\ & f(x_\star) \geq f(x_k) + \langle f'(x_k);\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) + \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) - \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star\rangle \\
\Leftrightarrow\ & -\langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star\rangle \leq f(x_\star) - f(x_k)
\end{aligned}$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$
\begin{aligned}
& f(y) \geq f(x) + \langle f'(x);\, y - x\rangle \\
\overset{y \leftarrow x_\star, x \leftarrow x_k}{\Rightarrow}\ & f(x_\star) \geq f(x_k) + \langle f'(x_k);\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) + \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_\star - x_k\rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) - \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star\rangle \\
\Leftrightarrow\ & -\langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star\rangle \leq f(x_\star) - f(x_k)
\end{aligned}
$$

Convergence analysis: bound $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]$

Now recall that

$$
\begin{aligned}
& f(y) \geq f(x) + \langle f'(x);\, y - x \rangle \\
\overset{y \leftarrow x_\star,\, x \leftarrow x_k}{\Rightarrow}\ & f(x_\star) \geq f(x_k) + \langle f'(x_k);\, x_\star - x_k \rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) + \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_\star - x_k \rangle \\
\Leftrightarrow\ & f(x_\star) \geq f(x_k) - \langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star \rangle \\
\Leftrightarrow\ & -\langle \mathbf{E}\left[g_k \mid x_k\right];\, x_k - x_\star \rangle \leq f(x_\star) - f(x_k)
\end{aligned}
$$

So

$$
\begin{aligned}
\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] &\leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 + 2\alpha_k \left(-\langle x_k - x_\star;\, \mathbf{E}\left[g_k \mid x_k\right]\rangle\right) \\
&\leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 + 2\alpha_k \left(f(x_\star) - f(x_k)\right)
\end{aligned}
$$

$$
\therefore \mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k \left(f(x_k) - f(x_\star)\right)
$$

Using Adam's law and montonicity of expectation

- Adam's law says that $\mathbf{E}\left[\mathbf{E}\left[Y \mid X\right]\right] = \mathbf{E}\left[Y\right]$
- Montonicity of expectation $X \leq Y \Rightarrow \mathbf{E}\left[X\right] \leq \mathbf{E}\left[Y\right]$
- We have
  $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k \left(f(x_k) - f(x_\star)\right)$
- Taking expectation (wrt x_k) on both sides we get

$$\mathbf{E}\left[\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]\right] \leq \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k \left(f(x_k) - f(x_\star)\right)\right]$$

Using Adam's law and montonicity of expectation

- Adam's law says that $\mathbf{E}\left[\mathbf{E}\left[Y \mid X\right]\right] = \mathbf{E}\left[Y\right]$
- Montonicity of expectation $X \leq Y \Rightarrow \mathbf{E}\left[X\right] \leq \mathbf{E}\left[Y\right]$
- We have
  $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k \left(f(x_k) - f(x_\star)\right)$
- Taking expectation (wrt x_k) on both sides we get

$$\mathbf{E}\left[\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]\right] \leq \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k \left(f(x_k) - f(x_\star)\right)\right]$$
$$\Leftrightarrow \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_k - x_\star\|^2\right] + \mathbf{E}\left[\alpha_k^2 G^2\right] - \mathbf{E}\left[2\alpha_k \left(f(x_k) - f(x_\star)\right)\right]$$

Using Adam's law and montonicity of expectation

- Adam's law says that $\mathbf{E}\left[\mathbf{E}\left[Y \mid X\right]\right] = \mathbf{E}\left[Y\right]$
- Montonicity of expectation $X \leq Y \Rightarrow \mathbf{E}\left[X\right] \leq \mathbf{E}\left[Y\right]$
- We have
  $\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right] \leq \|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k\left(f(x_k) - f(x_\star)\right)$
- Taking expectation (wrt x_k) on both sides we get

$$
\begin{aligned}
&\mathbf{E}\left[\mathbf{E}\left[\|x_{k+1} - x_\star\|^2 \mid x_k\right]\right] \leq \mathbf{E}\left[\|x_k - x_\star\|^2 + \alpha_k^2 G^2 - 2\alpha_k\left(f(x_k) - f(x_\star)\right)\right] \\
&\Leftrightarrow \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_k - x_\star\|^2\right] + \mathbf{E}\left[\alpha_k^2 G^2\right] - \mathbf{E}\left[2\alpha_k\left(f(x_k) - f(x_\star)\right)\right] \\
&\Leftrightarrow \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_k - x_\star\|^2\right] + \alpha_k^2 G^2 - 2\alpha_k \mathbf{E}\left[f(x_k) - f(x_\star)\right]
\end{aligned}
$$

Using Adam's law and montonicity of expectation

- Lets do a telescoping sum

$$
\begin{aligned}
\mathbf{E}\left[\|x_{k+1}-x_\star\|^2\right]-\mathbf{E}\left[\|x_k-x_\star\|^2\right] &\leq -2\alpha_k \mathbf{E}\left[f(x_k)-f(x_\star)\right]+\alpha_k^2 G^2 \\
\mathbf{E}\left[\|x_k-x_\star\|^2\right]-\mathbf{E}\left[\|x_{k-1}-x_\star\|^2\right] &\leq -2\alpha_k \mathbf{E}\left[f(x_{k-1})-f(x_\star)\right]+\alpha_{k-1}^2 G^2 \\
&\vdots \\
\mathbf{E}\left[\|x_{m+1}-x_\star\|^2\right]-\mathbf{E}\left[\|x_m-x_\star\|^2\right] &\leq -2\alpha_m \mathbf{E}\left[f(x_m)-f(x_\star)\right]+\alpha_m^2 G^2,
\end{aligned}
$$

Using Adam's law and montonicity of expectation

- Lets do a telescoping sum

$$\mathbf{E}\left[\|x_{k+1}-x_\star\|^2\right]-\mathbf{E}\left[\|x_k-x_\star\|^2\right] \leq -2\alpha_k\mathbf{E}\left[f(x_k)-f(x_\star)\right]+\alpha_k^2G^2$$
$$\mathbf{E}\left[\|x_k-x_\star\|^2\right]-\mathbf{E}\left[\|x_{k-1}-x_\star\|^2\right] \leq -2\alpha_k\mathbf{E}\left[f(x_{k-1})-f(x_\star)\right]+\alpha_{k-1}^2G^2$$
$$\vdots$$
$$\mathbf{E}\left[\|x_{m+1}-x_\star\|^2\right]-\mathbf{E}\left[\|x_m-x_\star\|^2\right] \leq -2\alpha_m\mathbf{E}\left[f(x_m)-f(x_\star)\right]+\alpha_m^2G^2,$$

- We get

$$\mathbf{E}\left[\|x_{k+1}-x_\star\|^2\right]-\mathbf{E}\left[\|x_m-x_\star\|^2\right] \leq -2\sum_{i=m}^{k}\alpha_i\mathbf{E}\left[f(x_i)-f(x_\star)\right]+G^2\sum_{i=m}^{k}\alpha_i^2$$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$
- Also $\mathbf{E}\left[\min_i X_i\right] \leq \min_i \mathbf{E}\left[X_i\right]$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$
- Also $\mathbf{E}[\min_i X_i] \leq \min_i \mathbf{E}[X_i]$
- Now

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] - \mathbf{E}\left[\|x_m - x_\star\|^2\right] \leq -2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Leftrightarrow 0 \leq \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Rightarrow 0 \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=1}^{m} \alpha_i^2$$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$
- Also $\mathbf{E}[\min_i X_i] \leq \min_i \mathbf{E}[X_i]$
- Now

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] - \mathbf{E}\left[\|x_m - x_\star\|^2\right] \leq -2 \sum_{i=m}^{k} \alpha_i \mathbf{E}[f(x_i) - f(x_\star)] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Leftrightarrow 0 \leq \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}[f(x_i) - f(x_\star)] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Rightarrow 0 \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}[f(x_i) - f(x_\star)] + G^2 \sum_{i=1}^{m} \alpha_i^2$$

$$\Leftrightarrow \sum_{i=m}^{k} \alpha_i \mathbf{E}[f(x_i) - f(x_\star)] \leq \frac{1}{2} \left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2 \right)$$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$
- Also $\mathbf{E}[\min_i X_i] \leq \min_i \mathbf{E}[X_i]$
- Now

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] - \mathbf{E}\left[\|x_m - x_\star\|^2\right] \leq -2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Leftrightarrow 0 \leq \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Rightarrow 0 \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=1}^{m} \alpha_i^2$$

$$\Leftrightarrow \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] \leq \frac{1}{2}\left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2\right)$$

$$\Rightarrow \left(\sum_{i=m}^{k} \alpha_i\right)\left(\min_{i \in \{m,\ldots,k\}} \mathbf{E}\left[f(x_i) - f(x_\star)\right]\right) \leq \frac{1}{2}\left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2\right)$$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$
- Also $\mathbf{E}[\min_i X_i] \leq \min_i \mathbf{E}[X_i]$
- Now

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] - \mathbf{E}\left[\|x_m - x_\star\|^2\right] \leq -2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Leftrightarrow 0 \leq \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Rightarrow 0 \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=1}^{m} \alpha_i^2$$

$$\Leftrightarrow \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] \leq \frac{1}{2} \left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2 \right)$$

$$\Rightarrow \left(\sum_{i=m}^{k} \alpha_i \right) \left(\min_{i \in \{m,\ldots,k\}} \mathbf{E}\left[f(x_i) - f(x_\star)\right] \right) \leq \frac{1}{2} \left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2 \right)$$

$$\Leftrightarrow \min_{i \in \{m,\ldots,k\}} \mathbf{E}\left[f(x_i) - f(x_\star)\right] \leq \frac{\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2}{2 \sum_{i=m}^{k} \alpha_i}$$

Modifying the telescoping sum

- Recall that $a_k \geq 0, b_k \geq 0$, we have $(\min_k a_k) \sum_k b_k \leq \sum_k a_k b_k$
- Also $\mathbf{E}\left[\min_i X_i\right] \leq \min_i \mathbf{E}\left[X_i\right]$
- Now

$$\mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] - \mathbf{E}\left[\|x_m - x_\star\|^2\right] \leq -2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Leftrightarrow 0 \leq \mathbf{E}\left[\|x_{k+1} - x_\star\|^2\right] \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=m}^{k} \alpha_i^2$$

$$\Rightarrow 0 \leq \mathbf{E}\left[\|x_m - x_\star\|^2\right] - 2 \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] + G^2 \sum_{i=1}^{m} \alpha_i^2$$

$$\Leftrightarrow \sum_{i=m}^{k} \alpha_i \mathbf{E}\left[f(x_i) - f(x_\star)\right] \leq \frac{1}{2}\left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2\right)$$

$$\Rightarrow \left(\sum_{i=m}^{k} \alpha_i\right)\left(\min_{i \in \{m,\ldots,k\}} \mathbf{E}\left[f(x_i) - f(x_\star)\right]\right) \leq \frac{1}{2}\left(\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2\right)$$

$$\Leftrightarrow \min_{i \in \{m,\ldots,k\}} \mathbf{E}\left[f(x_i) - f(x_\star)\right] \leq \frac{\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2}{2 \sum_{i=m}^{k} \alpha_i}$$

$$\therefore \mathbf{E}\left[\min_{i \in \{m,\ldots,k\}} \{f(x_i) - f(x_\star)\}\right] \leq \frac{\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2 \sum_{i=m}^{k} \alpha_i^2}{2 \sum_{i=m}^{k} \alpha_i}$$

Showing convergence

- We have shown that

$$\mathbf{E}\left[\min_{i\in\{m,\ldots,k\}}\{f(x_i)-f(x_\star)\}\right] \leq \frac{\mathbf{E}\left[\|x_m - x_\star\|^2\right] + G^2\sum_{i=m}^k \alpha_i^2}{2\sum_{i=m}^k \alpha_i}.$$

Showing convergence

- We have shown that

$$\mathbf{E}\left[\min_{i\in\{m,\ldots,k\}}\{f(x_i)-f(x_\star)\}\right] \leq \frac{\mathbf{E}\left[\|x_m-x_\star\|^2\right]+G^2\sum_{i=m}^k \alpha_i^2}{2\sum_{i=m}^k \alpha_i}.$$

- In the last inequality, m is arbitrary, so set $m \leftarrow 0$, which leads to:

$$0 \leq \mathbf{E}\left[\min_{i\in\{0,\ldots,k\}}\{f(x_i)-f(x_\star)\}\right] \leq \frac{\|x_0-x_\star\|^2+G^2\sum_{i=0}^k \alpha_i^2}{2\sum_{i=0}^k \alpha_i}.$$

Showing convergence

- We have shown that

$$\mathbf{E}\left[\min_{i\in\{m,\dots,k\}}\{f(x_i)-f(x_\star)\}\right] \le \frac{\mathbf{E}\left[\|x_m-x_\star\|^2\right]+G^2\sum_{i=m}^k \alpha_i^2}{2\sum_{i=m}^k \alpha_i}.$$

- In the last inequality, m is arbitrary, so set $m \leftarrow 0$, which leads to:

$$0 \le \mathbf{E}\left[\min_{i\in\{0,\dots,k\}}\{f(x_i)-f(x_\star)\}\right] \le \frac{\|x_0-x_\star\|^2+G^2\sum_{i=0}^k \alpha_i^2}{2\sum_{i=0}^k \alpha_i}.$$

- if we have $\sum_{i=0}^k \alpha_i^2 < \infty$ and $\sum_{i=0}^k \alpha_i = \infty$, then we have

$$\mathbf{E}\left[\min_{i\in\{0,\dots,k\}} f(x_i)\right] \to f(x_\star).$$

Convergence rate

- Additional assumption required:
- C is bounded (besides closed and convex), for all $x \in C$, we have $\|x\| \leq B$
- Set $\alpha_k = \frac{\alpha}{\sqrt{k+1}}$
- Then we can show that $\mathbf{E}\left[f(\bar{x}_k) - f(x_\star)\right] \leq \left(\frac{3B^2}{\alpha} + \alpha G^2\right)\frac{1}{\sqrt{k}}$, where $\bar{x}_k = \frac{1}{k}\sum_{i=0}^{k-1} x_k$
- To the best of my knowledge, no proof that establishes a rate on $\mathbf{E}\left[f(x_k) - f(x_\star)\right]$

Outline

HW4 and project

Stochastic gradient descent for nonsmooth convex setup

Minibatch SGD and momentum SGD

Minibatch SGD

- Problem

$$p^\star = \left(\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \right) \qquad (\mathcal{P})$$

- For $B \subset \{1, \ldots, n\}$, define $f'_B(x_t) = \frac{1}{|B|} \sum_{i \in B} f'_i(x_t)$
- Minibatch SGD works as follows
  - Pick some $B_k \subset \{1, \ldots, n\}$ sampled uniformly among sets of size $b \in \{1, 2, \ldots, n\}$
  - Update $x_{k+1} = x_k - \alpha_k f'_{B_k}(x_k)$
- If we run N iterations, then convergence rate in *averaged* function value gap is $\mathcal{O}(1/\sqrt{N})$ for *smooth* convex f_i
- No convergence rate for nonsmooth convex function

Stochastic momentum method

- Problem

$$p^\star = \left(\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \right) \tag{$\mathcal{P}$}$$

- For most, if not all, deep learning solvers is some form of SGD with momentum
- Stochastic momentum method is as follows
- Pick some $i_k \in \{1, \ldots, n\}$ sampled uniformly with probability $1/n$
- Update $x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k) + \beta_k (x_k - x_{k-1})$
- If we run N iterations, then convergence rate in *last iterate* function value gap is $\mathcal{O}(1/\sqrt{N})$ for *smooth* convex f_i