Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 4: The role of computation

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2022)

lions@epfl swisscom E4S Enterprise for Society EPFL

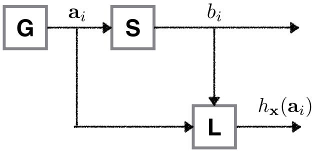
License Information for Mathematics of Data Slides

Outline

- This lecture

  1. Principles of iterative descent methods
  2. Gradient descent for smooth convex problems
  3. Gradient descent for smooth non-convex problems

Recall: Learning machines result in optimization problems



$$(\mathbf{a}_i, b_i)_{i=1}^n \xrightarrow[\text{parameter } \mathbf{x}]{\text{modeling}} P(b_i|\mathbf{a}_i, \mathbf{x}) \xrightarrow[\text{identical dist.}]{\text{independency}} \mathsf{p}_{\mathbf{x}}(\mathbf{b}) := \prod_{i=1}^{n} P(b_i|\mathbf{a}_i, \mathbf{x})$$

Definition (Maximum-likelihood estimator)

The maximum-likelihood (ML) estimator is given by

$$\mathbf{x}^\star_{\mathsf{ML}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ L(h_{\mathbf{x}}(\mathbf{a}), \mathbf{b}) := -\log \mathsf{p}_{\mathbf{x}}(\mathbf{b}) \right\},$$

where $\mathsf{p}_{\mathbf{x}}(\cdot)$ denotes the probability density function or probability mass function of $\mathbb{P}_{\mathbf{x}}$, for $\mathbf{x} \in \mathcal{X}$.

M-Estimators

Roughly speaking, estimators can be formulated as optimization problems of the following form:

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} \{F(\mathbf{x})\},$$

with some constraints $\mathcal{X} \subseteq \mathbb{R}^p$. The term "$M$-estimator" denotes "maximum-likelihood-type estimator" [2].

Unconstrained minimization

Problem (Mathematical formulation)

How can we find an optimal solution to the following optimization problem?

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{F(\mathbf{x}) := f(\mathbf{x})\right\}} \tag{1}$$

Note that (1) *is unconstrained.*

Definition (Optimal solutions and solution set)

1. $\mathbf{x}^\star \in \mathbb{R}^p$ is a solution to (1) if $\boxed{F(\mathbf{x}^\star) = F^\star}$.
2. $\boxed{\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathbb{R}^p \ : \ F(\mathbf{x}^\star) = F^\star\}}$ is the solution set of (1).
3. (1) has solution if $\mathcal{S}^\star$ is non-empty.

Approximate vs. exact optimality

Is it possible to solve an optimization problem?

"In general, optimization problems are **unsolvable***"* - Y. Nesterov [4]

Observations:
- ∘ Even when a closed-form solution exists, numerical accuracy may still be an issue.
- ∘ We must be content with **approximately** optimal solutions.

Definition

We say that $\mathbf{x}_\epsilon^\star$ is ϵ-optimal in **objective value** if

$$f(\mathbf{x}_\epsilon^\star) - f^\star \leq \epsilon \ .$$

Definition

We say that $\mathbf{x}_\epsilon^\star$ is ϵ-optimal in **sequence** if, for some norm $\|\cdot\|$,

$$\|\mathbf{x}_\epsilon^\star - \mathbf{x}^\star\| \leq \epsilon.$$

Remark: ∘ The latter approximation guarantee is considered stronger.

A basic *iterative* strategy

General idea of an optimization algorithm

Guess a solution, and then *refine* it based on *oracle information*.

Repeat the procedure until the result is *good enough*.

Basic principles of descent methods

Template for iterative descent methods

1. Let $\mathbf{x}^0 \in \mathrm{dom}(f)$ be a starting point.
2. Generate a sequence of vectors $\mathbf{x}^1, \mathbf{x}^2, \dots \in \mathrm{dom}(f)$ so that we have descent:

$$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k), \quad \text{for all } k = 0, 1, \dots$$

   until $\mathbf{x}^k$ is ϵ-optimal.

Such a sequence $\left\{\mathbf{x}^k\right\}_{k \geq 0}$ can be generated as:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{p}^k$$

where $\mathbf{p}^k$ is a descent direction and $\alpha_k > 0$ a step-size.

Remarks:

- Iterative algorithms can use various **oracle** information in the optimization problem
- The type of oracle information used becomes a defining characteristic of the algorithm
- Example oracles: Objective value, gradient, and Hessian result in 0-th, 1-st, 2-nd order methods
- The oracle choices determine α_k and $\mathbf{p}^k$ as well as the overall convergence rate and complexity

Basic principles of descent methods

A condition for local descent directions

The iterates are given as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{p}^k.$$

For a differentiable f, we have by Taylor's theorem

$$f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k) + \alpha_k \langle \nabla f(\mathbf{x}^k),\ \mathbf{p}^k \rangle + \mathcal{O}(\alpha_k^2 \|\mathbf{p}\|_2^2).$$

For α_k small enough, the term $\alpha_k \langle \nabla f(\mathbf{x}^k),\ \mathbf{p}^k \rangle$ dominates $\mathcal{O}(\alpha_k^2)$ for a fixed $\mathbf{p}^k$.

Therefore, in order to have $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$, we require

$$\langle \nabla f(\mathbf{x}^k),\ \mathbf{p}^k \rangle < 0$$

Basic principles of descent methods

Local steepest descent direction

Since

$$\langle \nabla f(\mathbf{x}^k),\ \mathbf{p}^k \rangle = \|\nabla f(\mathbf{x}^k)\| \|\mathbf{p}^k\| \cos\theta\ ,$$

where θ is the angle between $\nabla f(\mathbf{x}^k)$ and $\mathbf{p}^k$, we have

$$\mathbf{p}^k := -\nabla f(\mathbf{x}^k)$$

as the local *steepest descent* direction.

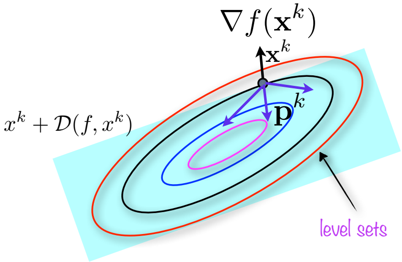
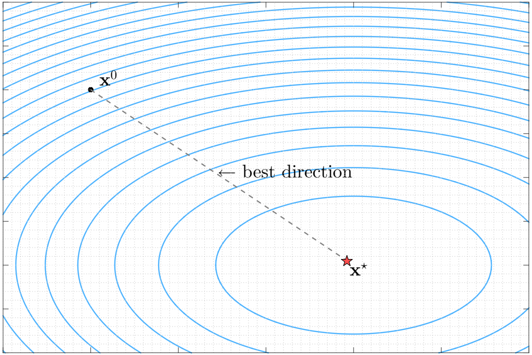


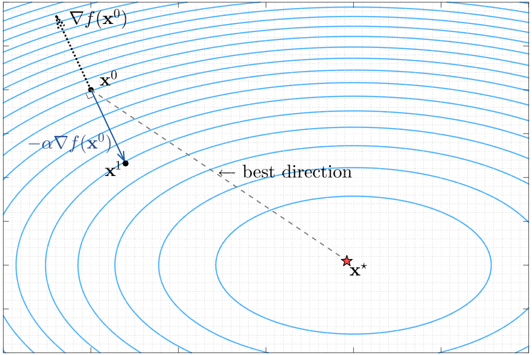
Figure: Descent directions in 2D should be an element of the cone of descent directions $\mathcal{D}(f, \cdot)$.

A simple iterative algorithm: Gradient descent



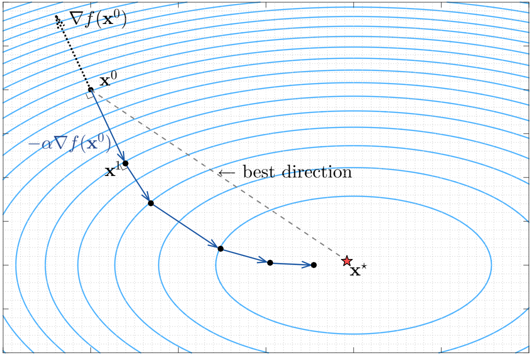
1. Choose initial point: x^0.

A simple iterative algorithm: Gradient descent



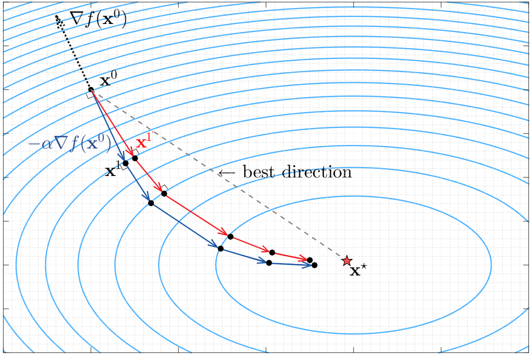
1. Choose initial point: x^0.
2. Take a step in the negative gradient direction with a step size $\alpha > 0$: $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha\nabla f(\mathbf{x}^k)$.

A simple iterative algorithm: Gradient descent



1. Choose initial point: x^0.
2. Take a step in the negative gradient direction with a step size $\alpha > 0$: $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha\nabla f(\mathbf{x}^k)$.
3. Repeat this procedure until x^k is accurate enough.

A simple iterative algorithm: Proximal-point method [5]



1. Choose initial point: x^0.
2. Take a step in the negative gradient direction with a step size $\alpha > 0$: $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha\nabla f(\mathbf{x}^{k+1})$.
3. Repeat this procedure until x^k is accurate enough.

Recall the statistical estimation context

Observations:

- Denote $\mathbf{x}^\natural$ is the unknown true parameter
- The estimator $\mathbf{x}^\star$'s performance, e.g., $\|\,\mathbf{x}^\star - \mathbf{x}^\natural\,\|_2^2$ depends on the data size n.
- Evaluating $\|\,\mathbf{x}^\star - \mathbf{x}^\natural\,\|_2^2$ is not enough for evaluating the performance of a Learning Machine
  - We can only *numerically approximate* the solution of

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{F(\mathbf{x})\right\}.$$

- We use algorithms to *numerically approximate* $\mathbf{x}^\star$.

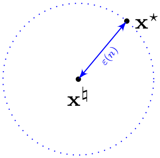
Practical performance

Denote the numerical approximation by an algorithm at time t by $\mathbf{x}^t$.

The practical performance at time t using n data samples is determined by

$$\underbrace{\|\,\mathbf{x}^t - \mathbf{x}^\natural\,\|_2}_{\bar{\varepsilon}(t,n)} \leq \underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_2}_{\epsilon(t)} + \underbrace{\|\,\mathbf{x}^\star - \mathbf{x}^\natural\,\|_2}_{\varepsilon(n)},$$

where $\varepsilon(n)$ denotes the statistical error, $\epsilon(t)$ is the numerical error, and $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine.



Recall the statistical estimation context

Observations:

- Denote $\mathbf{x}^\natural$ is the unknown true parameter
- The estimator $\mathbf{x}^\star$'s performance, e.g., $\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$ depends on the data size n.
- Evaluating $\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$ is not enough for evaluating the performance of a Learning Machine
  - We can only *numerically approximate* the solution of
$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x})\}.$$
- We use algorithms to *numerically approximate* $\mathbf{x}^\star$.

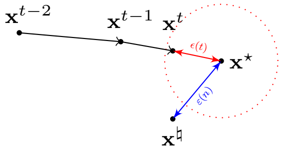
Practical performance

Denote the numerical approximation by an algorithm at time t by $\mathbf{x}^t$.

The practical performance at time t using n data samples is determined by

$$\underbrace{\| \mathbf{x}^t - \mathbf{x}^\natural \|_2}_{\bar{\varepsilon}(t,n)} \leq \underbrace{\left\| \mathbf{x}^t - \mathbf{x}^\star \right\|_2}_{\epsilon(t)} + \underbrace{\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2}_{\varepsilon(n)},$$

where $\varepsilon(n)$ denotes the statistical error, $\epsilon(t)$ is the numerical error, and $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine.



Recall the statistical estimation context

Observations:

- Denote $\mathbf{x}^\natural$ is the unknown true parameter
- The estimator $\mathbf{x}^\star$'s performance, e.g., $\|\mathbf{x}^\star - \mathbf{x}^\natural\|_2^2$ depends on the data size n.
- Evaluating $\|\mathbf{x}^\star - \mathbf{x}^\natural\|_2^2$ is not enough for evaluating the performance of a Learning Machine
  - ▶ We can only *numerically approximate* the solution of
$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x})\}.$$
- We use algorithms to *numerically approximate* $\mathbf{x}^\star$.

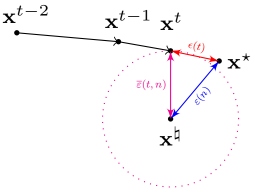
Practical performance

Denote the numerical approximation by an algorithm at time t by $\mathbf{x}^t$.

The practical performance at time t using n data samples is determined by

$$\underbrace{\|\mathbf{x}^t - \mathbf{x}^\natural\|_2}_{\bar{\varepsilon}(t,n)} \leq \underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_2}_{\epsilon(t)} + \underbrace{\|\mathbf{x}^\star - \mathbf{x}^\natural\|_2}_{\varepsilon(n)},$$

where $\varepsilon(n)$ denotes the statistical error, $\epsilon(t)$ is the numerical error, and $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine.

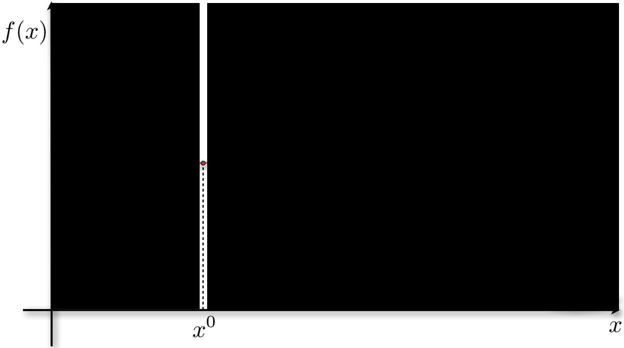


Challenges for an iterative optimization algorithm

Problem

Find the minimum $x^\star$ of $f(x)$, given starting point x^0 based on only local information.

- **Fog of war**

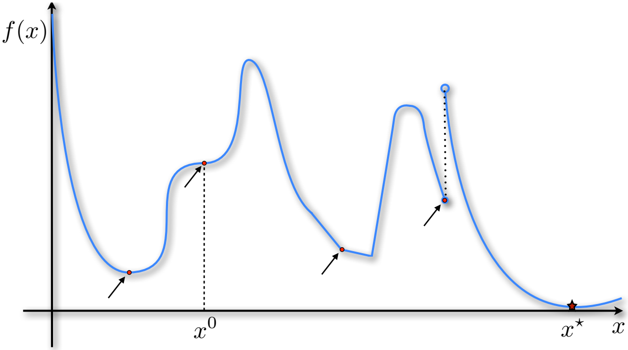


Challenges for an iterative optimization algorithm

Problem

Find the minimum $x^\star$ of $f(x)$, given starting point x^0 based on only local information.

- Fog of war, non-differentiability, discontinuities, local minima, stationary points...



A notion of convergence: Stationarity

∘ Let $f : \mathbb{R}^p \to \mathbb{R}$ be twice-differentiable and $\mathbf{x}^\star = \min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$.

Gradient method

Choose a starting point $\mathbf{x}^0$ and iterate

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha\nabla f(\mathbf{x}^k)$$

where $\alpha > 0$ is a step-size to be chosen so that $\mathbf{x}^k$ converges to $\mathbf{x}^\star$.

Definition (First order stationary point (FOSP))

A point $\bar{\mathbf{x}}$ is a first order stationary point of a twice differentiable function f if

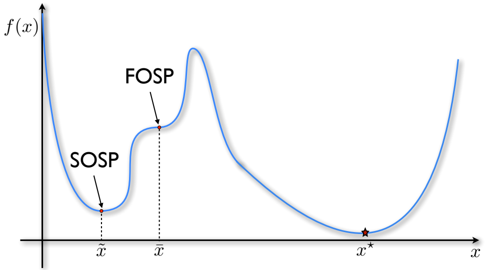
$$\nabla f(\bar{\mathbf{x}}) = \mathbf{0}.$$

Fixed-point characterization

Multiply by -1 and add $\bar{\mathbf{x}}$ to both sides to obtain the fixed point condition:

$$\bar{\mathbf{x}} = \bar{\mathbf{x}} - \alpha\nabla f(\bar{\mathbf{x}}) \qquad \text{for all } \alpha \in \mathbb{R}.$$

Geometric interpretation of stationarity



Observation: ◦ Neither $\bar{\mathbf{x}}$, nor $\tilde{\mathbf{x}}$ is **necessarily** equal to $\mathbf{x}^\star$!!

Proposition (⋆Local minima, maxima, and saddle points)

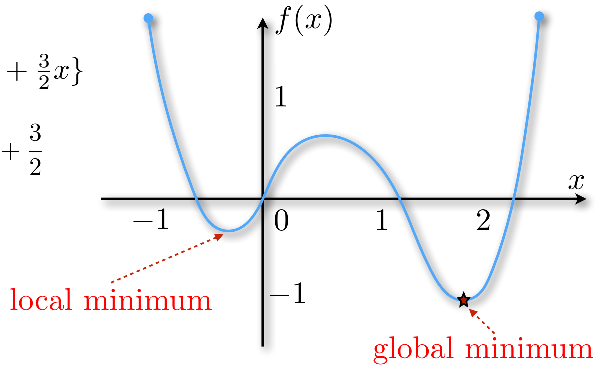
Let $\bar{\mathbf{x}}$ be a stationary point of a twice differentiable function f.

1. *If $\nabla^2 f(\bar{\mathbf{x}}) \succ 0$, then the point $\bar{\mathbf{x}}$ is called a local minimum or a second order stationary point (SOSP).*
2. *If $\nabla^2 f(\bar{\mathbf{x}}) \prec 0$, then the point $\bar{\mathbf{x}}$ is called a local maximum.*
3. *If $\nabla^2 f(\bar{\mathbf{x}}) = 0$, then the point $\bar{\mathbf{x}}$ can be a saddle point, a local minimum, or a local maximum.*

Local minima

$$\min_{x \in \mathbb{R}} \{x^4 - 3x^3 + x^2 + \tfrac{3}{2}x\}$$

$$\frac{df}{dx} = 4x^3 - 9x^2 + 2x + \frac{3}{2}$$



Choose $x^0 = 0$ and $\alpha = \frac{1}{6}$

$x^1 = x^0 - \alpha \frac{df}{dx}\Big|_{x=x^0} = 0 - \frac{1}{6}\frac{3}{2} = -\frac{1}{4}$

$x^2 = -\frac{5}{16}$

...

x^k converges to a **local minimum**!

From local to global optimality

Definition (Local minimum)

Given $f : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, a vector $\mathbf{x}^\star \in \mathbb{R}^p$ is called a *local minimum* of f if there exists $\epsilon > 0$ s.t.

$$f(\mathbf{x}^\star) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^p \quad \text{with} \quad \|\mathbf{x} - \mathbf{x}^\star\| \leq \epsilon.$$

Theorem

If $\mathcal{Q} \subset \mathbb{R}^p$ is a convex set and $f : \mathbb{R}^p \to (-\infty, +\infty]$ is a proper convex function, then a local minimum of f over $\mathcal{Q}$ is also a global minimum of f over $\mathcal{Q}$.

Proof.

Suppose $\mathbf{x}^\star$ is a local minimum but not global, i.e. there exist $\mathbf{x} \in \mathbb{R}^p$ s.t. $f(\mathbf{x}) < f(\mathbf{x}^\star)$. By convexity,

$$f(\alpha\mathbf{x}^\star + (1-\alpha)\mathbf{x}) \leq \alpha f(\mathbf{x}^\star) + (1-\alpha)f(\mathbf{x}) < f(\mathbf{x}^\star), \forall \alpha \in [0, 1]$$

which contradicts the local minimality of $\mathbf{x}^\star$. □

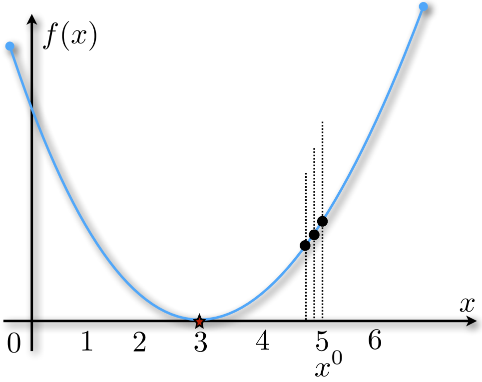
Theorem

Let $f : \mathbb{R}^p \to \mathbb{R}$ be a convex differentiable function. Then any stationary point of f is a global minimum.

Effect of very small step-size α...

$$\min_{x \in \mathbb{R}} \frac{1}{2}(x-3)^2$$

$$\frac{df}{dx} = x - 3$$



Choose $x^0 = 5$ and $\boldsymbol{\alpha = \frac{1}{10}}$

$x^1 = x^0 - \alpha \frac{df}{dx}\Big|_{x=x^0} = 5 - \frac{1}{10}2 = 4.8$

$x^2 = x^1 - \alpha \frac{df}{dx}\Big|_{x=x^1} = 4.8 - \frac{1}{10}1.8 = 4.62$

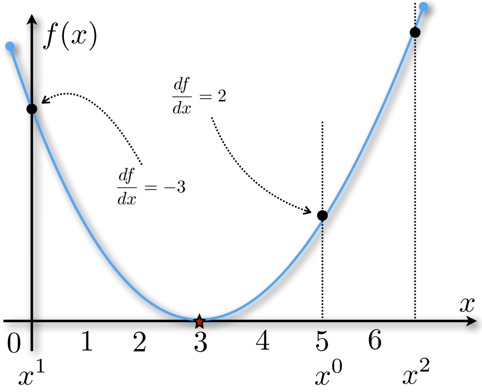
...

x^k converges **very slowly**.

Effect of very large step-size α...

$$\min_{x\in\mathbb{R}} \frac{1}{2}(x-3)^2$$

$$\frac{df}{dx} = x - 3$$



Choose $x^0 = 5$ and $\boldsymbol{\alpha = \frac{5}{2}}$

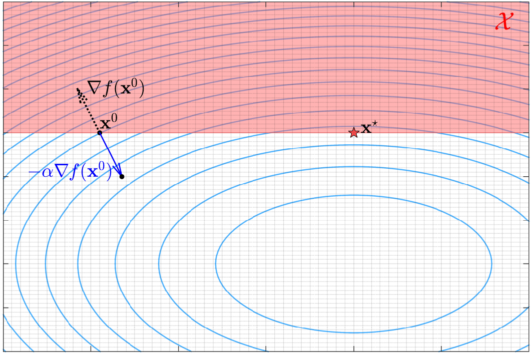
$x^1 = x^0 - \alpha \frac{df}{dx}\Big|_{x=x^0} = 5 - \frac{5}{2}2 = 0$

$x^2 = x^1 - \alpha \frac{df}{dx}\Big|_{x=x^1} = 0 - \frac{5}{2}(-3) = \frac{15}{2}$

$\ldots$

x^k **diverges**.

Discontinuities

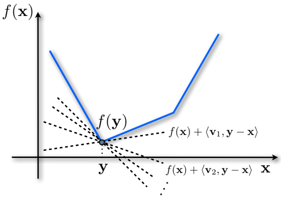


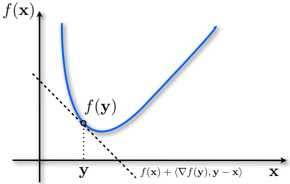
In many practical problems,

we need to **minimize** the cost **under some constraints**.

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \right\}$$

Nonsmooth functions





Definition (Subdifferential)

The subdifferential of f at x, denoted $\partial f(x)$, is the set of all vectors v satisfying

$$f(y) \geq f(x) + \langle v, y - x \rangle + o(\|y - x\|) \quad \text{as } y \to x$$

If the function f is differentiable, then its subdifferential contains only the gradient.

Subgradient method

Choose a starting point $\mathbf{x}^0$, receive a subgradient from the (set of) subdifferential, and iterate

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \partial f(\mathbf{x}^k)$$

where $\alpha_k > 0$ is a step-size procedure to be chosen so that $\mathbf{x}^k$ converges to a stationary point.

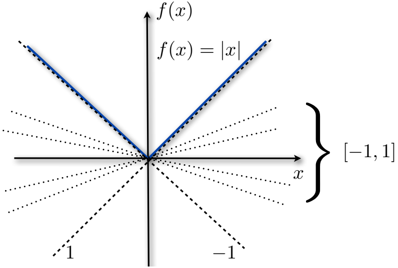
Subdifferentials and (sub)gradients

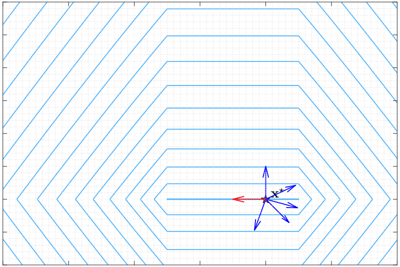
Subgradient method

Choose a starting point $\mathbf{x}^0$, receive a subgradient from the (set of) subdifferential, and iterate

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \partial f(\mathbf{x}^k)$$

where $\alpha_k > 0$ is a step-size procedure to be chosen so that $\mathbf{x}^k$ converges to a stationary point.





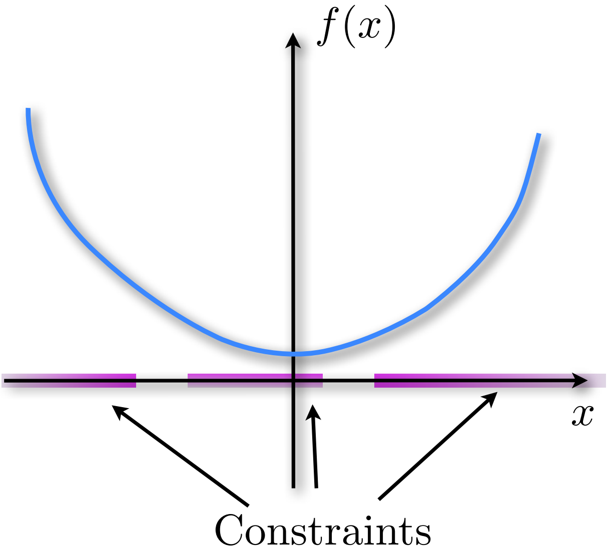
Example

$\partial|x| = \{\text{sgn}(x)\}$, if $x \neq 0$, but $[-1, 1]$, if $x = 0$.

Remark:

The step-size α_k often needs to decrease with k.

Is convexity of f enough for an iterative optimization algorithm?



Smooth unconstrained convex minimization

Problem (Mathematical formulation)

The unconstrained convex minimization problem is defined as:

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$$

1. f *is a convex function that is*
   - *proper* : $\forall \mathbf{x} \in \mathbb{R}^p$, $-\infty < f(\mathbf{x})$ *and there exists* $\mathbf{x} \in \mathbb{R}^p$ *such that* $f(\mathbf{x}) < +\infty$.
   - *closed* : *The epigraph* epi$f = \{(\mathbf{x}, t) \in \mathbb{R}^{p+1}, f(\mathbf{x}) \leq t\}$ *is closed.*
   - *smooth* : f *is differentiable and its gradient* ∇f *is* L*-Lipschitz.*
2. *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}\,(f) : f(\mathbf{x}^\star) = f^\star\}$ *is nonempty.*

Example: Maximum likelihood estimation and M-estimators

Problem

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ *be unknown and* $b_1, ..., b_n$ *be i.i.d. samples of a random variable* B *with p.d.f.* $p_{\mathbf{x}^\natural}(b) \in \mathcal{P} := \{p_{\mathbf{x}}(b) : \mathbf{x} \in \mathbb{R}^p\}$. ***Goal:*** *Estimate* $\mathbf{x}^\natural$ *from* $b_1, \ldots, b_n$.

Optimization formulation (ML estimator)

$$\mathbf{x}^\star_{\text{ML}} := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \ln\left[\mathsf{p}_{\mathbf{x}}(b_i)\right] \right\} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$$

Theorem (Performance of the ML estimator [3, 6])

The random variable $\hat{\mathbf{x}}_{ML}$ *satisfies*

$$\lim_{n\to\infty} \sqrt{n}\,\mathbf{J}^{-1/2}\left(\hat{\mathbf{x}}_{ML} - \mathbf{x}^\natural\right) \stackrel{d}{=} Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $\mathbf{J} := -\mathbb{E}\left[\nabla^2_{\mathbf{x}} \ln\left[p_{\mathbf{x}}(B)\right]\right]\Big|_{\mathbf{x}=\mathbf{x}^\natural}$ *is the Fisher information matrix associated with one sample. Roughly speaking,*

$$\| \sqrt{n}\,\mathbf{J}^{-1/2}\left(\hat{\mathbf{x}}_{ML} - \mathbf{x}^\natural\right) \|_2^2 \sim \mathrm{Tr}\left(\mathbf{I}\right) = p \quad \Rightarrow \quad \boxed{\| \hat{\mathbf{x}}_{ML} - \mathbf{x}^\natural \|_2^2 = \mathcal{O}(p/n)}.$$

Gradient descent methods

Definition

Gradient descent (GD) Starting from $\mathbf{x}^0 \in \mathrm{dom}(f)$, update $\{\mathbf{x}^k\}_{k\geq 0}$ as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k) = \mathbf{x}^k + \alpha_k \mathbf{p}^k.$$

Notice that $\mathbf{p}^k := -\nabla f(\mathbf{x}^k)$ is the steepest descent (anti-gradient) search direction.

Key question: how to choose α_k to have descent/contraction?

Gradient descent methods

Definition

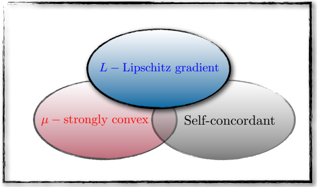
Gradient descent (GD) Starting from $\mathbf{x}^0 \in \mathrm{dom}(f)$, update $\{\mathbf{x}^k\}_{k\geq 0}$ as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k) = \mathbf{x}^k + \alpha_k \mathbf{p}^k.$$

Notice that $\mathbf{p}^k := -\nabla f(\mathbf{x}^k)$ is the steepest descent (anti-gradient) search direction.

Key question: how to choose α_k to have descent/contraction?

Next few slides: structural assumptions



L-smooth, μ-strongly convex functions

Definition (Recall Recitation 2)

Let $f : \mathcal{Q} \to \mathbb{R}, \mathcal{Q} \subseteq \mathbb{R}^p$ be a continuously differentiable function. Then, f μ-strongly convex if for any $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|_2^2.$$

The function f is L-smooth if for any $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|_2^2.$$

If f is twice differentiable, an equivalent characterization of f being L-smooth and μ-strongly convex is

$$\mu \mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L \mathbf{I}.$$

L-smooth, μ-strongly convex functions

Definition (Recall Recitation 2)

Let $f : \mathcal{Q} \to \mathbb{R}, \mathcal{Q} \subseteq \mathbb{R}^p$ be a continuously differentiable function. Then, f μ-strongly convex if for any $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|_2^2.$$

The function f is L-smooth if for any $\mathbf{x}, \mathbf{y} \in \mathcal{Q}$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|_2^2.$$

If f is twice differentiable, an equivalent characterization of f being L-smooth and μ-strongly convex is

$$\mu \mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbf{I}.$$

Observations:
- Both μ and L show up in convergence rate characterization of algorithms
- **Unfortunately, μ, L are usually not known a priori...**
- When they are known, they can help significantly (even in stopping algorithms)

Example: Least-squares estimation

Problem

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ *and* $\mathbf{A} \in \mathbb{R}^{n\times p}$ *(full column rank). Goal: estimate* $\mathbf{x}^\natural$*, given* $\mathbf{A}$ *and*

$$\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w},$$

where $\mathbf{w}$ *denotes unknown noise.*

Optimization formulation (Least-squares estimator)

$$\min_{\mathbf{x}\in\mathbb{R}^p} \underbrace{\frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2}_{f(\mathbf{x})}.$$

Structural properties

1. $\nabla f(\mathbf{x}) = \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$, and $\nabla^2 f(\mathbf{x}) = \mathbf{A}^T\mathbf{A}$.
2. $\lambda_p \mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq \lambda_1 \mathbf{I}$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ are the eigenvalues of $\mathbf{A}^T\mathbf{A}$.
3. It follows that $L = \lambda_1$ and $\mu = \lambda_p$. If $\lambda_p > 0$, then f is L-smooth and μ-strongly convex, otherwise f is just L-smooth.
4. Since $\mathrm{rank}(\mathbf{A}^T\mathbf{A}) \leq \min\{n, p\}$, if $n < p$, then $\lambda_p = 0$.

Back to gradient descent methods

Gradient descent (GD) algorithm

Starting from $\mathbf{x}^0 \in \mathrm{dom}(f)$, produce the sequence $\mathbf{x}^1, ..., \mathbf{x}^k, ...$ according to

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k) = \mathbf{x}^k + \alpha_k \mathbf{p}^k.$$

Notice that $\mathbf{p}^k := -\nabla f(\mathbf{x}^k)$ is the steepest descent (anti-gradient) direction.
Key question: how do we choose α_k to have descent/contraction?

Back to gradient descent methods

Gradient descent (GD) algorithm

Starting from $\mathbf{x}^0 \in \mathrm{dom}(f)$, produce the sequence $\mathbf{x}^1, \ldots, \mathbf{x}^k, \ldots$ according to

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k) = \mathbf{x}^k + \alpha_k \mathbf{p}^k.$$

Notice that $\mathbf{p}^k := -\nabla f(\mathbf{x}^k)$ is the steepest descent (anti-gradient) direction.
Key question: how do we choose α_k to have descent/contraction?

Step-size selection

Case 1: If f is L-smooth, then:

1. We can choose $0 < \alpha_k < \frac{2}{L}$. The optimal choice is $\alpha_k := \frac{1}{L}$.
2. α_k can be determined by a line-search procedure:
   2.1 **Exact line search**: $\alpha_k := \arg\min_{\alpha > 0} f(\mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k))$.
   2.2 **Back-tracking line search** with Armijo-Goldstein's condition:

$$f(\mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)) \leq f(\mathbf{x}^k) - c\alpha \|\nabla f(\mathbf{x}^k)\|^2, \quad c \in (0, 1/2].$$

Case 2: If in addition to being L-smooth, f is μ-strongly convex, then:

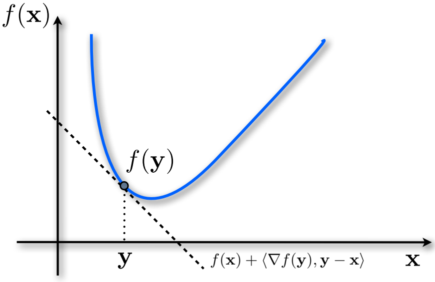
1. We can choose $0 < \alpha_k \leq \frac{2}{L+\mu}$. The optimal choice is $\alpha_k := \frac{2}{L+\mu}$.

Towards a geometric interpretation I

Remarks:

- Let f be L-smooth with gradient $\nabla f(\mathbf{x})$ and Hessian $\nabla^2 f(\mathbf{x})$.
- First-order Taylor approximation of f at $\mathbf{y}$:

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$$



- Convex functions: 1^{st}**-order Taylor approximation is a global lower surrogate.**

An equivalent characterization of smoothness

Lemma

Let f be a continuously differentiable convex function :

$$f \text{ is } L\text{-Lipschitz gradient} \implies f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|_2^2$$

Proof:

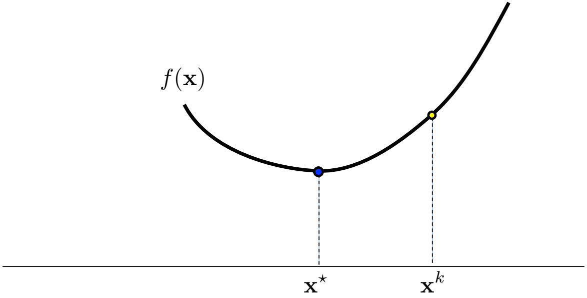
- By Taylor's theorem:

$$f(\mathbf{y}) = f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \int_0^1 \langle \nabla f(\mathbf{x}+\tau(\mathbf{y}-\mathbf{x})) - \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle d\tau.$$

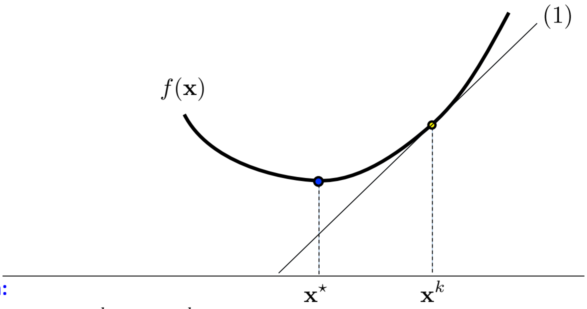
Therefore,

$$\begin{aligned} f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle &\leq \int_0^1 \|\nabla f(\mathbf{x}+\tau(\mathbf{y}-\mathbf{x})) - \nabla f(\mathbf{x})\|^* \cdot \|\mathbf{y}-\mathbf{x}\| d\tau \\ &\leq L\|\mathbf{y}-\mathbf{x}\|_2^2 \int_0^1 \tau d\tau = \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \end{aligned}$$

Gradient descent methods: geometrical intuition



Gradient descent methods: geometrical intuition



Structure in optimization:

(1) $\quad f(\mathbf{x}) \geq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle$

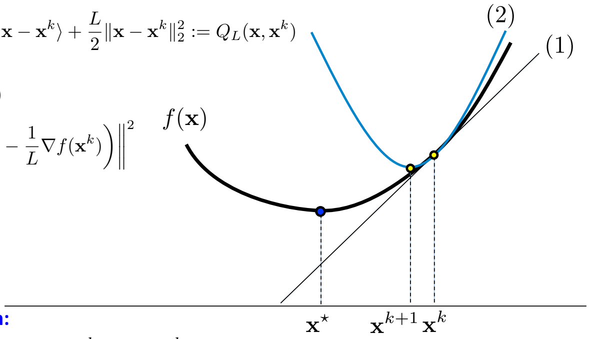
Gradient descent methods: geometrical intuition

Majorize:

$$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 := Q_L(\mathbf{x}, \mathbf{x}^k)$$

Minimize:

$$\begin{aligned}\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} Q_L(\mathbf{x}, \mathbf{x}^k) \\ &= \arg\min_{\mathbf{x}} \left\| \mathbf{x} - \left(\mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k) \right) \right\|^2 \\ &= \mathbf{x}^k - \frac{1}{L}\nabla f(\mathbf{x}^k)\end{aligned}$$



Structure in optimization:

(1) $f(\mathbf{x}) \geq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle$

(2) $f(\mathbf{x}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2$

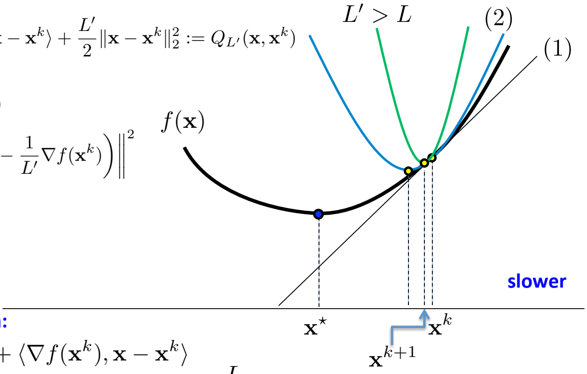
Gradient descent methods: geometrical intuition

Majorize:

$$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L'}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 := Q_{L'}(\mathbf{x}, \mathbf{x}^k)$$

Minimize:

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} Q_{L'}(\mathbf{x}, \mathbf{x}^k) \\ &= \arg\min_{\mathbf{x}} \left\| \mathbf{x} - \left(\mathbf{x}^k - \frac{1}{L'}\nabla f(\mathbf{x}^k) \right) \right\|^2 \\ &= \mathbf{x}^k - \frac{1}{L'}\nabla f(\mathbf{x}^k) \end{aligned}$$



Structure in optimization:

(1) $\quad f(\mathbf{x}) \geq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle$

(2) $\quad f(\mathbf{x}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2$

Convergence rate of gradient descent

Theorem

Let f be a twice-differentiable convex function, if

$$f \text{ is } L\text{-smooth}, \qquad \alpha = \frac{1}{L}: \quad f(\mathbf{x}^k) - f(\mathbf{x}^\star) \leq \frac{2L}{k+4} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2^2$$

$$f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \qquad \alpha = \frac{2}{L+\mu}: \quad \|\mathbf{x}^k - \mathbf{x}^\star\|_2 \leq \left(\frac{L-\mu}{L+\mu}\right)^k \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$$

$$f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \qquad \alpha = \frac{1}{L}: \quad \|\mathbf{x}^k - \mathbf{x}^\star\|_2 \leq \left(\frac{L-\mu}{L+\mu}\right)^{\frac{k}{2}} \|\mathbf{x}^0 - \mathbf{x}^\star\|_2$$

Note that $\frac{L-\mu}{L+\mu} = \frac{\kappa-1}{\kappa+1}$, where $\kappa := \frac{L}{\mu}$ is the condition number of $\nabla^2 f$.

Convergence rate of gradient descent

Theorem

Let f be a twice-differentiable convex function, if

f is L-smooth,	$\alpha = \frac{1}{L}$:	$f(\mathbf{x}^k) - f(\mathbf{x}^\star)$	$\leq$	$\frac{2L}{k+4}$	$\lVert\mathbf{x}^0 - \mathbf{x}^\star\rVert_2^2$
f is L-smooth and μ-strongly convex,	$\alpha = \frac{2}{L+\mu}$:	$\lVert\mathbf{x}^k - \mathbf{x}^\star\rVert_2$	$\leq$	$\left(\frac{L-\mu}{L+\mu}\right)^k$	$\lVert\mathbf{x}^0 - \mathbf{x}^\star\rVert_2$
f is L-smooth and μ-strongly convex,	$\alpha = \frac{1}{L}$:	$\lVert\mathbf{x}^k - \mathbf{x}^\star\rVert_2$	$\leq$	$\left(\frac{L-\mu}{L+\mu}\right)^{\frac{k}{2}}$	$\lVert\mathbf{x}^0 - \mathbf{x}^\star\rVert_2$

Note that $\frac{L-\mu}{L+\mu} = \frac{\kappa-1}{\kappa+1}$, where $\kappa := \frac{L}{\mu}$ is the condition number of $\nabla^2 f$.

- **Assumption:** Lipschitz gradient. **Result:** convergence rate in **objective values**.
- **Assumption:** Strong convexity. **Result:** convergence rate in **sequence** of the iterates and in **objective values**.

Remark:
- Note that the suboptimal step-size choice $\alpha = \frac{1}{L}$ adapts to the strongly convex case
- That is, it features a linear rate vs. the standard sublinear rate.

Example: Ridge regression

Optimization formulation

- Let $\mathbf{A} \in \mathbb{R}^{n \times p}$ and $\mathbf{b} \in \mathbb{R}^n$ given by $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{w} \in \mathbb{R}^n$ is some noise.
- A classical estimator of $\mathbf{x}^\natural$, known as ridge regression, is

$$\min_{\mathbf{x} \in \mathbb{R}^p} f(\mathbf{x}) := \frac{1}{2} \| \mathbf{b} - \mathbf{A}\mathbf{x} \|_2^2 + \frac{\rho}{2} \|\mathbf{x}\|_2^2.$$

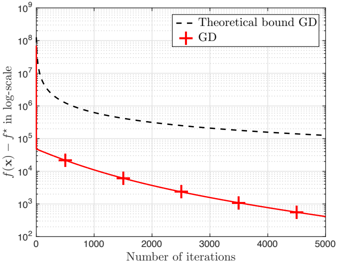
where $\rho \geq 0$ is a regularization parameter

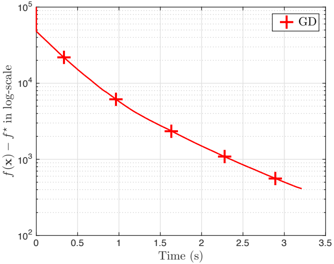
Remarks:

- f is L-smooth and μ-strongly convex with:
  1. $L = \lambda_1(\mathbf{A}^T\mathbf{A}) + \rho$;
  2. $\mu = \lambda_p(\mathbf{A}^T\mathbf{A}) + \rho$;
  3. where $\lambda_1 \geq \ldots \geq \lambda_p$ are the eigenvalues of $\mathbf{A}^T\mathbf{A}$.
- The ratio $\kappa = \frac{L}{\mu}$ decreases as ρ increases, leading to faster linear convergence.
- Note that if $n < p$ and $\rho = 0$, we have $\mu = 0$, hence f is only L-smooth.
- We can expect only $\mathcal{O}(1/k)$ convergence from the gradient descent method.

Example: Ridge regression

<u>**Case 1:**</u> $n = 500, p = 2000, \rho = 0$

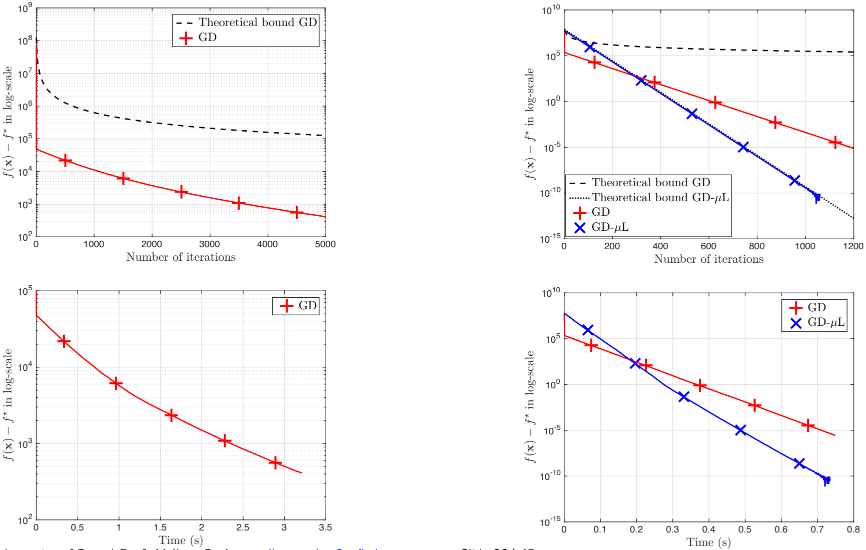




Example: Ridge regression

Case 1: $n = 500, p = 2000, \rho = 0$

Case 2: $n = 500, p = 2000, \rho = 0.01\lambda_p(\mathbf{A}^T\mathbf{A})$



Smooth unconstrained non-convex minimization

Problem (Mathematical formulation)

Let us consider the following problem formulation:

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

- f *is a smooth and possibly non-convex function.*
- *Recall that finding the global minimizer, i.e.,* $f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$, *is NP-hard*

Example: Image classification using neural networks

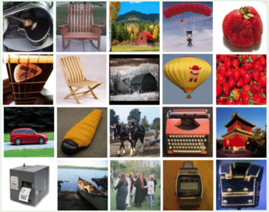
Neural network formulation

- $(\mathbf{a}_i, b_i)$: sample points, $\sigma(\cdot)$: non-linear activation function
- the function class $\mathcal{H}$ is given by $\mathcal{H} := \left\{h_{\mathbf{x}}(\mathbf{a}), \mathbf{x} \in \mathbb{R}^d\right\}$, where

$$\mathbf{x} = (\mathbf{W}_1, \boldsymbol{\mu}_1, \mathbf{W}_2, \boldsymbol{\mu}_2, \ldots, \mathbf{W}_k, \boldsymbol{\mu}_k), \quad \mathbf{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}, \quad \boldsymbol{\mu}_i \in \mathbb{R}^{d_i},$$
$$h_{\mathbf{x}}(\mathbf{a}) = \sigma\left(\mathbf{W}_k \sigma\left(\cdots \sigma\left(\mathbf{W}_2 \sigma\left(\mathbf{W}_1 \mathbf{a} + \boldsymbol{\mu}_1\right) + \boldsymbol{\mu}_2\right) \cdots\right) + \boldsymbol{\mu}_k\right)$$

- the loss function is given by $L(h_{\mathbf{x}}(\mathbf{a}), b) := (b - h_{\mathbf{x}}(\mathbf{a}))^2$.

Example: Image classification



Imagenet: 1000 object classes.
1.2M/100K train/test images
Below human level error rates!

Example: Phase retrieval for fourier ptychography

Definition (Phase retrieval)

Given a set of measurements of the amplitude of a signal, phase retrieval is the task of finding the phase for the original signal that satisfies certain constraints/properties.

Definition (Fourier ptychography)

Fourier ptychography is the task of reconstructing high-resolution images from low resolution samples, based on optical microscopy. It is a special case of phase retrieval problem.

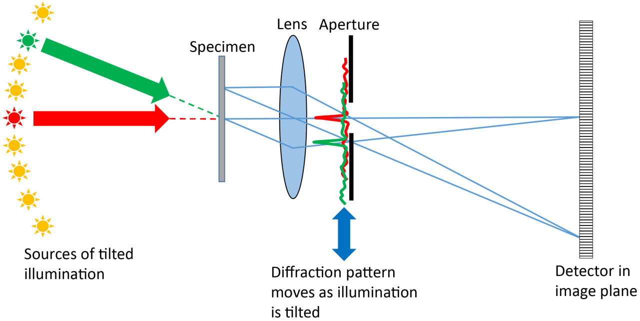
Example: Phase retrieval for fourier ptychography

Definition (Phase retrieval)

Given a set of measurements of the amplitude of a signal, phase retrieval is the task of finding the phase for the original signal that satisfies certain constraints/properties.

Definition (Fourier ptychography)

Fourier ptychography is the task of reconstructing high-resolution images from low resolution samples, based on optical microscopy. It is a special case of phase retrieval problem.



The necessity of non-convex optimization

Why non-convex?

- Inherent properties of optimization problem, e.g., phase retrieval
- Robustness or better estimation, e.g., binary classification with non-convex losses

Optimization Formulation: Phase Retrieval

$$\min_{\mathbf{x}} \||\mathbf{A}\mathbf{x}|^2 - \mathbf{b}\|_2^2$$

where $\mathbf{x} \in \mathbb{C}^p$ is a complex signal and $|\mathbf{A}\mathbf{x}|$ is the component-wise magnitude of the measurement $\mathbf{A}\mathbf{x}$.

Optimization Formulation: Binary Classification

$$\min_{x} \left\{ \frac{1}{n} \sum_{i=1}^{n} (b_i - g(\mathbf{a}_i, \mathbf{x}))^2 \right\}$$

where $g(\cdot, \cdot)$ is non-linear, and hence, the loss function is non-convex.

Notion of convergence: Stationarity

∘ Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice-differentiable and $\mathbf{x}^\star \in \arg\min_{x \in \mathbb{R}^d} f(\mathbf{x})$

Definition (Recall - First order stationary point)

A point $\bar{\mathbf{x}}$ is a first order stationary point of a twice differentiable function $f(\mathbf{x})$ if

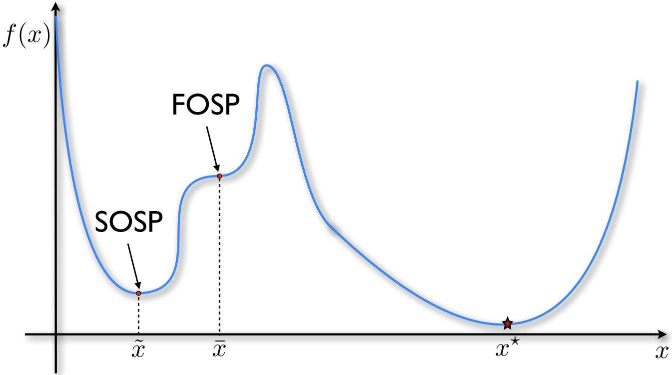
$$\nabla f(\bar{\mathbf{x}}) = \mathbf{0}.$$

Definition (Recall - Second order stationary point)

A point $\tilde{\mathbf{x}}$ is a second order stationary point of a twice differentiable function $f(\mathbf{x})$ if

$$\nabla f(\tilde{\mathbf{x}}) = \mathbf{0} \quad \text{and} \quad \nabla^2 f(\tilde{\mathbf{x}}) \succeq \mathbf{0}.$$

Geometric interpretation of stationarity



- Note that neither $\bar{\mathbf{x}}$, nor $\tilde{\mathbf{x}}$ is **not necessarily** equal to $\mathbf{x}^\star$!!

Assumptions and the gradient method

Assumption: Smoothness

Let f be a twice differentiable function that is L-Lipschitz gradient with respect to ℓ_2-norm, such that,

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})||_2 \leq L||\mathbf{x} - \mathbf{y}||_2$$

Gradient descent

Let $\alpha \leq \frac{1}{L}$ be the constant step size and $\mathbf{x}^0 \in \mathrm{dom}(f)$ be the initial point. Then, gradient method produces iterates using the following iterative update,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k)$$

Convergence rate and iteration complexity

Theorem ([1])

Let f be a twice differentiable L-Lipschitz gradient function, and $\alpha \leq \frac{1}{L}$. Then, gradient method converges to the FOSP with the following properties:

Convergence rate to an ϵ-FOSP:

$$\|\nabla f(\mathbf{x}^k)\| = O\left(\frac{1}{\sqrt{k}}\right).$$

Iteration complexity to reach an ϵ-FOSP:

$$O\left(\frac{1}{\epsilon^2}\right).$$

Wrap up!

- Questions/Self study on Monday 11:00 - 12:00
- Lecture on Friday 16:00 - 18:00
- Unsupervised work on Friday 18:00 - 19:00

References I

[1] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford.
Lower bounds for finding stationary points i.
Mathematical Programming, pages 1–50, 2019.
(Cited on page 58.)

[2] Peter J. Huber and Elvezio M. Ronchetti.
Robust Statistics.
John Wiley & Sons, Hoboken, NJ, 2009.
(Cited on page 4.)

[3] Lucien Le Cam.
Asymptotic methods in Statistical Decision Theory.
Springer-Verl., New York, NY, 1986.
(Cited on page 31.)

[4] Y. Nesterov.
Introductory lectures on convex optimization: A basic course, volume 87.
Springer, 2004.
(Cited on page 6.)

References II

[5] R Tyrrell Rockafellar.
Monotone operators and the proximal point algorithm.
SIAM journal on control and optimization, 14(5):877–898, 1976.
(Cited on page 14.)

[6] A. W. van der Vaart.
Asymptotic Statistics.
Cambridge Univ. Press, Cambridge, UK, 1998.
(Cited on page 31.)