Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 1: Introduction to Continuous Optimization

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

EE-556 (Fall 2019)

License Information for Mathematics of Data Slides

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

Logistics

- **Credits:** 4
- **Prerequisites:** Previous coursework in calculus, linear algebra, and probability is required. Familiarity with optimization is useful.
- **Grading:** Continuous control via homework exercises & exam (cf., syllabus)
- **HW topics:** Support vector machines, compressive subsampling, neural networks, power flow...
- **Moodle:** My courses > Genie electrique et electronique (EL) > Master > EE-556

  syllabus & course outline & HW exercises
- **TA's**: Thomas Sanchez (head TA), Paul Rolland, Maria Vladarean, Chaehwan Song, Ali Kavis, Mehmet Fatih Sahin, Fabian Latorre, and Ahmet Alacaoglu.

Outline

- This class:
  1. What is an optimization problem?
  2. Gradient descent: A basic introduction
  3. Common templates on convex/non-convex optimization
- Next class
  1. Review of probability, statistics and linear algebra

Recommended reading material

- Chapter 1 in S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, 2009.
- Chapter 1 in Nocedal, Jorge, and Wright, Stephen J., *Numerical Optimization*, Springer, 2006.

What is optimization?

Problem (Mathematical formulation)

The optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

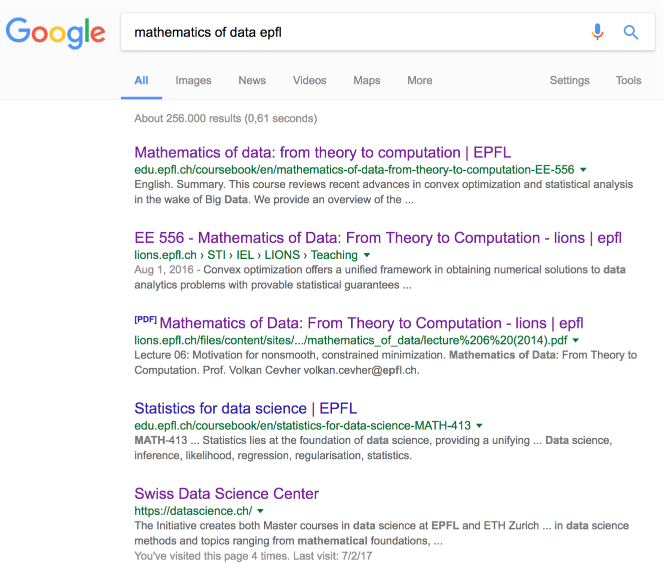
consists in finding a minimizer $\mathbf{x}^\star \in \mathcal{S}_f^\star$*. We say* (1) *has a solution if* $\mathcal{S}_f^\star$ *is non-empty*

Definition (Set of minimizers)

Let $f : \mathcal{X} \to \mathbb{R}$ and $\delta > 0$

- $\mathbf{x}^\star \in \mathcal{X}$ is a global minimizer of f if $\boxed{f(\mathbf{x}^\star) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathcal{X}}$
- $\mathbf{x}^\star_{\text{loc}} \in \mathcal{X}$ is a local minimizer of f if $\boxed{f(\mathbf{x}^\star_{\text{loc}}) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} : \left\|\mathbf{x} - \mathbf{x}^\star_{\text{loc}}\right\| \leq \delta}$
- $\boxed{f^\star := f(\mathbf{x}^\star)}$ is the global minimum value of f
- $\mathcal{S}_f^\star$ is the set of minimizers of f

Example: Google PageRank



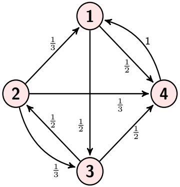
Modeling Google PageRank

- A basic model

- Compute the conditional probabilities:

$$P(\text{The Washington Post}|\text{Google News}) = 2/8$$
$$P(\text{The Atlantic}|\text{Google News}) = 1/8$$

- A toy graph and transition matrix:



$$\mathbf{E} = \begin{bmatrix} 0 & \frac{1}{3} & 0 & 1 \\ 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{2} & 0 \end{bmatrix}$$

Modeling Google PageRank

- Transition matrix for world wide web:

$$\mathbf{E} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

- $\sum_{i=1}^{n} c_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, n\}$ ($n \approx 5.91$billion)
- Estimated memory to store $\mathbf{E}$: 10^{12} GB!

Modeling Google PageRank

- Transition matrix for world wide web:

$$\mathbf{E} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

- $\sum_{i=1}^{n} c_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, n\}$ ($n \approx 5.91$billion)
- Estimated memory to store $\mathbf{E}$: 10^{12} GB!
- A bit of mathematical modeling:
  - r_i^k : Probability of being at node i at k^{th} state. Let us define a state vector

$$\mathbf{r}^k = \left[r_1^k, r_2^k, \ldots, r_n^k\right]^\top$$

  - Multiplying $\mathbf{r}^k$ by $\mathbf{E}$ takes one random step along the edges of the graph:

$$r_i^1 = \sum_{j=1}^{n} c_{ij} r_j^0 = (\mathbf{E}\mathbf{r}^0)_i,$$

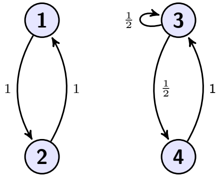
since $c_{ij} = P(i|j)$ (by the law of total probability).

Towards a Formal Formulation for Google PageRank

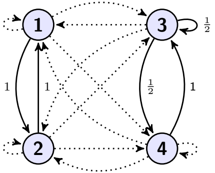
Goal

Find the ranking vector $\mathbf{r}^\star$ after an infinite number of random steps.

- Disconnected web: Initial state vector affects the ranking vector.



A solution: Model the event that the surfer will quit the current webpage and open another.

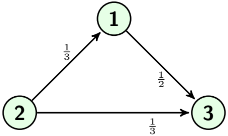


Towards a Formal Formulation for Google PageRank

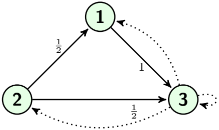
Goal

Find the ranking vector $\mathbf{r}^\star$ after an infinite number of random steps.

- Sink nodes: Column of zeros in $\mathbf{E}$, moves $\mathbf{r}$ to $\mathbf{0}$!



A solution: Create artifical links from sink nodes to all the nodes.



Towards a Formal Formulation for Google PageRank

Goal

Find the ranking vector $\mathbf{r}^\star$ after an infinite number of random steps.

- Disconnected web: Initial state vector affects the ranking vector.

A solution: Model the event that the surfer quits the current webpage to open another.

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} = \frac{1}{n}\mathbb{1}\mathbb{1}^\top$$

- Sink nodes: Column of zeros in $\mathbf{E}$, moves $\mathbf{r}$ to $\mathbf{0}$!

A solution: Create artifical links from sink nodes to all the nodes.

$$\lambda_i = \begin{cases} 1 & \text{if i}^{th} \text{ node is a sink node,} \\ 0 & \text{otherwise.} \end{cases}$$

Optimization formulation of Google PageRank

- Define the pagerank matrix $\mathbf{M}$ as

$$\mathbf{M} = (1-p)(\mathbf{E} + \frac{1}{n}\mathbb{1}\lambda^T) + p\mathbf{B}.$$

$\mathbf{M}$ is a column stochastic matrix.

Problem Formulation

- We characterize the solution as
  - $\mathbf{M}\mathbf{r}^\star = \mathbf{r}^\star$.
  - $\mathbf{r}^\star$ is a probability state vector:

$$r_i \geq 0, \quad \sum_{i=1}^{n} r_i = 1.$$

- Find $\mathbf{r} \geq 0$ such that $\mathbf{M}\mathbf{r} = \mathbf{r}$ and $\mathbb{1}^\top \mathbf{r} = 1$.

Optimization formulation of Google PageRank

- Define the pagerank matrix $\mathbf{M}$ as

$$\mathbf{M} = (1-p)(\mathbf{E} + \frac{1}{n}\mathbb{1}\lambda^T) + p\mathbf{B}.$$

$\mathbf{M}$ is a column stochastic matrix.

Problem Formulation

- We characterize the solution as
  - $\mathbf{M}\mathbf{r}^\star = \mathbf{r}^\star$.
  - $\mathbf{r}^\star$ is a probability state vector:

$$r_i \geq 0, \quad \sum_{i=1}^{n} r_i = 1.$$

- Find $\mathbf{r} \geq 0$ such that $\mathbf{M}\mathbf{r} = \mathbf{r}$ and $\mathbb{1}^\top \mathbf{r} = 1$.

Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ f(\mathbf{x}) = \frac{1}{2}\|\boldsymbol{M}\mathbf{x} - \mathbf{x}\|^2 + \frac{\gamma}{2}\left(\mathbb{1}^T\mathbf{x} - 1\right)^2 \right\}.$$

The general formulation: Least-squares

Optimization formulation (Least-squares estimator)

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

where $\mathbf{x}=\mathbf{r}$, $\mathbf{b}=\begin{bmatrix}\mathbf{r}\\ \frac{\gamma}{n}\mathbb{1}\end{bmatrix}$, $\mathbf{A}=\begin{bmatrix}\mathbf{M}\\ \frac{\gamma}{2n}\mathbb{1}\mathbb{1}^\top\end{bmatrix}$, $d=n$ in Google PageRank problem.

Linear regression problem

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

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

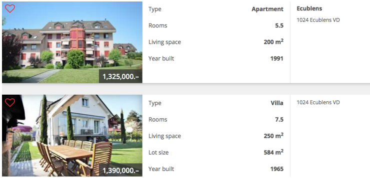
where $\mathbf{w}$ denotes unknown noise.

- Many other examples:

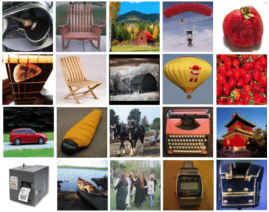
  Image reconstruction (MRI), stock market prediction, house pricing, etc.

Regression/Classification/Generation Examples

- Example: Taking a mortgage.
- Houses data (source: https://www.homegate.ch)



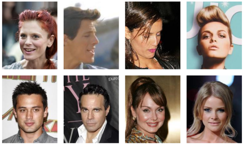
- Example: Image classification



- Imagenet: 1000 object classes. 1.2M/100K train/test images

source: https://www.imagenet.org

- Example: Image generation (GANs).
- Target distribution:



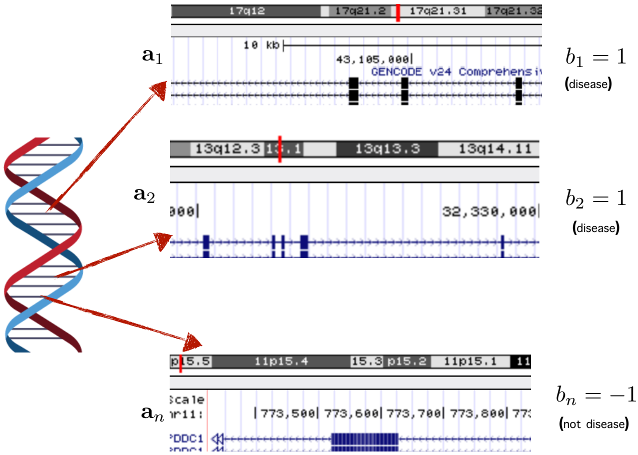
source: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

- How to generate (fake) images from the same distribution?

source: Progressive Growing of GANs for Improved Quality, Stability, and Variation Karras et al. ICLR 2018

Breast Cancer Detection

- Genome data for breast cancer (source: http://genome.ucsc.edu):



- A patient with genome data $\mathbf{a}_t$: has he got breast cancer or not (i.e., $b_t = 1$ or -1)?

Score-based Classifiers (I)

Goal

Predict either $b = 1$ or $b = -1$ given $\mathbf{a}$.

- For a genome sequence $\mathbf{a}$ compute a *score* $s_{\mathbf{x}}(\mathbf{a}) \in (-\infty, \infty)$:

$$\text{Example:} \quad \mathbf{a} \rightarrow s_{\mathbf{x}}(\mathbf{a}) = \underbrace{\mathbf{x}^\top}_{\text{weights = importance of genes}} \mathbf{a}$$

- Use the logistic function

$$t \mapsto h(t) := \frac{1}{1 + \exp(-t)}.$$

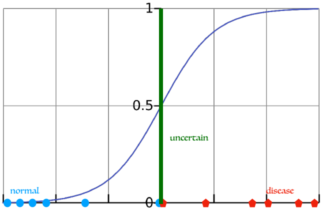
to transform $s_{\mathbf{x}}(\mathbf{a})$ into a probability of disease:

$$P(b = 1 | \mathbf{a}, \mathbf{x}) = h(s_{\mathbf{x}}(\mathbf{a})) \in (0, 1)$$

Score-based Classifiers (II)

- We have a model for the conditional probability of disease given $\mathbf{a}$

$$P(b = 1|\mathbf{a}, \mathbf{x}) = \frac{1}{1 + \exp(-s_{\mathbf{x}}(\mathbf{a}))}$$



$$P(b = 1|\mathbf{a}, \mathbf{x}) \begin{cases} > 0.5, & \text{if } s_{\mathbf{x}}(\mathbf{a}) \text{ is positive,} \\ \leq 0.5, & \text{otherwise.} \end{cases}$$

$$\text{Prediction} = \begin{cases} \text{disease}, & \text{if } P(b = 1|\mathbf{a}, \mathbf{x}) > 0.5, \\ \text{normal}, & \text{if } P(b = 1|\mathbf{a}, \mathbf{x}) < 0.5. \\ \text{uncertain}, & \text{if } P(b = 1|\mathbf{a}, \mathbf{x}) = 0.5. \end{cases}$$

Score-based Classifiers: How do we choose $\mathbf{x}$?

- Classification diagram:

$$
\begin{tikzcd}
(\mathbf{a}_i, b_i)_{i=1}^n \arrow[r, "\text{modeling}", "\text{parameter } \mathbf{x}"'] & P(b_i|\mathbf{a}_i, \mathbf{x}) \arrow[rr, "\text{independency}"] & & p(\mathbf{x}) := \prod_{i=1}^n P(b_i|\mathbf{a}_i, \mathbf{x}) \arrow[d, "\text{maximizing w.r.t } \mathbf{x}"] \\
\mathbf{a}_t \arrow[r] & P(b|\mathbf{a}_t, \mathbf{x}^\star) \arrow[d, "\text{evaluating logistic function}"'] & & \mathbf{x}^\star \arrow[ll] \\
& b_t & &
\end{tikzcd}
$$

- Maximizing $\log p(\mathbf{x})$ gives the log-likelihood estimator.

Logistic regression vs Neural networks

Optimization formulation (max Log-likelihood)

$$\boxed{\min_{\mathbf{x}\in\mathbb{R}^p} \underbrace{\log p(\mathbf{x}) = \sum_{i=1}^{n} \log(1+\exp\left(-b_i(s_{\mathbf{x}}(\mathbf{a}_i))\right))}_{f(\mathbf{x})}} \tag{2}$$

Problem (Choice of score function)

- *Logistic Regression:*

$$s_{\mathbf{x}}(\mathbf{a}) = \mathbf{x}^\top \mathbf{a} \qquad \Rightarrow \text{(2) } \textit{is convex} \tag{3}$$

- *Neural Networks* $(\mathbf{x} = [\mathbf{v}, \mathbf{W}])$*:*

$$s_{\mathbf{x}}(\mathbf{a}) = \mathbf{v}^\top \sigma(\mathbf{W}\mathbf{a}) \qquad \Rightarrow \text{(2) } \textit{is non-convex} \tag{4}$$

Optimization landscapes

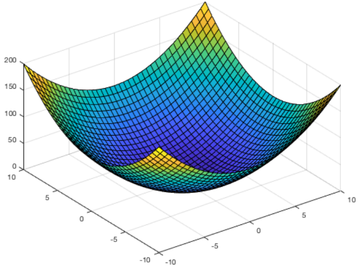


Figure: convex function

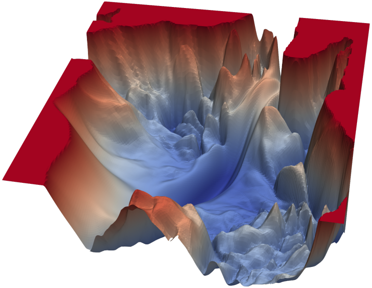


Figure: nonconvex function

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*.

Approximate vs. exact optimality and local optimality (I)

Is it possible to solve an optimization problem?

"In general, optimization problems are **unsolvable***"* - Y. Nesterov [1]

- Even when a closed-form solution exists, numerical accuracy may still be an issue.
- We must be content with **approximately** optimal solutions.
- Sometimes we can only find **locally** optimal solutions.

Approximate vs. exact optimality and local optimality (II)

Definition

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

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

Definition

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

$$\|\mathbf{x}^\star_\epsilon - \mathbf{x}^\star\| \leq \epsilon \,,$$

- The latter approximation guarantee is considered stronger.

Definition

We say that $\mathbf{x}^\star_{\text{loc}}$ is a local minimizer if for some $\delta > 0$,

$$f(\mathbf{x}^\star_{\text{loc}}) \leq f(\mathbf{x}) \qquad \text{for all } \mathbf{x} \text{ such that } \left\|\mathbf{x} - \mathbf{x}^\star_{\text{loc}}\right\| \leq \delta$$

Necessary/Sufficient optimality conditions

Lemma (First-order necessary local optimality condition)

Let $\mathbf{x}^\star_{loc}$ be a local minimizer of a differentiable convex function f. It holds that

$$\nabla f(\mathbf{x}^\star_{loc}) = \mathbf{0}.$$

Lemma (sufficient optimality condition for convex functions)

Let f be a convex function. It holds that

$$\mathbf{x}^\star_{loc} \textit{ is a local minimizer } \Rightarrow \mathbf{x}^\star_{loc} \textit{ is a global minimizer}$$

A gradient method

Gradient method

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

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

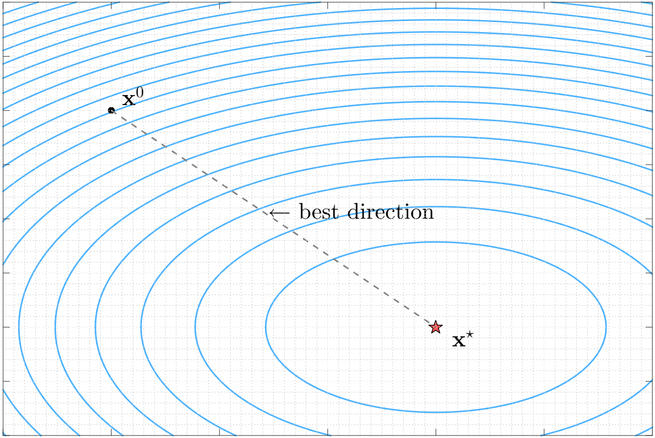
where α_k is a step-size to be chosen so that $\mathbf{x}^k$ converges to some $\mathbf{x}^\star_{\text{loc}}$.

Fixed-point characterization

Given that $\nabla f(\mathbf{x}^\star_{\text{loc}}) = 0$, the following fixed point condition holds

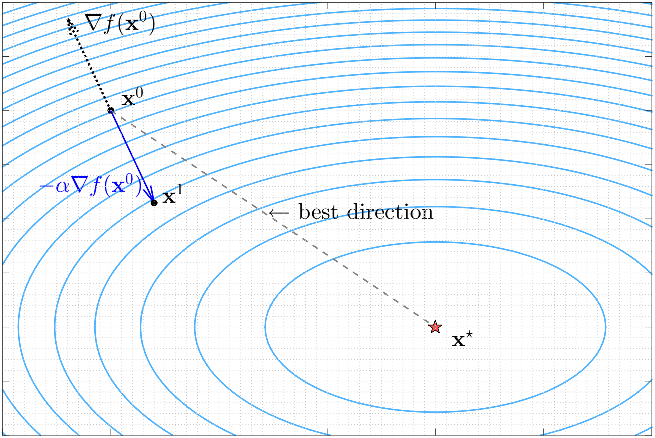
$$\mathbf{x}^\star_{\text{loc}} = \mathbf{x}^\star_{\text{loc}} - \alpha \nabla f(\mathbf{x}^\star_{\text{loc}}) \qquad \text{for all } \alpha \in \mathbb{R}$$

A simple example



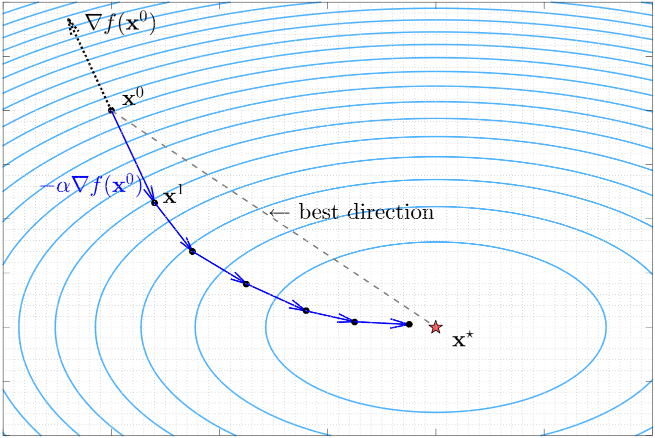
- Choose initial point: x^0, and a step size $\alpha > 0$.

A simple example



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

A simple example



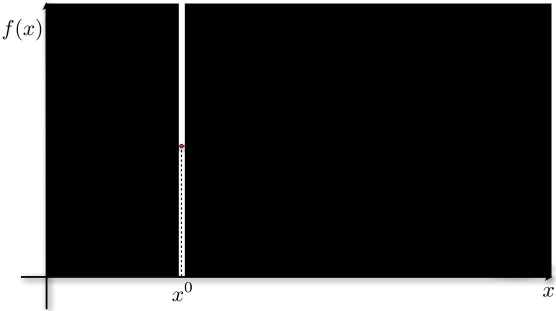
- Choose initial point: x^0, and a step size $\alpha > 0$.
- Take a step in the negative gradient direction: $x^{k+1} = x^k - \alpha\nabla f(x^k)$
- Repeat this procedure until x^k is accurate enough.

Challenges for an iterative optimization algorithm

Problem

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

- **Fog of war**

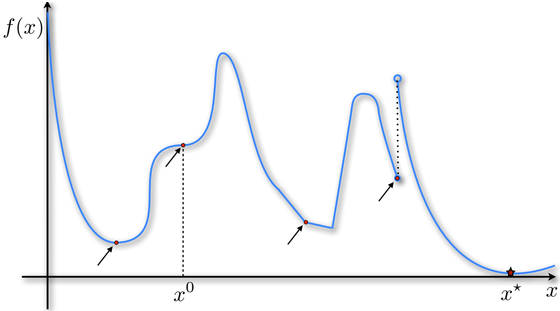


Challenges for an iterative optimization algorithm

Problem

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

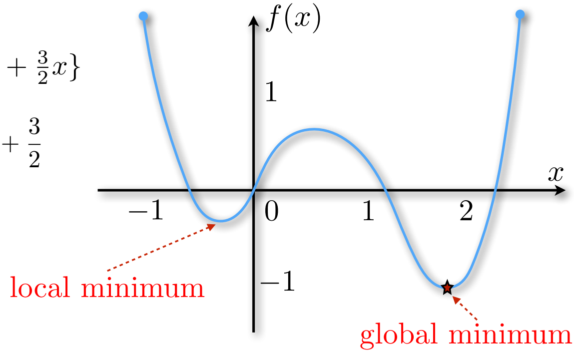
- Fog of war, non-differentiability, discontinuities, local minimizers, stationary points...



Local minimizers

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

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



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

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

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

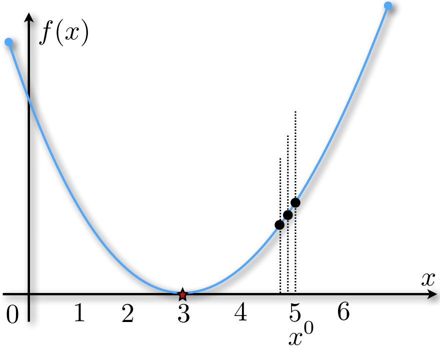
...

x^k is converging to **local minimizer**!

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 $\boldsymbol{\alpha = \frac{1}{10}}$

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

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

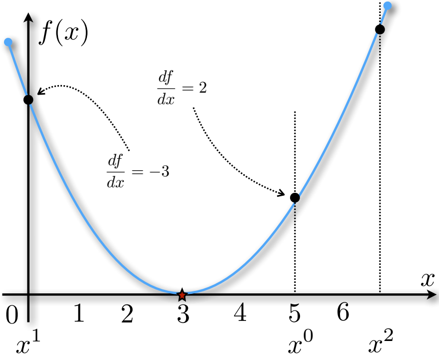
$\ldots$

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 $\boldsymbol{\alpha = \frac{5}{2}}$

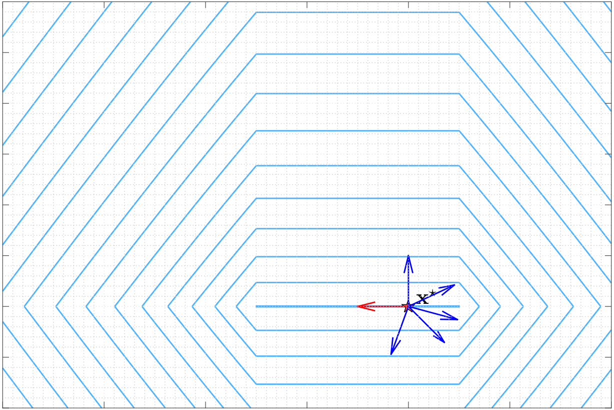
$x^1 = x^0 - \alpha \frac{df}{dx}\Big|_{x=x^0} = 5 - \frac{5}{2}2 = 0$

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

$\ldots$

x^k **diverges**.

Nonsmooth optimization

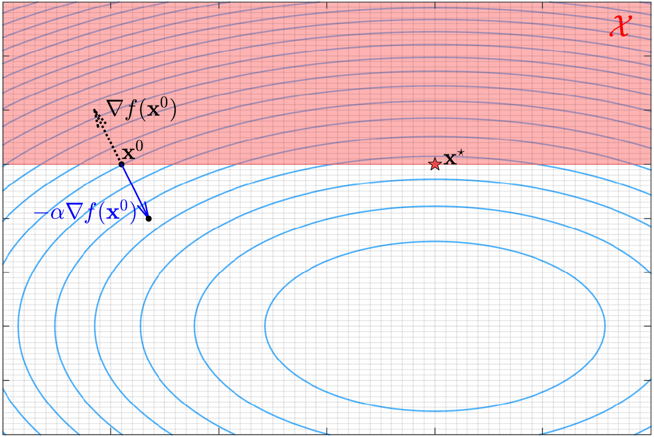


For nonsmooth optimization, the first order optimality condition

$$\nabla f(\mathbf{x}^\star) = \mathbf{0}$$

does not hold for every descent direction.

Constrained optimization

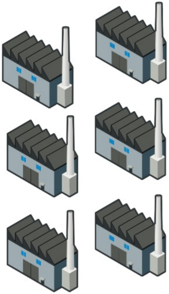


In many practical problems,
we need to **minimize** the cost **under some constraints**.

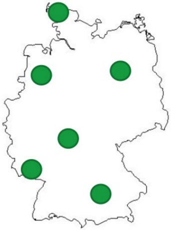
$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{x} \in \mathcal{X} \right\}$$

Example: Facility Location Problem

Assign facilities to locations to minimize the total assignment cost.



Facilities



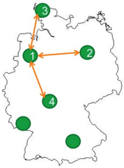
Locations

Example: Facility Location Problem

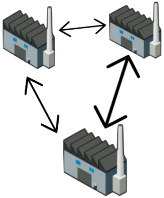
- **Goal:** To minimize the costs

- **Inputs:**

Distance between locations: $A = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & & \\ \vdots & & \ddots & \\ a_{n1} & & & 0 \end{bmatrix}$



Flow between facilities: $B = \begin{bmatrix} 0 & b_{12} & \dots & b_{1n} \\ b_{21} & \ddots & & \\ \vdots & & \ddots & \\ b_{n1} & & & 0 \end{bmatrix}$

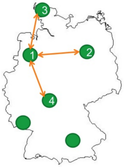


Example: Facility Location Problem

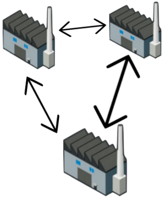
- **Goal:** To minimize the costs

- **Inputs:**

Distance between locations: $A = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & & \\ \vdots & & \ddots & \\ a_{n1} & & & 0 \end{bmatrix}$



Flow between facilities: $B = \begin{bmatrix} 0 & b_{12} & \dots & b_{1n} \\ b_{21} & \ddots & & \\ \vdots & & \ddots & \\ b_{n1} & & & 0 \end{bmatrix}$



- **Output:**

An assignment matrix $X \in \Pi_n$

Example: Quadratic Assignment Problem

Quadratic assignment problem, QAP, in the trace formulation

$$\boxed{\mu^\star := \min_{X \in \Pi_n} \operatorname{Tr}\left(AXBX^\top\right)}$$

Π_n : set of $n \times n$ permutation matrices

A and B : $n \times n$ real symmetric matrices

- **Non-convex, quadratic** objective over the (discrete) set of permutation matrices
- Convex relaxations exist

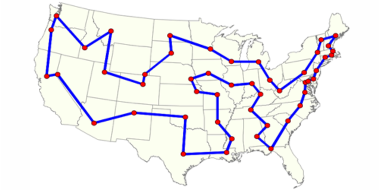
QAP example: Traveling Salesman Problem

Find a path passing from all vertices (*e.g.*, cities) once to minimize the total trip time

$A = \frac{1}{2}D$, D: Matrix of edge weights such that $D_{ij} = D_{ji} \geq 0$ $(i \neq j)$

$B = C$ C: The adjacency matrix of the cities

$$\boxed{TSP_{opt} := \min_{X \in \Pi_n} \mathrm{Tr}\left(\frac{1}{2}DXCX^\top\right)}$$



Implications of convexity

If f is convex,

- any local minimizer is also a **global minimizer**,
- we have a **principled step-size** selection,
- we can handle **non-smooth** problems like **constraints**.

Unfortunately, **convexity does not imply tractability**...

Implications of non-convexity

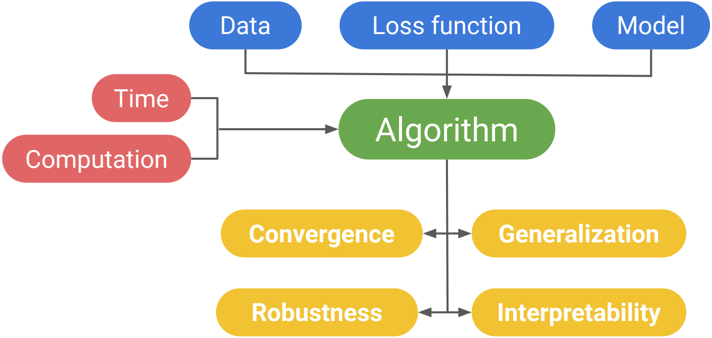
If f is not convex,

- Gradient descent converges only to **first order stationary points** ,
- Sometimes, they are **global minimizers**
- In certain applications, local minimizers can be **good enough**

Fortunately, **non-convexity does not imply intractability**

or **uselessness**...

Overview of mathematics of data



Do not forget!

- Lecture on Monday and recitation on Friday
  - Lecture: Basic probability theory and statistics.
  - Recitation: Terminology of optimization theory, gradient descent for logistic regression.

References

[1] Yu. Nesterov.
Introductory Lectures on Convex Optimization: A Basic Course.
Kluwer, Boston, MA, 2004.