Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 1: Introduction to Convex Optimization

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

EE-556 (Fall 2017)

License Information for Mathematics of Data Slides

Motivation

- Collecting data at unprecedented rates

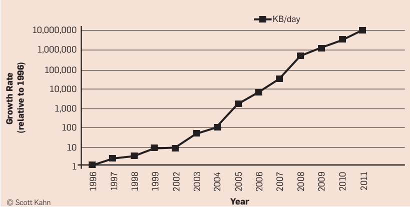


Figure: Next-generation sequencing data size
"Big data and its technical challenges."
[Communications of the ACM, July 2014]

Motivation

- Collecting data at unprecedented rates
- ***Outpacing the growth of computation***
  - data: more of a *burden* than a blessing

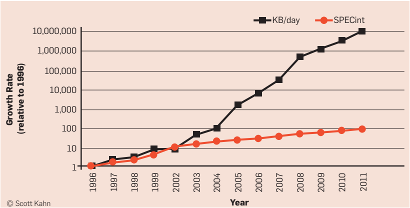


Figure: Next-gen. sequencing data size vs SPECint. *"Big data and its technical challenges."* [Communications of the ACM, July 2014]

Motivation

- Collecting data at unprecedented rates
- ***Outpacing the growth of computation***
  - data: more of a *burden* than a blessing

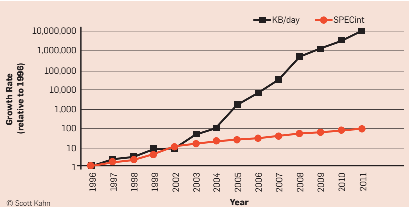


Figure: Next-gen. sequencing data size vs SPECint.
"Big data and its technical challenges."
[Communications of the ACM, July 2014]

- Dogma: *Running time of an algorithm increases with the size of its input data*

$\Rightarrow$ ***Important problems will take increasingly more time to solve!***

Challenge (EE-556)

Improve inferential precision within a time-budget as the data grows

- Convex optimization in the context of statistical analysis
  - review of linear algebra & probability theory in recitations

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, power flow...
- **Moodle:** My courses> Genie electrique et electronique (EL) > Master > EE-556

  syllabus & course outline & HW exercises
- **TA's**: Alp Yurtsever and Junhong Lin (head TA's); Marwa El Halabi, Baran Gozcu, Bang Cong Vu, Quang Van Nguyen, Ilija Bogunovic, Yen-Huan Li, Ya-Ping Hsieh, Kamal Parameswaran, and Ahmet Alacaoglu

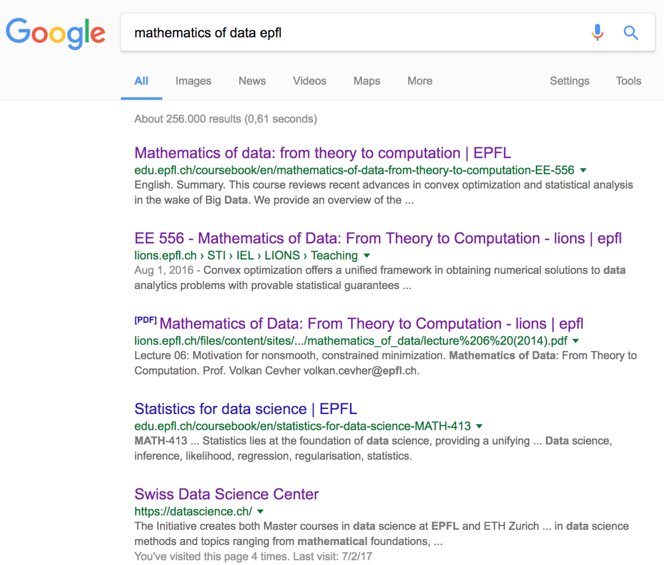
Outline

- This class:
  1. What is an optimization problem?
  2. Gradient descent: A basic introduction
  3. Common templates on 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.

Google PageRank



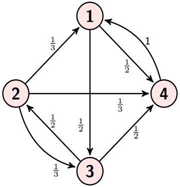
Modeling Google PageRank

- A basic model

- Compute the conditional probabilities:

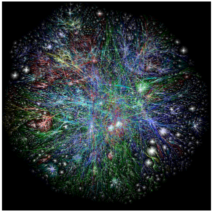
$$P(\text{The Washington Post}|\text{Google News}) = 2/8$$
$$P(\text{The Atlantic}|\text{Google News}) = 1/8$$

- 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

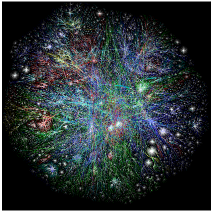


- Transition matrix for world wide web:

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

- $\sum_{i=1}^{n} c_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, n\}$ ($n \approx 4.5$billion)
- Estimated memory to store $\mathbf{E}$: 10^{11} GB!

Modeling Google PageRank



- Transition matrix for world wide web:

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

- $\sum_{i=1}^{n} c_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, n\}$ ($n \approx 4.5$billion)
- Estimated memory to store $\mathbf{E}$: 10^{11} 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:

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

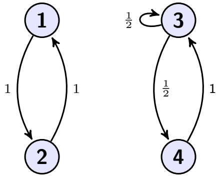
since $c_{ij} = P(i|j)$ (by the law of total probability).

Towards a Formal Formulation for Google PageRank

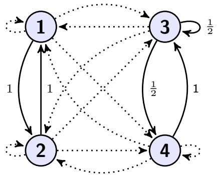
Goal

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.

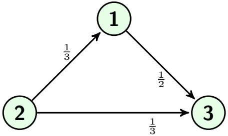


Towards a Formal Formulation for Google PageRank

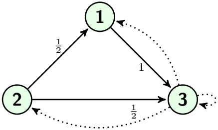
Goal

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.

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

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 $\mathbb{1}^\top \mathbf{r} = 1$.

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 proglem.

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.

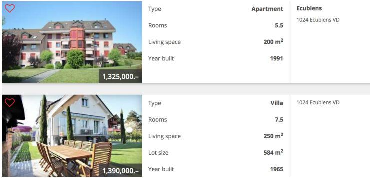
- Many other examples:

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

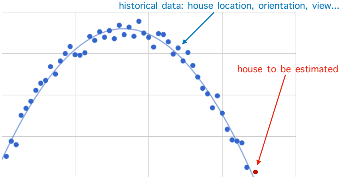
Regression vs Classification

Regression

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



- Banks: estimate the loan based on location, orientation, view, etc.

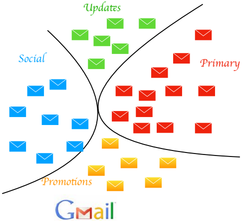


- Output values: continuous.

Classification

- Example: Spam classification.
- Incoming emails:

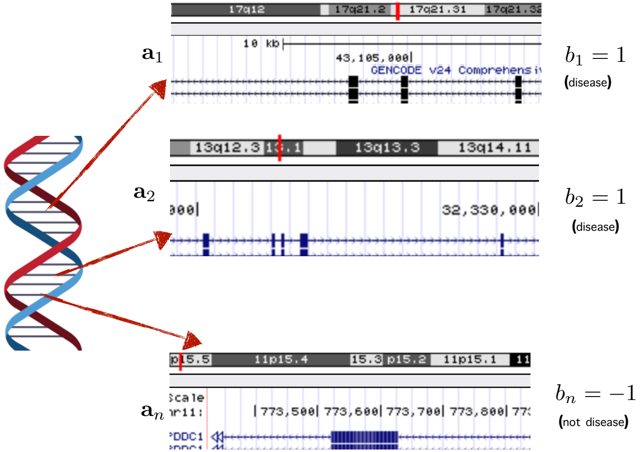
- How to group emails in categories?



- Output values: discrete, categorical.

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)?

Breast Cancer Detection

Goal

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

- Pre-examination: extract important genes from the genome sequence $\mathbf{a}_t$:

$$\mathbf{a}_t \to \underbrace{\mathbf{a}_t^\top \mathbf{x}}_{\uparrow \text{ weights = importance of genes}} + \underbrace{\mu}_{\uparrow \text{ intercept = bias}}$$

- Conclusion: choose a probability P and predict as follow:

$$b_t = \begin{cases} 1, & \text{if } P(b = 1|\mathbf{a}_t) > P(b = -1|\mathbf{a}_t), \\ -1, & \text{otherwise.} \end{cases}$$

- How do we model probabilities?

logistic function

Classification with logistic transform

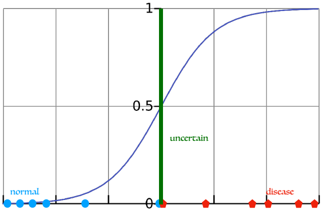
- Logistic function:

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

- Model the conditional probability of the label b given test result $\mathbf{a}$

$$P(b|\mathbf{a}) := h\Big(b(\mathbf{a}^\top \mathbf{x} + \mu)\Big) = \frac{1}{1+\exp\left(-b(\mathbf{a}^\top \mathbf{x} + \mu)\right)}.$$

where $\mathbf{x}$ = weights, μ = intercept.



$$P(b|\mathbf{a}) \begin{cases} \geq 0.5, & \text{if } \mathbf{a}^\top \mathbf{x} + \mu, b \text{ have the same sign,} \\ < 0.5, & \text{otherwise.} \end{cases}$$

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

$P(b|\mathbf{a}) = 0.5$ (green line): uncertain.

Classification: How does it work?

- Classification diagram:

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

$$\downarrow \text{maximizing w.r.t } \mathbf{x}$$

$$\mathbf{a}_t \longrightarrow P(b|\mathbf{a}_t, \mathbf{x}^\star) \longleftarrow \qquad \mathbf{x}^\star$$

$$\text{evaluating logistic function} \downarrow$$

$$b_t$$

- Maximizing $\log p(\mathbf{x})$ gives the log-likelihood estimator (covered later in this course).

Logistic regression

Problem (**Logistic regression**)

Given a sample vector $\mathbf{a}_i \in \mathbb{R}^p$ *and a binary class label* $b_i \in \{-1, +1\}$ *(*$i = 1, \ldots, n$*), we define the conditional probability of* b_i *given* $\mathbf{a}_i$ *as:*

$$\mathbb{P}(b_i | \mathbf{a}_i, \mathbf{x}^\natural, \mu) \propto 1/(1 + e^{-b_i(\langle \mathbf{x}^\natural, \mathbf{a}_i \rangle + \mu)}),$$

where $\mathbf{x}^\natural \in \mathbb{R}^p$ *is some true weight vector,* μ *is called the intercept.*
How do we estimate $\mathbf{x}^\natural$ *given the sample vectors, the binary labels, and* μ*?*
Logistic regression is a classification problem!

Log-likelihood

$$\log p(\mathbf{x}) = -\sum_{i=1}^{n} \log(1 + \exp\left(-b_i(\mathbf{a}_i^\top \mathbf{x} + \mu)\right))$$

Optimization formulation

$$\min_{\mathbf{x} \in \mathbb{R}^p} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-b_i(\mathbf{a}_i^T \mathbf{x} + \mu)))}_{f(\mathbf{x})} \tag{1}$$

Unconstrained minimization

Problem (**Mathematical formulation**)

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

$$\boxed{F^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x}) := f(\mathbf{x})\}} \tag{2}$$

Note that (2) *is unconstrained.*

Definition (Optimal solutions and solution set)

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

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

Is it possible to solve a convex 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.

Definition

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

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

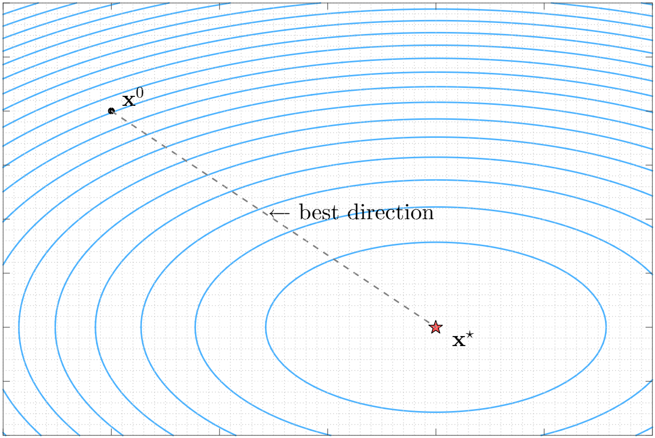
Definition

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

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

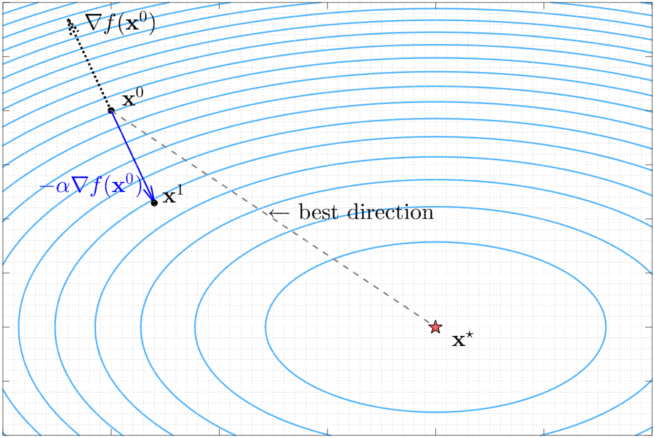
- The latter approximation guarantee is considered stronger.

A simple example



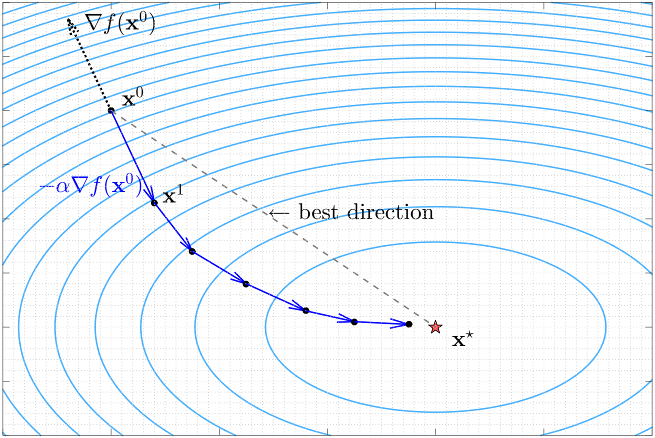
- Choose initial point: x^0, and a step size $\alpha > 0$.

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.

A gradient method

Lemma (First-order necessary optimality condition)

Let $\mathbf{x}^\star$ *be a global minimum of a differentiable convex function* f*. Then, it holds that*

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

Fixed-point characterization

Multiply by -1 and add $\mathbf{x}^\star$ to both sides to obtain a fixed point condition,

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

Gradient method

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

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

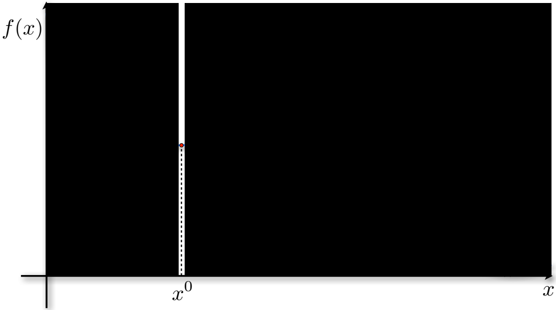
where α_k is a step-size to be chosen so that $\mathbf{x}^k$ converges to $\mathbf{x}^\star$.

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**

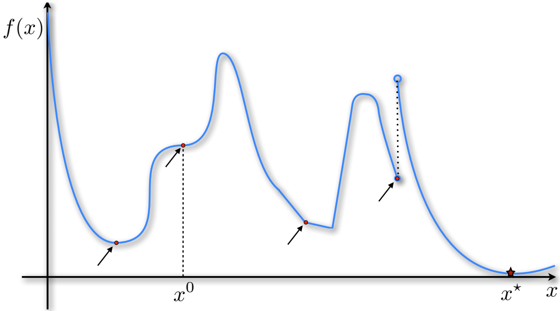


Challenges for an iterative optimization algorithm

Problem

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

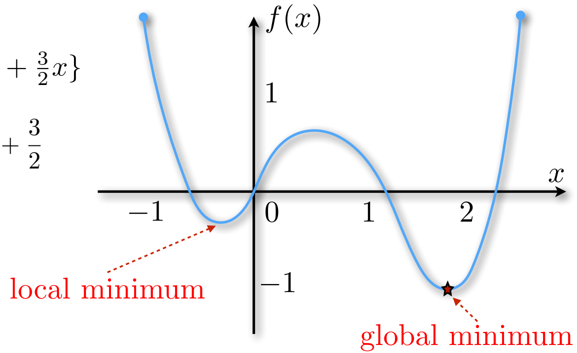
- Fog of war, non-differentiability, discontinuities, local minima, stationary points...



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 $\alpha = \frac{1}{6}$

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

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

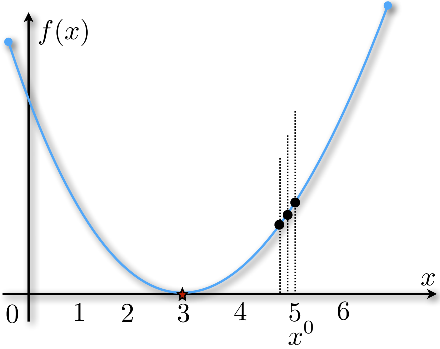
...

x^k is converging to **local 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 $\boldsymbol{\alpha = \frac{1}{10}}$

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

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

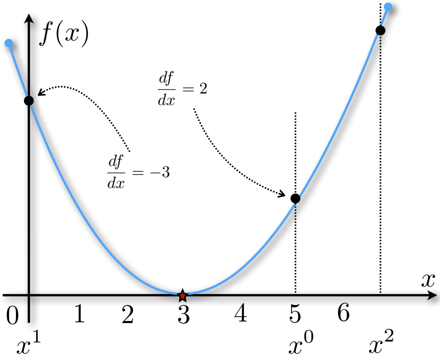
$\ldots$

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

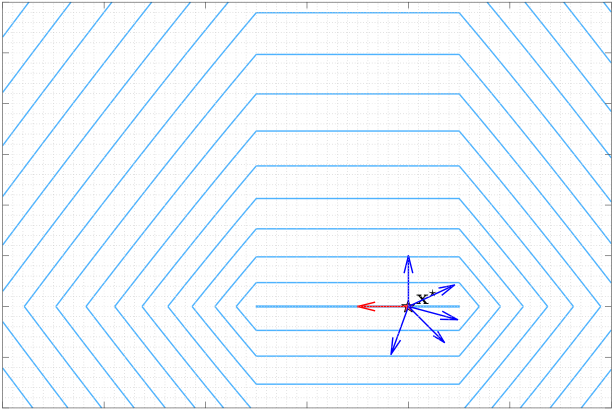
$x^1 = x^0 - \alpha \frac{df}{dx}\Big|_{x=x^0} = 5 - \frac{5}{2}2 = 0$

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

$\ldots$

x^k **diverges**.

Nonsmooth optimization

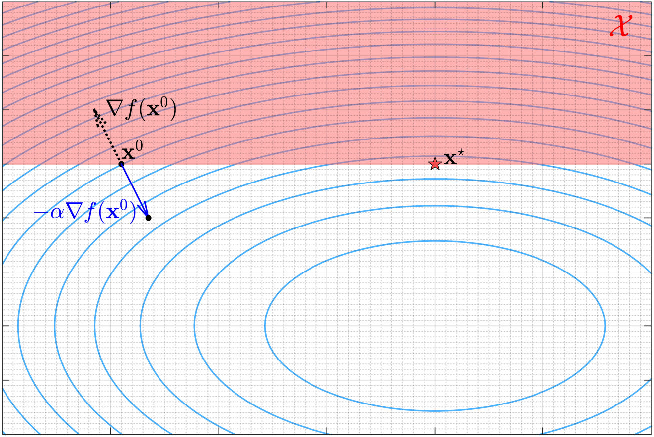


For nonsmooth optimization, the first order optimality condition

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

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: Optimal Power Flow

Goal is to design generator outputs to minimize the cost.

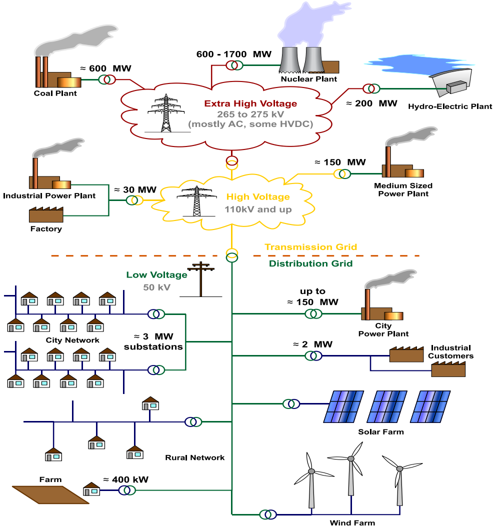


image from http://wikipedia.org/wiki/Automatic_Generation_Control

OPF: Model

Objective:

minimize total generation cost by designing the generator outputs

subject to:

- physical constraints - conservation of energy

  power generated - power used = power lost

- generator limit constraints

  minimum and maximum power output of each generator

- line capacity constraints

  maximum power that can be transferred from each line

OPF: Notation

A power network with

- set of buses $\mathcal{N} := \{1, 2, \ldots, n\}$
- set of generator buses $\mathcal{G} \subseteq \mathcal{N}$
- set of flow lines $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$

Denote by

- known constant power load $P_{D_j} + iQ_{D_j}$ at bus $j \in \mathcal{N}$
- known admittance y_{lm} at line $(l, m) \in \mathcal{L}$
- unknown generator output $P_{G_j} + iQ_{G_j}$ at generator bus $j \in \mathcal{G}$

To formulate this problem, define

- V_j: unknown complex voltage at bus $j \in \mathcal{N}$
- P_{lm}: unknown active power transferred from bus $l \in \mathcal{N}$ through the line $(l, m) \in \mathcal{L}$
- S_{lm}: unknown complex power transferred from bus $l \in \mathcal{N}$ through the line $(l, m) \in \mathcal{L}$
- $f_j(P_{G_j})$: known convex generating cost function for generator $j \in \mathcal{G}$

OPF: Formulation

$$
\begin{aligned}
\underset{\begin{bmatrix}\mathbf{V}\\ \mathbf{P}_G\\ \mathbf{Q}_G\end{bmatrix}}{\text{minimize}} \quad & \sum_{j\in\mathcal{G}} f_j(P_{G_j}) \\
\text{subject to} \quad & P_{G\,j} - P_{D\,j} = \sum_{l\in\mathcal{N}(j)} \mathrm{Re}\{V_j(V_j^* - V_l^*)y_{jl}^*\} \\
& Q_{G\,j} - Q_{D\,j} = \sum_{l\in\mathcal{N}(j)} \mathrm{Im}\{V_j(V_j^* - V_l^*)y_{jl}^*\} \\
& P_j^{\text{min}} \leq P_{G\,j} \leq P_j^{\text{max}} \\
& Q_j^{\text{min}} \leq Q_{G\,j} \leq Q_j^{\text{max}} \\
& V_j^{\text{min}} \leq |V_j| \leq V_j^{\text{max}} \\
& |\mathrm{Re}\{V_l(V_l^* - V_m^*)y_{lm}^*\}| \leq P_{lm}^{\text{max}} \\
& \qquad \forall j \in \mathcal{N},\ \forall (l,m) \in \mathcal{L}
\end{aligned}
$$

energy conservation (first two constraints)

generator limits (next three constraints)

line capacity (last constraint)

This is a **nonsmooth, nonconvex, constrained** optimization problem.
In the final homework, we will solve this problem via a convex relaxation.

Convexity is the key

If f is convex,

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

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

Do not forget!

- Recitation on Friday
  - A short review of linear algebra
  - Exercise session for the lecture

References

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