Convex Optimization & Machine Learning

Introduction to Optimization

mcuturi@i.kyoto-u.ac.jp
Why do we need optimization in machine learning

- We want to find the best possible decision w.r.t. a problem
- In a supervised setting for instance, we want to **learn** a map $\mathcal{X} \rightarrow \mathcal{Y}$
- We consider a set of candidates $\mathcal{F}$ for such a decision
Why do we need optimization in machine learning

- **Quantify** how well a candidate function in $\mathcal{F}$ fits with the database
  - define a **data-dependent** criterion $C_{\text{data}}$
  - Typically, given a function $f$, $C_{\text{data}}(f)$ is **big** if $f$ **not accurate** on the data.
- Given both $\mathcal{F}$ and $C_{\text{data}}$, a method to find an **optimal** candidate:

$$\min_{f \in \mathcal{F}} C_{\text{data}}(f).$$
Quizz: Regression

Scatter plot of Rent vs. Surface

\begin{align*}
  y &= 0.13x + 2.4 \\
  y &= -0.0033x^2 + 0.35x - 0.71 \\
  y &= 6.2e^{-05}x^3 - 0.01x^2 + 0.59x - 3.1 \\
  y &= 1.4e^{-06}x^4 - 0.00016x^3 + 0.0016x^2 + 0.33x - 1.2 \\
  y &= -1.8e^{-07}x^5 + 3.7e^{-05}x^4 - 0.0028x^3 + 0.092x^2 - 1.1x + 7.1
\end{align*}

Does Least-Square regression fall into this approach?

1. Yes  
2. No
Quizz: $k$-nearest neighbors

Does $k$-nearest neighbors fall into this approach?

1. Yes  
2. No
Quizz: SVM

Does the SVM fall into this approach?

1. Yes  
2. No
What is optimization?

• A general formulation for optimization problem is that of defining
  ∘ unknown variables $x_1, x_2, \cdots, x_n \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$, and solve

  minimize (or maximize) $f(x_1, x_2, \cdots, x_n)$,

  subject to $f_i(x_1, x_2, \cdots, x_n) \begin{cases} <, > \end{cases} b_i, i = 1, 2, \cdots, m;$
What is optimization?

- A general formulation for optimization problem is that of defining
  - unknown variables $x_1, x_2, \cdots, x_n \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$, and solve

    \[\begin{align*}
    \text{minimize (or maximize)} & \quad f(x_1, x_2, \cdots, x_n), \\
    \text{subject to} & \quad f_i(x_1, x_2, \cdots, x_n) \begin{cases} <,> \\
    = \end{cases} b_i, i = 1, 2, \cdots, m; \end{align*}\]

- where
  - the $b_i \in \mathbb{R}$
  - functions $f$ (objective) and $g_1, g_2, \cdots, g_m$ (constraints) are functions

    \[\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n \rightarrow \mathbb{R}\]
What is optimization?

- A general formulation for optimization problem is that of defining
  - unknown variables $x_1, x_2, \cdots, x_n \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$, and solve
    
    minimize (or maximize) \quad f(x_1, x_2, \cdots, x_n),
    
    subject to \quad f_i(x_1, x_2, \cdots, x_n) \begin{cases} <, > \\ = \\ \leq, \geq \end{cases} b_i, i = 1, 2, \cdots, m;

- the sets $\mathcal{X}_i$ need not be the same, as $\mathcal{X}_i$ might be
  - $\mathbb{R}$ scalar numbers,
  - $\mathbb{Z}$ integers,
  - $\mathbf{S}_n^+$ positive definite matrices,
  - strings of letters,
  - etc.

- When the $\mathcal{X}_i$ are different, the adjective *mixed* usually comes in.
Optimization is field of **applied mathematics** on its own.

Also called **Mathematical Programming**.

**Mathematical Programming** is not about programming code for mathematics!
Mathematical Programming is not about programming code for mathematics!

- George Dantzig, who proposed the “first” optimization algorithm, explains:
  
  - The military refer to their various plans or proposed schedules of training, logistical supply and deployment of combat units as a program. When I first analyzed the Air Force planning problem and saw that it could be formulated as a system of linear inequalities, I called my paper Programming in a Linear Structure. Note that the term program was used for linear programs long before it was used as the set of instructions used by a computer. In the early days, these instructions were called codes.
In the summer of 1948, Koopmans and I visited the Rand Corporation. One day we took a stroll along the Santa Monica beach. Koopmans said: Why not shorten Programming in a Linear Structure to Linear Programming? I replied: That's it! From now on that will be its name. Later that day I gave a talk at Rand, entitled Linear Programming; years later Tucker shortened it to Linear Program.

The term Mathematical Programming is due to Robert Dorfman of Harvard, who felt as early as 1949 that the term Linear Programming was too restrictive.
• Today mathematical programming = optimization. A relatively new discipline

○ What seems to characterize the pre-1947 era was lack of any interest in trying to optimize. T. Motzkin in his scholarly thesis written in 1936 cites only 42 papers on linear inequality systems, none of which mentioned an objective function.
Before we move on to some reminders

- Keep in mind that optimization is hard. very hard in general.
- In 60 years, we have gone from nothing to quite a few successes.
- But always keep in mind that most problems are intractable.
- For some particular problems there is hope: CONVEX problems
Before we move on to some reminders

Evaluation = Programming Assignments
Reminders

Sources: Stephen Boyd’s slides
**Reminders: Convex set**

**line segment** between $x_1$ and $x_2$: all points

\[ \{ x = \lambda x_1 + (1 - \lambda)x_2, \quad 0 \leq \lambda \leq 1 \} \]

**convex set**: contains line segment between any two points in the set

\[ C \text{ is convex } \iff \forall x_1, x_2 \in C, 0 \leq \lambda \leq 1; \quad \lambda x_1 + (1 - \lambda)x_2 \in C \]

**examples** (one convex, two nonconvex sets)
**Convex combination and convex hull**

**Convex combination** of $x_1, \ldots, x_k$: any point $x$ of the form

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_k x_k$$

with $\lambda_1 + \cdots + \lambda_k = 1$, $\lambda_i \geq 0$

**Convex hull** $\langle S \rangle$: set of all convex combinations of points in $S$
**Convex cone**

**conic (nonnegative) combination** of \( x_1 \) and \( x_2 \): any point of the form

\[
x = \lambda_1 x_1 + \lambda_2 x_2
\]

with \( \lambda_1 \geq 0, \lambda_2 \geq 0 \)

**convex cone**: set that contains all conic combinations of points in the set
**Hyperplanes and halfspaces**

**hyperplane:** set of the form \( \{ x \mid a^T x = b \} \) \((a \neq 0)\)

- \(a\) is the normal vector
- hyperplanes are affine and convex; halfspaces are convex

**halfspace:** set of the form \( \{ x \mid a^T x \leq b \} \) \((a \neq 0)\)
**Norm balls and norm cones**

**norm:** a function $\| \cdot \|$ that satisfies

- $\| x \| \geq 0$; $\| x \| = 0$ if and only if $x = 0$
- $\| tx \| = |t| \| x \|$ for $t \in \mathbb{R}$
- $\| x + y \| \leq \| x \| + \| y \|$

**notation:** $\| \cdot \|$ is general (unspecified) norm; $\| \cdot \|_{symb}$ is particular norm

**norm ball** with center $x_c$ and radius $r$: $\{ x \mid \| x - x_c \| \leq r \}$

**norm cone:** $\{ (x, t) \mid \| x \| \leq t \}$

Euclidean norm cone is called second-order cone

Cones are convex
Usual norms for vectors in $\mathbb{R}^d$

- $l_2$ norm:
  \[ \|x\|_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{d} x_i^2} \]

- $l_1$ norm:
  \[ \|x\|_1 = \sum_{i=1}^{d} |x_i| \]

- $l_p$ norm:
  \[ \|x\|_p = \left( \sum_{i=1}^{d} |x_i|^p \right)^{\frac{1}{p}} \]
The **unit ball** of the $l_p$ norm is $\{x \mid \|x\|_p \leq 1\}$

The **unit ball** of the $l_p$ norm is convex.

1. True  2. False
Polyhedra

solution set of finitely many linear inequalities and equalities

\[ Ax \preceq b, \quad Cx = d \]

\((A \in \mathbb{R}^{m \times n}, C \in \mathbb{R}^{p \times n}, \preceq \text{ is componentwise inequality})\)

polyhedron is intersection of \textbf{finite} number of halfspaces and hyperplanes
Positive semidefinite cone

notation:

- \( S^n \) is set of symmetric \( n \times n \) matrices
- \( S_+^n = \{ X \in S^n \mid X \succeq 0 \} \): positive semidefinite \( n \times n \) matrices

\[
X \in S_+^n \iff z^T X z \geq 0 \text{ for all } z
\]

- \( S_+^n \) is a convex cone
- \( S_{++}^n = \{ X \in S^n \mid X \succ 0 \} \): positive definite \( n \times n \) matrices

example: \[
\begin{bmatrix}
x & y \\
y & z
\end{bmatrix} \in S_+^2
\]
Euclidean balls and ellipsoids

(Euclidean) ball with center \( x_c \) and radius \( r \):

\[
B(x_c, r) = \{ x \mid \| x - x_c \|_2 \leq r \} = \{ x_c + ru \mid \| u \|_2 \leq 1 \}
\]

ellipsoid: set of the form

\[
\{ x \mid (x - x_c)^T P^{-1} (x - x_c) \leq 1 \}
\]

with \( P \in S^n_{++} \) (i.e., \( P \) symmetric positive definite)

other representation: \( \{ x_c + Au \mid \| u \|_2 \leq 1 \} \) with \( A \) square and nonsingular
optimization problems

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

- \( x = (x_1, \ldots, x_n) \): optimization variables
- \( f_0 : \mathbb{R}^n \to \mathbb{R} \): objective function
- \( f_i : \mathbb{R}^n \to \mathbb{R}, \ i = 1, \ldots, m \): constraint functions

**optimal solution** \( x^* \) has smallest value of \( f_0 \) among all vectors that satisfy the constraints
minimize \( f_0(x) \)
subject to \( f_i(x) \leq b_i, \quad i = 1, \ldots, m \)

If an optimization problem has an optimal solution \( x^* \), this solution is unique.

1. True    2. False
Examples

**portfolio optimization**
- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

**device sizing in electronic circuits**
- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
- objective: power consumption

**data fitting**
- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error
Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, *e.g.*, very long computation time, or not always finding the solution

exceptions: certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems
Least-squares

\[
\text{minimize } \|Ax - b\|_2^2
\]

solving least-squares problems

- analytical solution: \( x^* = (A^T A)^{-1} A^T b \)
- reliable and efficient algorithms and software
- computation time proportional to \( n^2k \) (\( A \in \mathbb{R}^{k \times n} \)); less if structured
- a mature technology

using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility (\( e.g. \), including weights, adding regularization terms)
Linear programming

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2m$ if $m \geq n$; less with structure
- a mature technology

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs
  (e.g., problems involving $\ell_1$- or $\ell_{\infty}$-norms, piecewise-linear functions)
Convex optimization problem

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

- objective and constraint functions are convex:

\[
f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)
\]

if \( \alpha + \beta = 1, \alpha \geq 0, \beta \geq 0 \)

- includes least-squares problems and linear programs as special cases
Convex optimization problem

solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to \(\max\{n^3, n^2m, F\}\), where \(F\) is cost of evaluating \(f_i\)'s and their first and second derivatives
- almost a technology

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization