1. Introduction

- course goals and topics
- mathematical optimization
- rough plan

Course goals and topics

Prerequisites

Requirements: One of the following courses

- CPSC 302 (Numerical Computation for Algebraic Problems)
- CPSC 303 (Numerical Approximation and Discretization)
- MATH 307 (Applied Linear Algebra)

In practice

- (very) comfortable with linear algebra
- multivariate calculus
- comfortable programming
  eg, Julia (recommended), Python, or Matlab (nor recommended)

Book (some reading and homework assignments)

- *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with Matlab*, by Amir Beck

Course goals and topics

Goals

- recognize and formulate the main optimization problem classes
- understand the strengths and weaknesses of standard algorithms
- hands-on experience with useful software

Emphasis

- formulating problems
- algorithms
- case studies
- mathematical software

Grades and homework policy

Grades

- 6 homework assignments 30%
- 1 midterm exam 30%
- final exam 40%

Homework policy

- welcome to work together (indicate collaborators)
- hand in your own assignments
- typeset assignments only (no handwritten submissions accepted)
- 3 late days total – yours to budget (weekend $\equiv$ 1 weekday)

Resources

Website

- Main site: `https://friedlander.io/19T2-406`
- Piazza: for discussions and solutions
  `piazza.com/ubc.ca/winterterm22020/cpsc406/home`

Office Hours

- TBD. Keep an eye on the webpage.

Mathematical optimization

Mathematical optimization

$$\underset{x}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad x \in \mathcal{C} \subseteq \mathbb{R}^n,$$

- $x = (x_1, \ldots, x_n)$ is vector of optimization variables
- $f : \mathbf{R}^n \to \mathbf{R}$ is the objective (or cost) function
- $\mathcal{C}$ is a constraint (feasible) set, eg,

$$\mathcal{C} = \{x \in \mathbf{R}^n \mid c_i(x) \leq b_i,\ i = 1, \ldots, m\}$$

  where $c_i \in \mathbf{R}^n \to \mathbf{R}$ are constraint functions

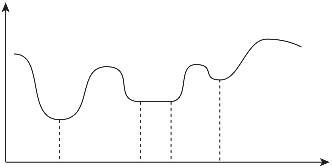
Optimal solution x^* has the **smallest** value of f among all vectors $x \in \mathcal{C}$, eg,

$$f(x^*) \leq f(x) \quad \text{for all } x \text{ such that} \quad c_i(x) \leq b_i$$

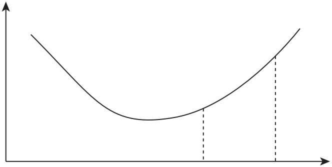
Problems don't necessarily appear this way. We may need to apply a transformation to reduce them to a standard form.

Illustrative examples

$$\underset{x}{\text{minimize}} \quad f(x)$$

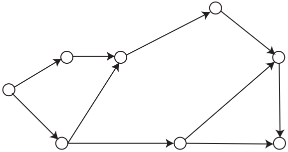


$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & l \le x \le u \end{array}$$



minimize cost of flow

subject to network capacity
flow conservation



Varieties of optimization problems

- continuous vs discrete
- unconstrained vs constrained
- global vs local
- stochastic vs deterministic

Our domain is in blue.

Solving optimization problems

General optimization problems (no structure)

- difficult to solve
- most methods involve some compromise, eg, long computation time vs finding exact/correct solution
- not usually clear which methods work best

Ideally recognize the type/class of problem – some are "easy"

- least-squares
- linear programs
- convex problems

Structure is often hidden

- network flows
- dynamic programs

Linear least squares

$$\underset{x}{\text{minimize}}\ \|Ax - b\|_2 \qquad \text{with} \qquad A \text{ and } m\text{-by-}n \text{ matrix}$$

Formulation

$$\text{Data: } A = [a_1\ a_2\ \cdots a_n],\ a_i \in \mathbb{R}^m$$
$$b = (b_1\ b_2\ \cdots b_m),\ b \in \mathbb{R}^m$$

Model 1: $b \approx a_1x_1 + a_2x_2 + \cdots + a_nx_n$

Model 2: $b \approx Ax + \epsilon$, where ϵ is "noise"

Solution

- $x^* = (A^TA)^{-1}A^Tb$ — rarely this easy!
- reliable and efficient algorithms and software — direct and indirect solvers
- standard techniques to increase flexibility — eg, incorporate prior information
- easy to recognize

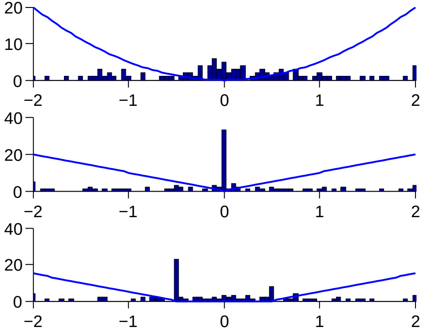
Different fitting criteria

$$\underset{x}{\text{minimize}} \;\; \|Ax - b\|$$

least-squares approximation:

$\|r\|_2^2 = r_1^2 + \cdots + r_m^2$

closed form solution x^* solves $A^T A x = A^T b$



one-norm approximation:

$\|r\|_1 = |r_1| + \cdots + |r_m|$

deadzone approximation:

$$\|r\| = \sum_i^m \max\{|r_i| - \alpha, 0\}$$

Linear Programming (LP)

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & Ax = b,\ x \geq 0 \end{array}$$

Formulation

- equality and inequality constraints
- both objective and constraint functions are linear

Solving LPs

- generally no analytic solution exists
- robust and efficient algorithms and software (eg, simplex method)

Using LPs

- arises in unexpected places
- many problems can be turned into LPs

Example: Scheduling

Objective: schedule weekly night-shifts for nurse staff at minimum cost

Constraint:

1. every nurse must work 5 straight nights
2. on night $j = 1, \ldots, 7$, d_j nurses are required

Variables: not obvious! First attempt:

- y_j nurses work on night j
- $\underset{y}{\text{minimize}} \quad \sum_j y_j \quad$ subject to $\quad y_j \geq d_j,\ j = 1, \ldots, 7$

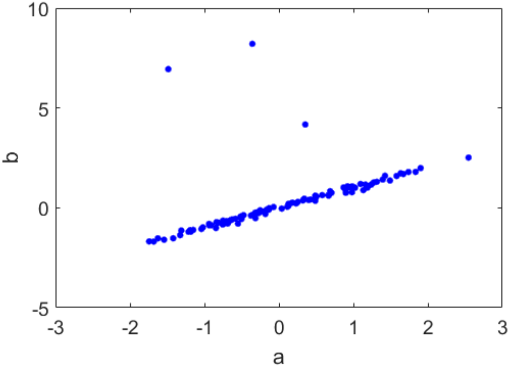
Second attempt: Let x_j be no. of nurses **starting** their 5 day shift on day j:

$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{subject to} & x_1 + x_4 + x_5 + x_6 + x_7 \geq d_1 \\
& x_1 + x_2 + x_5 + x_6 + x_7 \geq d_2 \\
& x_1 + x_2 + x_3 + x_6 + x_7 \geq d_3 \\
& x_1 + x_2 + x_3 + x_4 + x_7 \geq d_4 \\
& x_1 + x_2 + x_3 + x_4 + x_5 \geq d_5 \\
& x_2 + x_3 + x_4 + x_5 + x_6 \geq d_6 \\
& x_3 + x_4 + x_5 + x_6 + x_7 \geq d_7 \\
& x_1, \ldots, x_7 \geq 0
\end{array}
$$

- note the constraint structure. This is almost always true of practical LPs
- we may want to restrict x_j to be integer. This is a much harder problem!

Example: Robust data fitting

Almost linear data with a few outliers

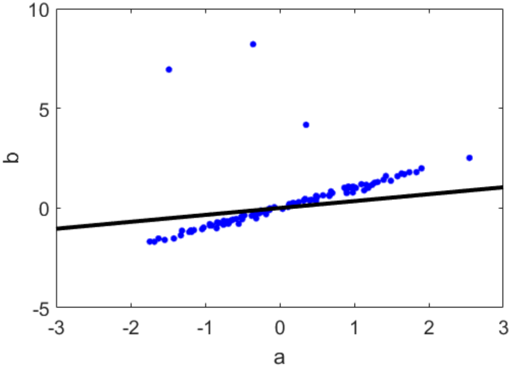


Example: Robust data fitting

Idea 1: Solve least squares

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{n} (a_i x - b_i)^2$$

Plot $f(a) = ax$ (black line)



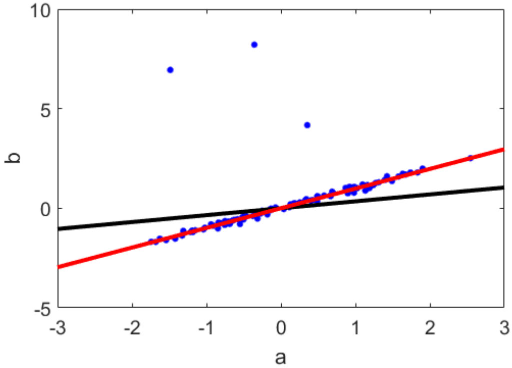
Terrible!

Example: Robust data fitting

Idea 2: Solve with different error term

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{n} |a_i x - b_i|$$

Plot $f(a) = ax$ (red line)



So much better!

Example: Robust data fitting

In higher dimensions,

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_1 = \sum_{i=1}^{m} |a_i^T x - b_i|$$

LP reformulation.

$$\begin{array}{ll} \underset{x,v}{\text{minimize}} & \sum_{i=1}^{m} v_i \\ \text{subject to} & -v_i \le a_i^T x - b_i \le v_i, \ i = 1, \ldots, m \end{array}$$

Rough plan

The (rough) plan

- ($\sim$ 3 weeks) Advanced linear algebra, simple optimization problems
- ($\sim$ 3 weeks) Convex optimization, important methods

Midterm (good luck!)

- ($\sim$ 3 weeks) Applications, extensions
- ($\sim$ 3 weeks) Linear programming, duality

Final (good luck!)

Any questions?