Table of Contents

The following notes use the color scheme:
Definition
Theorem
Fact
Example
Method/step-by-step recipe
Proof
Super-duper important things

Introduction to Optimizations

Definition of an optimization problem

An *optimization problem* has the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant b_i, \quad i = 1, \ldots, m \end{array}$$

where:

- $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the *optimization variable*;
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*;
- $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ are the *constraint functions*;
- $b_1, ..., b_m$ are the limits (bounds) for the constraints.

A vector x^* is *optimal* or is a *solution* if

$$\forall z \in \mathbb{R}^n \text{ satisfying the constraints}, \quad f_0(x^*) \leqslant f_0(z).$$

Classes of optimization problems

Optimization problems are bundled into classes based on the form of their objective (f_0) and constraint (f_i) functions. Here are some below.

Note: in optimization jargon, a "program" is an alias for an "optimization problem".

Linear program

$f_0, ..., f_m$ are linear functions (i.e. both objective and constraints). Recall:

$$f(x) : \mathbb{R}^n \to \mathbb{R} \text{ is linear} \Leftrightarrow \forall x, y \in \mathbb{R}^n, \ \forall \alpha \in \quad f(\alpha x + \beta y) = \alpha f(x) + \beta f(y).$$

Nonlinear program

An optimization problem is nonlinear if it is not linear. Of course, there are many classes of nonlinear programs. A general nonlinear program can be applied to basically any optimization problem, however:

- It may take a very long time to solve;
- It may not find the (global) solution;
- It may be very difficult to solve!

Convex program

...and convex optimization is one of them!

A **convex optimization problem** is one in which $f_0, \ldots, f_m$ are convex:
$f(x) : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if
$\forall x, y \in \mathbb{R}^n, \forall \alpha, \beta \in \mathbb{R}_+ \text{ s.t. } \alpha + \beta = 1 \quad \Rightarrow \quad f(\alpha x + \beta y) \leqslant \alpha f(x) + \beta f(y).$

Unlike general nonlinear programs, convex (and linear) programs can be solved *efficiently* and *reliably*, making them ideally suited for real world (and even high stakes) applications!

In fact, a convex program is a common parent of two well know, reliable and efficient optimization classes: least squares and linear programs.

Least-Squared Problems

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

Least-squares programming is a *mature technology*: least-squares problems can be solved efficiently and reliably. Many software packages exist to do this.

Least-squares properties:

- The analytical solution for $A \in \mathbb{R}^{k \times n}$ tall (i.e. $k > n$) and full rank (i.e. full column rank, $\text{null}(A) = \varnothing$): $x^* = (A^T A)^{-1} A^T b$;
  - NB: good solvers don't do literally this operation, though!
- Compute time $\propto n^2 k$.

How to know if your problem is a least-squares problem? **Simple, just one question**:

- Q: is the objective function the 2-norm squared of an affine function of x (and you have no constraints)?
  - A: Yes: it's a least-squares problem!
  - A: No: it's not a least squares problem!

Linear Programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leqslant b_i, \quad i = 1, \ldots, m \end{array}$$

Linear programming is, like least squares, a mature technology. Existing algorithms are (almost) as reliable as least squares.

Some linear programming properties:

- No analytical solution (except for trivial cases)!
- Computation time $\propto n^2 m$ if $m \geqslant n$ (more constraints than optimization variables);
  - Less with structure;
  - NB: this is the same computational time as least-squares!
  - It is good news that it's $n^2 m$ and not $m^2 n$ (since generally there can be very many constraints on a relatively small optimization variable)