Optimization Models

EECS 127 / EECS 227AT

Laurent El Ghaoui

EECS department
UC Berkeley

Fall 2018

LECTURE 1 (Short version)

Introduction

Constrained optimization
is the art of compromise
between conflicting objectives.
This is what design is all about.

William A. Dembski

Outline

1. Introduction
   - What is optimization?
   - Examples
2. Optimization problems
   - Definitions
   - Convex problems
   - Non-convex problems
3. Course outline

Introduction

A standard form of optimization

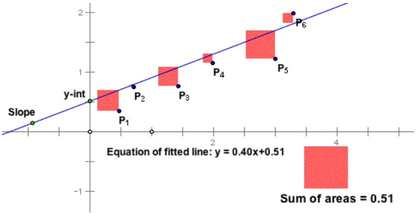
$$
\begin{aligned}
p^* = \min_x \quad & f_0(x) \\
\text{subject to:} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m,
\end{aligned}
$$

where

- vector $x \in \mathbb{R}^n$ is the *decision variable*;
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective* function, or *cost*;
- $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, represent the *constraints*;
- p^* is the *optimal value*.

Examples

Least-squares regression



$$\min_x \sum_{i=1}^{m} (y_i - x^\top z^{(i)})^2$$

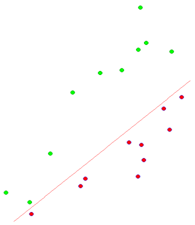
where

- $z^{(i)} \in \mathbb{R}^n$, $i = 1, \ldots, n$ are data points;
- $y \in \mathbb{R}^m$ is a "response" vector;
- $x^\top z$ is the scalar product $z_1 x_1 + \ldots + z_n x_n$ between the two vectors $x, z \in \mathbb{R}^n$.

- Many variants (with *e.g.*, constraints) exist (more on this later).
- Once x is found, allows to predict the output $\hat{y}$ corresponding to a new data point z: $\hat{y} = x^\top z$.
- Perhaps the most popular optimization problem.

Examples

Linear classification



Support Vector Machine (SVM):

$$\min_{x,b} \sum_{i=1}^{m} \max(0, 1 - y_i(x^\top z^{(i)} + b))$$

where

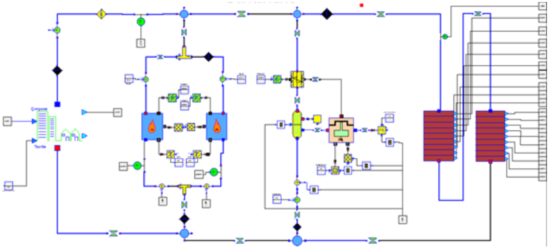
- $z^{(i)} \in \mathbb{R}^n$, $i = 1, \ldots, n$ are data points;
- $y \in \{-1, 1\}^m$ is a *binary* response vector;
- $x^\top z + b = 0$ defines a "separating hyperplane" in data space.

- Many variants exist (more on this later).
- Once x, b are found, we can predict the binary output $\hat{y}$ corresponding to a new data point z: $\hat{y} = \mathbf{sign}(x^\top z + b)$.
- Very useful for classifying data (*e.g.*, text documents).

Examples

Energy production

An energy production center is composed by a 1.3 MWe/1.6 MWth combined heat and power (CHP) gas engine, two 1.4 MW gas boilers and two $105m^3$ thermal storage containers, with a capacity of 4.5 MWh of heat. The center serves about 700 homes in a large city.



Examples

Optimal intra-day energy generation

Goals: minimize the total cost of running the plant over the day, while meeting demand constraints.

- Variables: engine and boilers production levels at each hour of the day
- Objective function to minimize: production costs (gas, maintenance) minus income (electricity sold).
- Constraints:
  - devices production upper bounds and non negative variables;
  - heat storage capacity;
  - demand satisfaction constraint;
  - process equations linking the gaz consumption and heat/electricity production.

Examples

Energy production: challenges

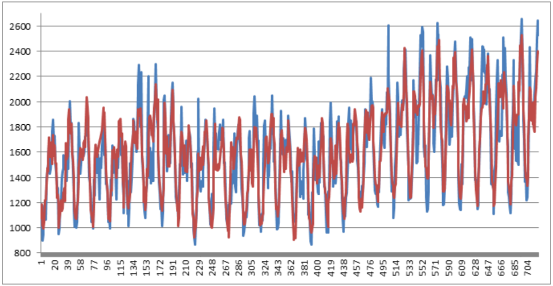


Figure: Real and forecasted demand.

- Demand is uncertain, there are prediction errors.
- Process is complex, highly non-linear.
- There are Boolean decision variables (on-off).

Examples

Pricing items for online retail

A large online retailer seeks to optimize the prices of a large catalog of items, based on estimated demand for the items.

- Variables: $p \in \mathbb{R}^n$ contains the prices of the items.
- Objective: maximize revenue.
- Constraints:
  - Upper and lower bounds $\underline{p}, \overline{p}$ on the prices (*e.g.*, MSRP).
  - Lower bound on profit.

Examples

Pricing items for online retail: elastic demand model

Model demand as a linear (in economic terms, "elastic") function:

$$D_i(p) = b_i - g_i(p_i - p_{0,i}), \quad i = 1, \ldots, n,$$

where

- b_i is a "baseline" demand for item i;
- $g_i > 0$ reflects the fact that demand decreases for price increases;
- p_0 is a set of reference prices.

- Revenue function: $R(p) = p^\top D(p)$.
- Profit function: $P(p) = (p - c)^\top D(p)$, where c is cost (to the retailer) of item.

Examples

Pricing items for online retail: model

$$\max_p \ R(p) \ : \ P(p) \geq P_{\min}, \ \ \underline{p} \leq p \leq \overline{p}.$$

Challenges:

- $n \simeq 10^7$ (some items are bundled), and problem has to be solved at that scale in real-time.
- Demand is uncertain.
- Sometimes there is an added constraint on the total number of price changes.

Optimization problems

A standard form of optimization

We shall mainly deal with optimization problems that can be written in the following standard form:

$$
\begin{aligned}
p^* = \min_x \quad & f_0(x) \\
\text{subject to:} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m,
\end{aligned}
\tag{1}
$$

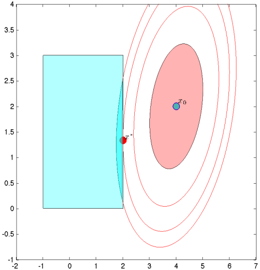
where

- Vector $x \in \mathbb{R}^n$ is the *decision variable*;
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective* function, or *cost*;
- $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, represent the *constraints*;
- p^* is the *optimal value*.

Nomenclature

A toy optimization problem

$$\begin{array}{ll} \min\limits_{x} & 0.9x_1^2 - 0.4x_1x_2 - 0.6x_2^2 - 6.4x_1 - 0.8x_2 \\ \text{s.t.} & -1 \le x_1 \le 2, \;\; 0 \le x_2 \le 3. \end{array}$$



- *Feasible set* in light blue.
- 0.1-*suboptimal set* in darker blue.
- *Unconstrained minimizer*: x_0; optimal point: x^*.
- *Level sets* of objective function in red lines.
- A *sub-level set* in red fill.

What is a solution?

- In an optimization problem, we are usually interested in computing
  - the optimal value p^* of the objective function;
  - very often we are interested in a corresponding *minimizer* x^*, which is a vector that achieves the optimal value, and satisfies the constraints.
- We say that the optimal value p^* is *attained* if there exist a feasible x^* such that

$$f_0(x^*) = p^*.$$

- In the optimization problem, the optimal value $p^* = -10.2667$ is attained by the optimal solution

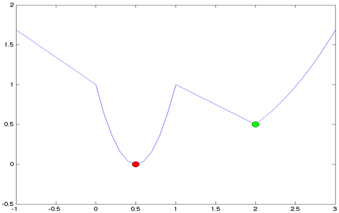
$$x_1^* = 2, \quad x_2^* = 1.3333.$$

Local vs. global optimal points

- A point z is *locally optimal* for problem (1) if there exist a value $R > 0$ such that z is optimal for problem

$$\min_x \; f_0(x) \;\; \text{s.t.} \;\; f_i(x) \leq 0, \;\; i = 1, \ldots, m, \;\; |x_i - z_i| \leq R, \;\; i = 1, \ldots, n.$$

- A local minimizer x minimizes f_0, but only compared to nearby points on the feasible set. The value of the objective function at that point is *not* necessarily the (global) optimal value of the problem. Locally optimal points might be of no practical interest to the user.



Local (green) vs. global (red) minima. The optimal set is the singleton $\mathcal{X}_{\text{opt}} = \{0.5\}$. The point $x = 2$ is a local minimum.

Convex problems

- Convex optimization problems are problems of the form (1), where the objective and constraint functions have the special property of *convexity*.
- Roughly speaking, a convex function has a "bowl-shaped" graph.

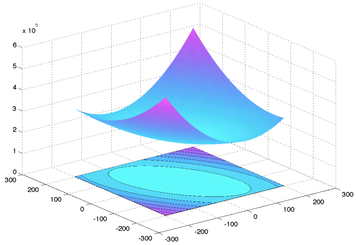


Figure: Convex function.

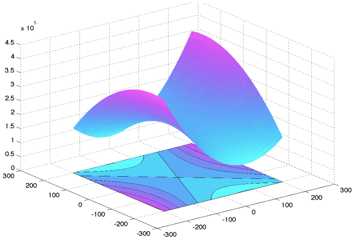
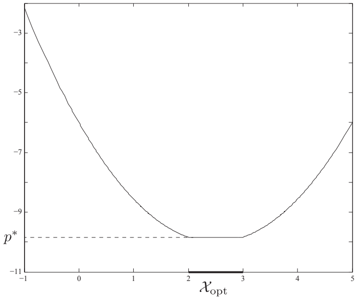


Figure: Non-convex function.

Convex problems

- For a convex function, any local minimum is global. In the example below, the minimizer is not unique, and the optimal set is the interval $\mathcal{X}_{\rm opt} = [2, 3]$. Every point in the interval achieves the global minimum value $p^* = -9.84$.



- Not all convex problems are easy to solve, but many of them are indeed computationally tractable. One key feature of convex problems is that all local minima are actually global.

Special convex models

- We shall deal specifically with convex optimization problems with special structure, such as:
  - Least-Squares (LS)
  - Linear Programs (LP)
  - Convex Quadratic Programs (QP)
  - Second-order cone programs (SOCP)
- For such specific models, very efficient solution algorithms exist, together with user-friendly prototyping software (such as CVX, Yalmip, Mosek, etc.)

Non-convex problems

- *Boolean/integer optimization:* some variables are constrained to be Boolean or integers. Convex optimization can be used for getting (sometimes) good approximations.
- *Cardinality-constrained problems:* we seek to bound the number of non-zero elements in a vector variable. Convex optimization can be used for getting good approximations.
- *Non-linear programming:* usually non-convex problems with differentiable objective and functions. Algorithms provide only local minima.

Most (but not all) non-convex problems are hard!

Course scope

What this course is for:

- Learning to model and efficiently solve problems arising in Engineering, Management, Control, Finance, Machine Learning, etc.
- Learning to prototype small- to medium- sized problems on numerical computing platforms.
- Learning basics of applied linear algebra, convex optimization.

What this course is NOT:

- A course on mathematical convex analysis.
- A course on details of optimization algorithms.

Course outline

- Linear algebra models
  - Vectors, projection theorem, matrices, symmetric matrices.
  - Linear equations, least-squares and minimum-norm problems.
  - Singular value decomposition (SVD) and related optimization problems.
- Convex optimization models
  - Convex sets, convex functions, convex problems.
  - Optimality conditions, duality.
  - Special convex models: LP, QP, SOCP.
- Applications
  - Machine learning.
  - Control.
  - Finance.
  - Engineering design.