Introduction

A standard form of optimization

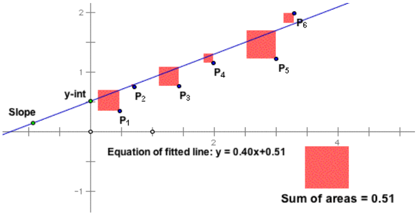
$$
\begin{aligned}
p^* = \min_x \quad & f_0(x) \\
\text{subject to:} \quad & f_i(x) \le 0, \quad i = 1, \dots, m,
\end{aligned}
$$

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*;
- $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$, represent the *constraints*;
- p^* is the *optimal value*.

Examples

Least-squares regression



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

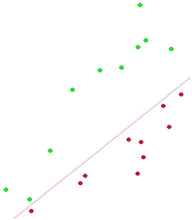
where

- $z^{(i)} \in \mathbb{R}^n$, $i = 1, \ldots, n$ 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):

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

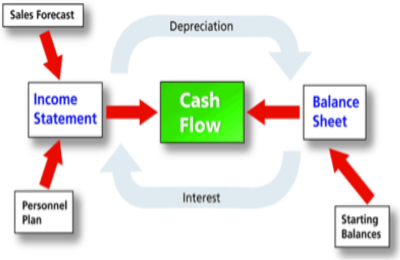
where

- $z^{(i)} \in \mathbb{R}^n$, $i = 1, \ldots, n$ 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: $\hat{y} = \textbf{sign}(x^\top z + b)$.
- Very useful for classifying data (*e.g.*, text documents).

Examples

Cash-flow management



A company needs to choose between three financial instruments to cover its liabilities over a six-months period into the future:

- A line of credit of maximum amount \$100k, with interest rate 1% per month;
- In any of the first 3 months it can issue 90-day commercial paper (a type of unsecured debt) bearing a total interest of 2% for the 3-month period;
- Excess funds (cash) can be invested at 0.3% per month.

Examples

Cash-flow management: variables and decision problem

Variables:

- Balance on the credit line x_i for month $i = 1, 2, 3, 4, 5$.
- Amount y_i of commercial paper issued ($i = 1, 2, 3$).
- Excess funds z_i for month $i = 1, 2, 3, 4, 5$.
- z_6, the company's wealth at the end of the 6-month period.

With these variables we have to meet certain cash-flow requirements each month.

Decision problem:

$$\text{maximize } z_6 \text{ subject to } \begin{cases} \text{Bounds on variables,} \\ \text{Cash-flow balance equations.} \end{cases}$$

Examples

Cash-flow management: optimization model

$$
\begin{aligned}
\max_{x,y,z} \quad & z_6 \\
\text{s.t.} \quad & x_1 + y_1 - z_1 = 150, \\
& x_2 + y_2 - 1.01x_1 + 1.003z_1 - z_2 = 100, \\
& x_3 + y_3 - 1.01x_2 + 1.003z_2 - z_3 = -200, \\
& x_4 - 1.02y_1 - 1.01x_3 + 1.003z_3 - z_4 = 200, \\
& x_5 - 1.02y_2 - 1.01x_4 + 1.003z_4 - z_5 = -50, \\
& -1.02y_3 - 1.01x_5 + 1.003z_5 - z_6 = -300, \\
& 0 \le x \le 100, \quad y \ge 0, \quad z \ge 0
\end{aligned}
$$

The right-hand side contains the liabilities that must be met.

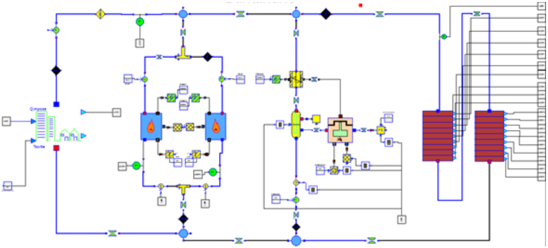
Challenges:

- Future liabilities are uncertain.
- Some instruments may have varying (thus, uncertain) interest rates.

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

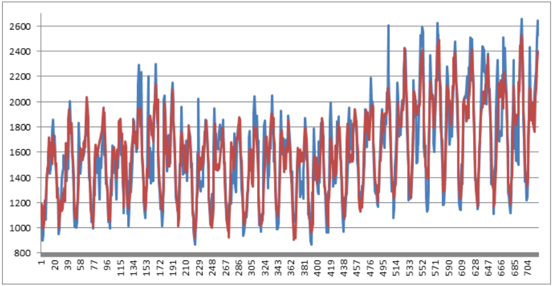


Figure: Real and forecasted demand.

- 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:

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

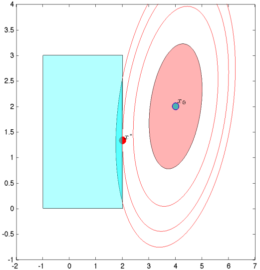
where

- 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.

Problems with equality constraints

- Sometimes the problem may present explicit *equality* constraints, along with inequality ones, that is

$$\begin{aligned} p^* = \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \le 0, \quad i = 1, \ldots, m, \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{aligned}$$

  where h_i's are given functions.

- Formally, however, we may reduce the above problem to a standard form with inequality constraints only, by representing each equality constraint via a pair of inequalities. That is, we represent

$$h_i(x) = 0 \quad \Leftrightarrow \quad h_i(x) \le 0, \; h_i(x) \ge 0.$$

Problems with set constraints

- Sometimes, the constraints of the problem are described abstractly via a set-membership condition of the form $x \in \mathcal{X}$, for some subset $\mathcal{X}$ of $\mathbb{R}^n$.
- The corresponding notation is

$$p^* = \min_{x \in \mathcal{X}} f_0(x),$$

or, equivalently,

$$\begin{aligned} p^* = \min_{x} \quad & f_0(x) \\ \text{s.t.} \quad & x \in \mathcal{X}. \end{aligned}$$

Problems in maximization form

- Some optimization problems come in the form of maximization (instead of minimization) of an objective function, i.e.,

$$p^* = \max_{x \in \mathcal{X}} g_0(x).$$

- Such problems can be readily recast in standard minimization form by observing that, for any g_0, it holds that

$$\max_{x \in \mathcal{X}} g_0(x) = -\min_{x \in \mathcal{X}} -g_0(x).$$

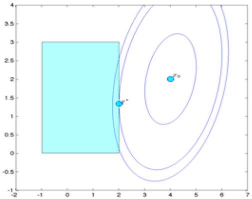
- Therefore, the problem in maximization form can be reformulated as one in minimization form as

$$-p^* = \min_{x \in \mathcal{X}} f_0(x),$$

  where $f_0 = -g_0$.

Feasible set

- The *feasible set* of problem (1) is defined as
$$\mathcal{X} = \{x \in \mathbb{R}^n \ \text{s.t.}\ f_i(x) \leq 0, \ \ i = 1, \ldots, m\}.$$
- A point x is said to be *feasible* for problem (1) if it belongs to the feasible set $\mathcal{X}$, that is, if it satisfies the constraints.
- The feasible set may be empty, if the constraints cannot be satisfied simultaneously. In this case the problem is said to be *infeasible*.
- We take the convention that the optimal value is $p^* = +\infty$ for infeasible minimization problems, while $p^* = -\infty$ for infeasible maximization problems.



In the toy optimization problem, the feasible set is the "box" in $\mathbb{R}^2$, described by
$$-1 \leq x_1 \leq 2, \ \ 0 \leq x_2 \leq 3.$$

Feasibility problems

- Sometimes an objective function is not provided. This means that we are just interested in finding a feasible point, or determine that the problem is infeasible.
- By convention, we set f_0 to be a constant in that case, to reflect the fact that we are indifferent to the choice of a point x, as long as it is feasible.
- For problems in the standard form (1), solving a feasibility problem is equivalent to finding a point that solves the system of inequalities $f_i(x) \leq 0$, $i = 1, \ldots, m$.
- Even in a problem with an objective, determining if there is a feasible solution or not, unambiguously, is important.

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

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

Optimal set

- The *optimal set*, or *set of solutions*, of problem (1) is defined as the set of feasible points for which the objective function achieves the optimal value:

$$\mathcal{X}_{\rm opt} = \{x \in \mathbb{R}^n \text{ s.t. } f_0(x) = p^*, \quad f_i(x) \le 0, \quad i = 1, \dots, m\}.$$

- A standard notation for the optimal set is via the arg min notation:

$$\mathcal{X}_{\rm opt} = \arg\min_{x \in \mathcal{X}} f_0(x) \longrightarrow \text{obj. function?}$$

- A point x is said to be *optimal* if it belongs to the optimal set.

When is the optimal set empty?

- Optimal points may not exist, and the optimal set may be empty. This can be due to two reasons. One is that the problem is infeasible, i.e., $\mathcal{X}$ itself is empty (there is no point that satisfies the constraints).
- Another, more subtle, situation arises when $\mathcal{X}$ is nonempty, but the optimal value is only reached in the limit. For example, the problem

$$p^* = \min_x \; e^{-x}$$

has no optimal points, since the optimal value $p^* = 0$ is only reached in the limit, for $x \to +\infty$.

- Another example arises when the constraints include strict inequalities, for example with the problem

$$p^* = \min_x \; x \quad \text{s.t.} \quad 0 < x \leq 1.$$

In this case, $p^* = 0$ but this optimal value is not attained by any x that satisfies the constraints.

Sub-optimality

- We say that a point x is ϵ-suboptimal for problem (1) if it is feasible, and satisfies

$$p^* \leq f_0(x) \leq p^* + \epsilon.$$

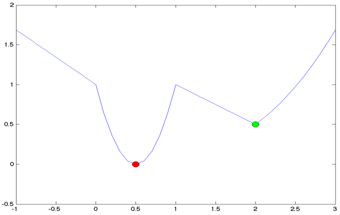
- In other words, x is ϵ-close to achieving the best value p^*. Usually, numerical algorithms are only able to compute suboptimal solutions, and never reach true optimality.

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.

Tractable vs. non-tractable problems

- Some problem classes, such as finding a solution to a finite set of linear equalities or inequalities, can be solved numerically in an efficient and reliable way.
- For some other classes of problems, no reliable efficient solution algorithm is known.
- Without entering a discussion on the *computational complexity* of optimization problems, we shall here refer to as "tractable" all those optimization models for which a globally optimal solution can be found numerically in a reliable way (i.e., always, in any problem instance), with a computational effort that grows gracefully with the *size* of the problem (informally, the size of the problem is measured by the number of decision variables and/or constraints in the model).
- Other problems are known to be "hard," and for yet other problems the computational complexity is unknown.

Tractable vs. non-tractable problems

- The focus of this course is on *tractable models*, and a key message is that models that can be formulated in the form of linear algebra problems, or in *convex* form, are typically tractable. Further, if a convex model has some special structure, then solutions can typically be found using existing and very reliable numerical solvers, such as CVX, Yalmip, etc.
- Tractability is often *not* a property of the problem itself, but a property of our formulation and modeling of the problem. A problem that may seem hard under a certain formulation may well become tractable if we put some more effort and intelligence in the modeling phase.
- One of the goals of this course is to teach the "art" of manipulating problems so to model them in a tractable form. Clearly, this is not always possible: some problems are just hard, no matter how much effort we put in trying to manipulate them. One example is the *travelling salesman* problem.

Problem transformations

The optimization formalism in (1) is extremely flexible and allows for many transformations, which may help to cast a given problem in a tractable formulation.

For example, the optimization problem

$$\max_x \; x_1 x_2^3 x_3 \quad \text{s.t.} \quad x_i \geq 0, \quad i = 1, 2, 3, \quad x_1 x_2 \leq 2, \quad x_2^2 x_3 \leq 1$$

can be equivalently written, after taking the log of the objective, in terms of the new variables $z_i = \log x_i$, $i = 1, 2, 3$, as

$$\max_z \; z_1 + 3z_2 + z_3 \quad \text{s.t.} \quad z_1 + z_2 \leq \log 2, \quad 2z_2 + z_3 \leq 0.$$

The advantage is that now the objective and constraint functions are all linear.

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.

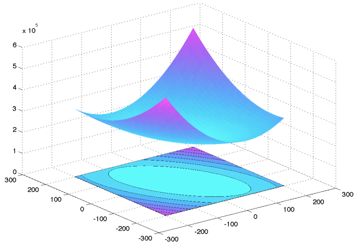


Figure: Convex function.

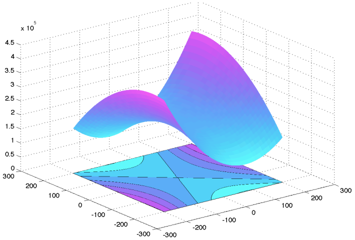
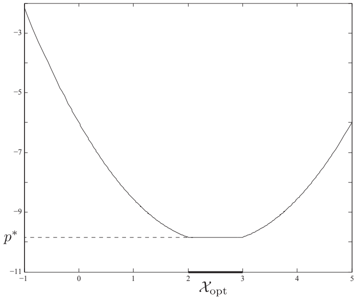


Figure: Non-convex function.

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)
  - Geometric programs (GP)
  - Second-order cone programs (SOCP)
  - Semi-definite programs (SDP).
- For such specific models, very efficient solution algorithms exist, together with user-friendly prototyping software (such as CVX, Yalmip, Mosek, etc.)

Non-convex problems

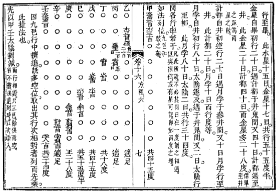
- *Boolean/integer optimization:* some variables are constrained to be Boolean or integers. Convex optimization can be used for getting (sometimes) good approximations.
- *Cardinality-constrained problems:* we seek to bound the number of non-zero elements in a vector variable. Convex optimization can 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!

Optimization in society

History

Early stages: birth of linear algebra

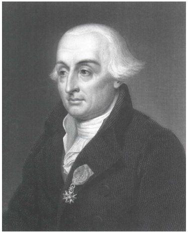
- The roots of optimization can perhaps be traced back to the earliest known appearance of a system of linear equations in ancient China.
- Indeed, the art termed *fangcheng* (often translated as "rectangular arrays") was used as early as 300 BC to solve practical problems which amounted to linear systems.
- Algorithms identical to Gauss elimination for solving such systems appear in Chapter 8 of the treatise *Nine Chapters on the Mathematical Art*, dated around 200 BC.
- A 9×9 matrix found in the treatise, as printed in the 1700's.



History

Optimization as a theoretical tool

- Around mid 1700, Euler formalized the *principle of least action*, according to which the motion of natural systems could be described as a minimization problem involving a certain cost function called "energy."
- In the 1800's, Gauss developed a method for solving least-squares problems. He used the method to accurately predict the trajectory of the planetoid Ceres.
- The Italian mathematician Giuseppe Luigi Lagrangia (1736–1813), also known as Lagrange, associated his name with the notion of duality.



History

Advent of numerical linear algebra

- With computers becoming available in the late 40s, the field of numerical linear algebra was ready to take off, motivated in no small part by the cold war effort. Early contributors include Von Neumann, Wilkinson, Householder, and Givens.
- Optimization played a key role in the development of linear algebra (e.g., Simplex Algorithm).
- In the 70s: Efficient packages written in FORTRAN, such as LINPACK and LAPACK, embodied the progresses on the algorithms and became available in the 80s.
- 80s-90s: scientific computing platforms, such as Matlab, Scilab, Octave, R, etc. Hid the FORTRAN packages developed earlier behind a user-friendly interface. Linear algebra became a commodity technology.
- Current research effort in the field of numerical linear algebra involves the solution of extremely large problems (e.g., Google PageRank).

History

Advent of linear and quadratic programming

- The LP model was introduced by George Dantzig in the 40s, in the context of logistical problems arising in military operations. Extending the scope of linear algebra (linear equalities) to inequalities, his efforts led to the famous *simplex algorithm* for solving such systems.
- QP models were introduced in the 50s by H. Markowitz (who was then a colleague of Dantzig at the RAND Corporation), to model investment problems. Markowitz won the Nobel prize in Economics in 1990, mainly for this work.
- In the 60s-70s, a lot of attention was devoted to non-linear optimization problems. Methods to find local minima were proposed. These methods could fail to find global minima, or even to converge. Hence the notion formed that, while linear optimization was numerically tractable, non-linear optimization was not, in general. This had concrete practical consequences: linear programming solvers could be reliably used for day-to-day operations (for example, for airline crew management), but non-linear solvers needed an expert to baby-sit them.

History

Advent of convex programming

- Most of the research in optimization in the United States in the 60s-80s focussed on nonlinear optimization algorithms, and contextual applications. The availability of large computers made that research possible and practical.
- In the Soviet Union at that time, the focus was more towards optimization theory. Soviet researchers asked the following question: what makes linear programs easy? Is it really linearity of the objective and constraint functions, or some other, more general, structure?
- In the late 80s, two researchers in the former Soviet Union, Yurii Nesterov and Arkadi Nemirovski, discovered that a key property that makes an optimization problem "easy" is not linearity, but actually *convexity*. They introduced so-called *interior-point methods* for solving (wide classes of) convex problems efficiently.
- Since the seminal work of Nesterov and Nemirovski, convex optimization has emerged as a powerful tool that generalizes linear algebra and linear programming: it has similar characteristics of reliability (it always converges to the global minimum) and tractability (it does so in reasonable time).

History

Present

- Strong interest in applying optimization techniques in a variety of fields, ranging from engineering design, statistics and machine learning, to finance and structural mechanics.
- As with linear algebra, recent interfaces to convex optimization solvers, such as CVX or YALMIP now make it extremely easy to prototype models for moderately-sized problems.
- In research, motivated by the advent of very large datasets, a strong effort is currently made towards enabling solution of extremely large-scale convex problems arising in machine learning, image processing and so on. The initial focus of the 1990's on interior-point methods has been replaced with a revisitation and development of earlier algorithms (mainly, the so-called "first-order" algorithms, developed in the 50's), which involve a large amount of cheap iterations.

Course material and references

- *Lecture slides:* Posted on bCourse.
- *Handouts:* Posted when available on bCourse.
- *Textbooks & printed material:*
  - G. Calafiore and L. El Ghaoui, *Optimization Models.* Cambridge University Press, 2014.
    This introductory textbook follows closely the stucture and topics of this course. *Not* required.
  - The livebook at

    `http://livebooklabs.com/keeppies/c5a5868ce26b8125`

    covers the same material, but in less detail. The livebook allows you (after a free registration step) to post questions.
- *Software:* we will rely on matlab or Python and CVX (toolbox for convex optimization), freely available at `cvxr.com`.