6.S098: Introduction to Convex Optimization

Theo Diamandis

January 16, 2023

1 Mathematical optimization

A *mathematical optimization problem* has the form

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

We call the vector $x \in \mathbf{R}^n$ the optimization variable, $f_0 : \mathbf{R}^n \to \mathbf{R}$ the objective function, and $f_i : \mathbf{R}^n \to \mathbf{R}$ the constraint functions. A *solution* (or *optimal* point) $x^\star$ has the smallest value f_0 among all the vectors that satisfy the constraints. These constraints often have the interpretation of budgets, with b_i being the amount of resource i that is available. Note that this model includes maximization problems by simply negating the objective function and equality constraints by adding two inequality constraints.

We typically categorize optimization problems by the mathematical properties of their objective function and constraints. In this course, we will focus on *convex optimization problems*: problems in which the objective function and all the constraint functions are convex. A convex function satisfies the inequality

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \tag{1}$$

for all x, $y \in \mathbf{R}^n$ and all α, $\beta \in \mathbf{R}_+$ with $\alpha + \beta = 1$. This inequality tells us that the function looks bowl-shaped or, more formally, that the function lies below all of its chords. (We will see equivalent definitions of convexity in the next lecture.) A huge number of practical problems can be formulated as convex optimization problems, and we can (usually) solve these problems efficiently with modern open source and commercial solvers.

1.1 Examples

Portfolio optimization.

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, max. risk (variance)
- objective: risk adjusted return or Sharpe ratio

Silicon phontonic design.

- variables: material at each point in the device
- constraints: laws of physics (electromagnetic wave equations), available materials
- objective: maximize efficiency (*e.g.,* focusing efficiency of a lens)

Data fitting.

- variables: parameters of a model
- constraints: prior information, parameter limits
- objective: misfit or prediction error, plus regularization term

Truss design.

- variables: cross sectional area and location of each bar
- constraints: laws of physics (statics), bounds on areas, total weight
- objective: minimize elastic stored energy

Maximum flow through a graph.

- variables: flow through each edge
- constraints: flow conservation at each node, bounds on flow through each edge
- objective: maximize total flow (from source to sink)

1.2 Solving optimization problems

In general, optimization problems are very difficult to solve. The methods to find a solution or approximate solution all have tradeoffs: either the point returned by the method is not guaranteed to be optimal (or even feasible!), or the runtime of the method is exponential in the worst case. (In TCS terms, these problems are NP-hard.) Fortunately, many very useful optimization problems, including least-squares, linear programming programs, and convex optimization problems, can be solved efficiently. Certain classes of NP-hard optimization problems, like integer linear programming, can be solved efficiently in practice[1] for many problems of interest, but these problems are outside the scope of this course.

We'll start our study of optimization by looking at least-squares and linear programming, two of the most common types of optimization problems solved in practice. Then, we'll return to the common parent, convex optimization.

[1]These problems, for example the scheduling problem for airlines, are solved many times every day. However, the methods have worse scaling, worst-case performance, and generalization than those used for convex optimization problems.

2 Least-squares to convex optimization

2.1 Least-Squares

The least-squares problem has the form

$$\text{minimize} \quad ||Ax - b||_2^2 = \sum_{i=1}^{m} (a_i^T x - b_i)^2.$$

Here $A \in \mathbf{R}^{m \times n}$ (with $m \geq n$) is the data matrix and $x \in \mathbf{R}^n$ is the optimization variable. Variants include regularization terms and weights on the summands. This problem can be solved via a set of linear equations,

$$A^T A x = A^T b,$$

which has the analytical solution $x^\star = (A^T A)^{-1} A^T b$. Geometrically, the solution gives the point $Ax^\star$ that is closest to b, *i.e.,* the residual $Ax^\star - b$ is orthogonal to $\mathbf{range}(A)$. We have efficient algorithms[2] and very good software implementations to solve this problem quickly and reliably. If A is dense, the problem can be solved in $O(n^2 m)$ time[3], and if A is sparse, the problem can often be solved much faster. We consider solving least-squares problems to be a mature technology; these problems can be readily solved by practitioners who do not know the mathematical or algorithmic details.

2.2 Linear programming

A linear program (LP) has the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b, \qquad i = 1, \ldots, m. \end{array}$$

Note that the LP has the form of (1) with f_i linear for $i = 0, \ldots, m$. Recall that a linear function satisfies

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

for all x, $y \in \mathbf{R}^n$ and all α, $\beta \in \mathbf{R}$. (Compare this with the definition of convexity (1).) Unlike least squares programs, linear programs have no analytical formula for the solution. Fortunately, there are efficient algorithms and software implementations to solve these. The first algorithm, the simplex method, dates back to Dantzig in 1947[4] and is still widely used

[2]The most common way to solve least-squares is via the QR factorization; forming the normal equations and then solving them is numerically unstable. See the book by Trefethen and Bau [TBI97] for more.

[3]Some exciting recent developments in randomized numerical linear algebra allow us to solve these problems even faster. See the survey by Martinsson and Tropp [MT20] for more.

[4]While Dantzig popularized linear programming in the West, after first using it for planning problems faced by the US Air Force, the formulation dates back to Kantorovich's work [Kan39] in the Soviet union, which proposed using LPs to organize production about a decade earlier. (A great novel on the failings of planned economies is Red Plenty.) However, it took around 20 years for Kantorovich's work to be published in western literature.

today. Algorithms to solve these programs take roughly $O(n^2m)$ time (with a larger and less well-characterized constant than least-squares) if $m \geq n$. We also consider them a mature technology: there is software that not only quickly solves linear programs but also converts input problems into a standard form.

Linear programs are not as easy to recognize as least-squares problems. In fact, LPs can be used to solve many problems involving nonlinear functions. There are a number of standard tricks to convert problems into a solver-accepted formulation. We will see two examples below, which are variants on the least-squares norm minimization problem.

ℓ_∞ (Chebyshev) norm minimization. Consider the least-squares problem with the squared ℓ_2 norm replaced with the ℓ_∞ norm:

$$\text{minimize} \quad ||Ax - b||_\infty = \max_i |a_i^T x - b_i|.$$

Note that we have $|y| \leq z$ if and only if $-z \leq y \leq z$. We can convert this problem into an LP by introducing a new variable $t \in \mathbf{R}$ that we require to be greater than the absolute value of every element of $Ax - b$. Thus the problem above is equivalent to the LP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \leq a_i^T x - b \leq t, \qquad i = 1, \ldots, m. \end{array}$$

ℓ_1 norm minimization. Consider the least-squares problem with the squared ℓ_2 norm replaced with the ℓ_1 norm:

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

We can convert this problem into an LP by introducing a new variable $t \in \mathbf{R}^m$. The technique is the same as for the ℓ_∞ norm minimization problem above, but now we have a new variable t_i for each element of $Ax - b$. Thus the problem above is equivalent to the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T t \\ \text{subject to} & -t_i \leq a_i^T x - b \leq t_i, \qquad i = 1, \ldots, m. \end{array}$$

2.3 Convex optimization

A convex optimization problem has the form

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

with f_i convex for $i = 0, \ldots, m$. Recall that a function f is convex if

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

for all x, $y \in \mathbf{R}^n$ and all $\lambda \in [0,1]$. Convex optimization includes least-squares and linear programs as special cases (why?).

Like linear programs, there is usually no analytical solution to a convex optimization problem. However, there are efficient algorithms and software to solve moderately sized problems, and a well-established literature helps guide the implementation of custom, large-scale solvers. Very roughly, the computational time for each step in an interior point algorithm is $\max\{n^3, n^2m, F\}$, where F is the cost of evaluating the f_i's and their first and second derivatives. Approximately 20-100 steps of an interior point solver[5] are usually required produce a solution that is accurate to machine precision. This complexity is often only a modest factor slower than least-squares, despite convex optimization having much more expressive power. Convex optimization is not quite a technology yet (it is for some problem classes), but it is getting there quickly.

Using convex optimization is much more difficult than using least-squares or linear programming, as it is often difficult to recognize if a problem is convex. In fact, many problems are not convex as stated but can be transformed into a convex optimization problems. For example, in circuit design, we often work with the logs of the lengths and widths of the components. In statistics, we often look for the inverse of the covariance matrix rather than the covariance matrix itself. This course is about learning to recognize these problems as convex and the tricks to transform them into forms that are solvable.

A note on nonlinear programming. You may hear about a related problem called a nonlinear program (NLP[6]). These are optimization problems in which the objective or constraints are not linear but not known to be convex. There are no effective methods for solving these problems in general. Often, we have to resort to heuristics (*e.g.,* local optimization, convex relaxations) or have to accept worst-case exponential complexity. Fundamentally, this decision is a tradeoff between optimality and speed. In some sense, the distinction between LPs and NLPs is a historic one, dating from the period of time when it was not widely known that some nonlinear optimization problems are much more difficult to solve than others. In his 1993 SIAM Review survey paper [Roc93], Rockafellar said:

> In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.

Convex analysis has been a well-developed field of mathematics since around 1970 (the canonical reference being [Roc70]), but the first formal argument for the efficiency of convex optimization dates to Nemirovski and Yudin in 1983 [NY83]. Later, Nesterov and Nemirovski [NN94] showed the effectiveness of interior point methods on these problems.

[5] In the past ten years, first order solvers for large-scale problems have become much more popular. We'll discuss the differences between solvers later in the course.

[6] Sorry ML people, we had it first.

3 Example: radiation treatment planning

In radiation treatment, radiation is delivered to a patient to kill the cells in a tumor, while minimally affecting the surrounding healthy tissue. Radiation is delivered via n beams, which have intensities $x_1, \ldots, x_n$ and must satisfy

$$0 \leq x_i \leq I^{\max}, \qquad i = 1, \ldots, n.$$

The tissue of the patient is divided into m voxels, labeled $i = 1, \ldots, m$. The dose y_i delivered to voxel i is linear in the beam intensities:

$$y_i = \sum_{j=1}^{n} a_{ij} x_j, \qquad i = 1, \ldots m.$$

The matrix $A \in \mathbf{R}_+^{m \times n}$ is known. A subset of the voxels correspond to the tumor region, while the others correspond to healthy tissue. Our goal is to find the beam intensities x that minimize the dose in the healthy tissue while ensuring the dose in the tumor is above some threshold. We model this by introducing a target dose $d_i^{\text{target}} \in \mathbf{R}_+$ for each voxel i and minimizing the squared deviation from this target. With this objective, the optimization problem is [7]

$$\begin{array}{ll} \text{minimize} & \|y - d^{\text{target}}\|_2^2 \\ \text{subject to} & y = Ax \\ & 0 \leq x \leq I^{\max}. \end{array} \tag{2}$$

We'll consider four approaches to this problem: two using least-squares, one using linear programming, and one using convex optimization. Only the last approach solves the problem (2) exactly. The others are heuristics but may work quite well in many cases. (In fact, these kinds of heuristics are often used in practice for engineering design. If you recognize this in your field, you have an opportunity to easily improve the state of the art!)

Approximate solution using least-squares. The problem (2) obviously cannot be handled directly via least-squares. One approximate approach is to solve the problem

$$\text{minimize} \quad \|Ax - d^{\text{target}}\|_2^2,$$

and then round x_j to the interval $[0, I^{\max}]$. This procedure, before rounding, gives the intensities in Figure 1. Red lines indicate the bounds of the feasible region. After rounding the beam intensities to a feasible point, the value of the objective is `3.474`. The residuals are shown in Figure 2. A better approach is to use weighted least-squares:[8]

$$\text{minimize} \quad \|Ax - d^{\text{target}}\|_2^2 + \sum_{i=1}^{m} w_j (x_j - I^{\max}/2)^2.$$

[7] This problem is modified from [Fu+19] (`https://web.stanford.edu/~boyd/papers/conrad.html`). I significantly simplified the problem, so if you're interested in learning more, check out the paper!

[8] Note that this problem still can be solved with standard least-squares solvers.

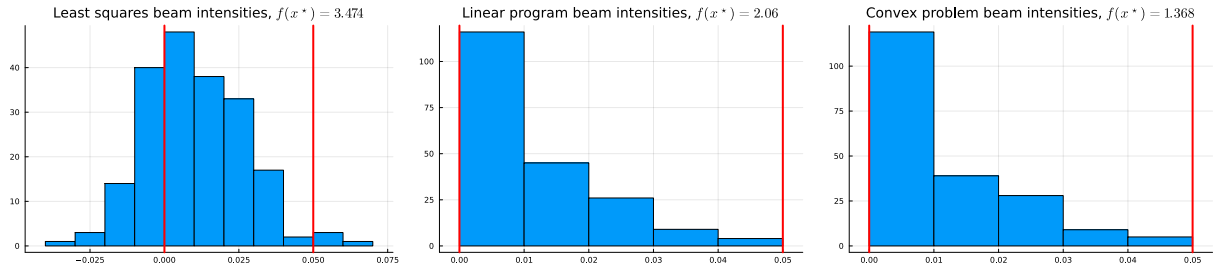

Figure 1: Beam intensities for the least-squares solution (left), linear programming solution (center), and convex optimization solution (right). Red lines indicate intensity constraints. The corresponding objective values are `3.474`, `2.060`, and `1.368` respectively.

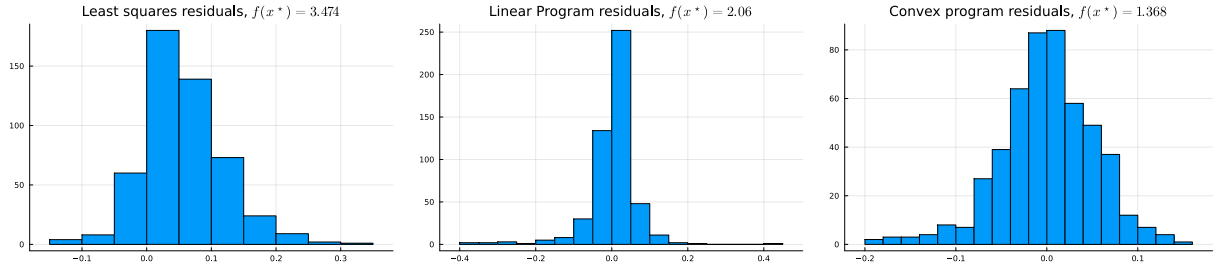

Figure 2: Residuals $Ax^\star - d^{\text{target}}$ for the least-squares solution (left), linear programming solution (center), and convex optimization solution (right).

This problem penalizes the variable x_j if it is far from the middle of the feasible interval, $I^{\max}/2$. We can start the weights at $w_j = 0$ and then iteratively adjust the weights until all $x_j \in [0, I^{\max}]$. Of course, this is still an approximate (*i.e.,* suboptimal) solution in general, although with careful and time consuming tuning, the true optimal value may be attained.

Approximate solution using linear programming. All the constraints are clearly linear. Thus, another heuristic approach is to modify the objective, then use linear programming. One possible relaxation is

$$\begin{array}{ll} \text{minimize} & \|y - d^{\text{target}}\|_1 \\ \text{subject to} & y = Ax \\ & 0 \leq x \leq I^{\max}, \end{array}$$

which can be converted into a linear program using the techniques we saw earlier. Note that instead of using the ℓ_1 norm, we could use the ℓ_∞ norm, or a combination of both the ℓ_1 and ℓ_∞ norms. Intensities found via this approach are shown in Figure 1, and the residuals are shown in Figure 2. Note that the ℓ_1 norm can lead to the presence of large outliers; the maximum and minimum residual from this approach are significantly greater than those from the other approaches. The value of the objective is `2.060`.

Solution via convex optimization. It can be directly verified that the problem (2) is convex; the objective is convex and the constraints are linear. Thus, the problem as written can be solved via standard convex optimization software. Intensities found via this approach are shown in Figure 1, and the residuals are shown in Figure 2. Note that this solution not only has the smallest objective value, but also has the tightest clustering of residuals. (And in this application, it's likely very important that we do not over-radiate or under-radiate cells.) The value of the objective is `1.368`. In general, many problems in practice are still solved via heuristic methods that are similar to those introduced above. We will see that convex optimization both provides a means to tackle these problems directly and greatly expands the set of problems we can tackle at all. A great reference for further reading, beyond what we have time to cover in the course, is the book by Boyd and Vandenberghe [BV04].

References

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Fu+19] Anqi Fu et al. "A convex optimization approach to radiation treatment planning with dose constraints". In: *Optimization and Engineering* 20.1 (2019), pp. 277–300.

[Kan39] LV Kantarovich. "Mathematical methods in the organization and planning of production". In: *Publication House of the Leningrad State University.[Translated in Management Sc. vol 66, 366-422]* (1939).

[MT20] Per-Gunnar Martinsson and Joel A Tropp. "Randomized numerical linear algebra: Foundations and algorithms". In: *Acta Numerica* 29 (2020), pp. 403–572.

[NN94] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial methods in convex programming*. SIAM, 1994.

[NY83] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[Roc70] R Tyrrell Rockafellar. *Convex analysis*. Vol. 18. Princeton university press, 1970.

[Roc93] R Tyrrell Rockafellar. "Lagrange multipliers and optimality". In: *SIAM review* 35.2 (1993), pp. 183–238.

[TBI97] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*. Vol. 50. Siam, 1997.