

Lecture 1 for EE127 (Fall 2018): Introduction

Scribes: Yagna Patel

8/23/2018

1 Introduction

The name of the course is Optimization Models. The term *Optimization* is the subject matter of this course. In this course, we view everything as an optimization problem. Optimization is a mathematical language we use to describe the world. The term *Models* refers to the fact that throughout the course, rather than solving for these various models, we will consider them as black boxes. We won't delve too deep into the algorithms.

1.1 What is optimization?

When optimizing something, we generally minimize or maximize some objective, or cost, function. The standard form for an optimization problem is:

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

where

- $\mathbf{x} \in \mathbb{R}^n$ is the decision variable. The decision, or choice, we have to make is encoded into a finite dimensional vector, \mathbf{x} . For example, in statistical learning, where our goal is to predict something such as a stock price, demand, etc., \mathbf{x} would be the parameter which helps you make an accurate prediction. In self driving cars, \mathbf{x} could be the speed of the car.
- $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective, or cost, function.
- $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ are the constraints. These constraints, $f_i(\mathbf{x})$, that our decision should satisfy. For example, in the case of self driving cars, if we wanted to restrict our car to drive within a certain speed limit, we could add a constraint for \mathbf{x} .

- \mathbf{p}^* is the optimal value. Solving an optimization problem essentially means to find a solution, \mathbf{p}^* , such that the constraints are satisfied and such that the objective function is minimized. Sometimes, if the constraints are too stringent, the solution may not exist. Essentially, \mathbf{p}^* is a solution to the optimization problem if for all \mathbf{x} such that for every i , $f_i(\mathbf{x}) \leq 0$, then $f_0(\mathbf{p}^*) \leq f_0(\mathbf{x})$.

Note that we can write *any* optimization problem in this standard form (1). For example, if we wanted to maximize our objective function, $\max_{\mathbf{x}} f_0(\mathbf{x})$ we could instead minimize $\min_{\mathbf{x}} -f_0(\mathbf{x})$.

1.2 Examples of Optimization Problems

Almost every field of science and engineering has applications of optimization. We will look at a few of those applications.

1.2.1 Least-Squares Regression

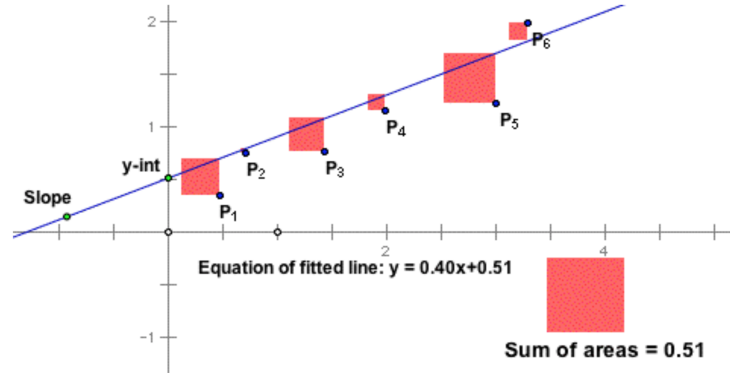


Figure 1: Least-Squares Example

One of the oldest optimization problems is the least-squares regression problem. It holds roots back to Gauss's time. This is an example of an optimization problem that doesn't have any constraints. In Figure 1, we see an example of performing least-squares regression on 6 data points. The goal of least-squares regression is to minimize the sum of the areas for each point for a fitted line. Therefore, the objective function becomes

$$f_0(\mathbf{x}) = \sum_{i=1}^m \left(y_i - \mathbf{x}^\top \mathbf{z}^{(i)} \right)^2 = \|\mathbf{y} - \mathbf{Z}\mathbf{x}\|_2^2$$

Note that $\|\cdot\|_2^2$ is the square of the euclidean norm. Therefore, our optimization problem is

$$\min_{\mathbf{x}} f_0(\mathbf{x})$$

where

- $\mathbf{Z} \in \mathbb{R}^{m \times n}$ is a matrix of input data points.
- $\mathbf{y} \in \mathbb{R}^m$ is a *response* vector. Response vectors are essentially output vectors.
- $\mathbf{x}^\top \mathbf{z}$ is the scalar product, $z_1 x_1 + \dots + z_n x_n$ between the two vectors $x, z \in \mathbb{R}^n$. $\mathbf{x}^\top \mathbf{z}$ is the fitted line. Once we have figured out the best fit line, the line is encoded into \mathbf{x} . \mathbf{x} is the *decision* variable; it allows us to make a precise prediction.

Now, if we get a new point, we can use the fitted line (blue line in Figure 1) to make a prediction.

1.2.2 Linear Classification

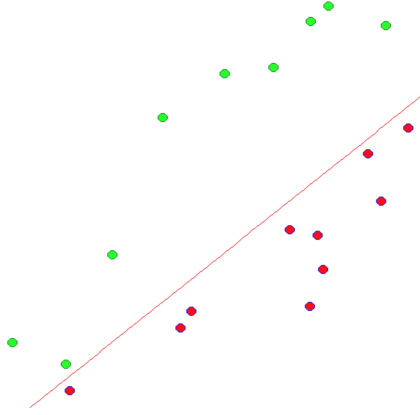


Figure 2: SVM Example

Another example of an optimization problem is linear classification. Suppose that, in Figure 2, each of the points are in 40,000 dimensions, and the red points represent patients who have *not* developed diabetes over the past 6 months, and the green points represent patients who have developed diabetes. Now, suppose that a new patient comes to us. How can we predict whether or not this patient will have diabetes?

This is a linear classification problem. We want to find a boundary which best separates the green points and the red points. One way to approach this problem is to use Support Vector Machines (SVMs).

The optimization problem for an SVM is

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

where

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

Now, in order to predict whether the patient will have diabetes, we can look at the sign of the hyperplane $\mathbf{x}^\top \mathbf{z} + \mathbf{b}$. This will tell us which side of the hyperplane the point lies, which in turn tells us whether or not the patient will have diabetes.

Notice that this wasn't a least-squares problem. It is different from a least-squares problem in that we are instead minimizing a convex function. Simply put, a convex function is a bowl-shaped function.

1.2.3 Further Examples

See [Lecture 1 slides](#) for more examples of optimization problems.

2 Optimization Problems

Recall the general structure of an optimization problem from equation (1). Let's look at some definitions that will prove useful when dealing with optimization problems.

2.1 Definitions

Definition 2.1. A point \mathbf{x} is **feasible** if $\mathbf{x} \in \text{dom } f_0$ and \mathbf{x} satisfies all the constraints.

Given that f_0 is our objective function, a point x is said to be feasible for an optimization problem, if it is in the domain of f_0 and it satisfies all the constraints of the optimization problem. The set of all such feasible x is called a **feasible set**. If a point is not feasible, it is **infeasible**.

Definition 2.2. A feasible point \mathbf{x} is **optimal** if $f_0(\mathbf{x}) = \mathbf{p}^*$.

The set of all such optimal \mathbf{x} is called an **optimal set**.

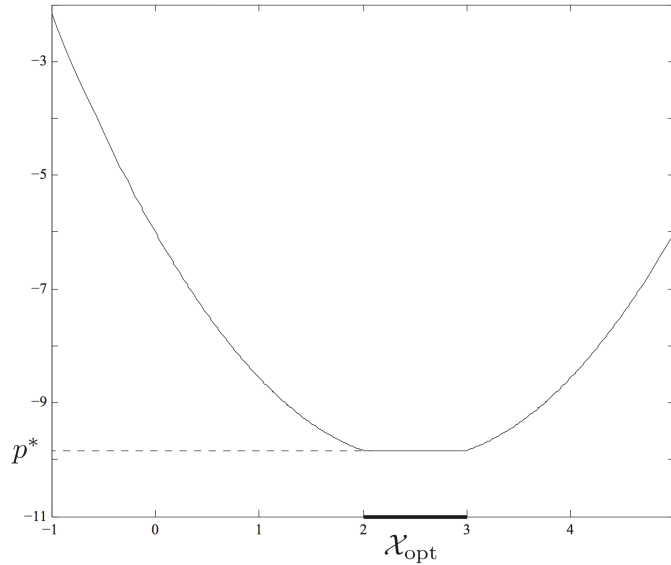


Figure 3: Optimal Set Example

Notice that in Figure 3 the optimal set is $\chi_{\text{opt}} = [2, 3]$. All the points in $[2, 3]$ achieve a global minimum value of $\mathbf{p}^* = -9.84$

Definition 2.3. A feasible point \mathbf{x} is ϵ -suboptimal if $f_0(\mathbf{x}) \leq \mathbf{p}^* + \epsilon$.

The set of all such ϵ -suboptimal points forms an ϵ -suboptimal set.

Definition 2.4. A point \mathbf{z} is locally optimal if there exists an $R > 0$ such that \mathbf{z} is optimal for

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & \|\mathbf{x}_i - \mathbf{z}_i\|_2 \leq R, \quad i = 1, \dots, n. \end{aligned}$$

Consider the following graph:

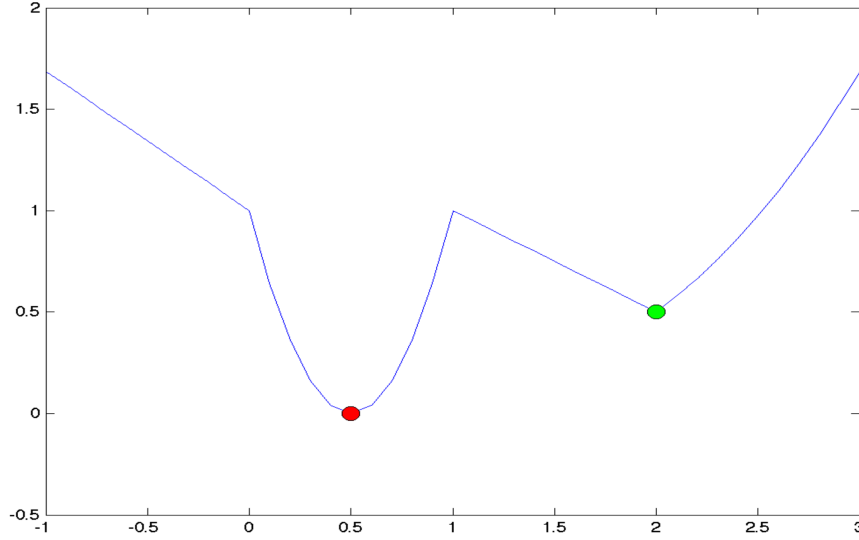


Figure 4: Local/Global Minimum Graph

In Figure 4 we see the graph of a function of one variable. Now, notice the two points, the green point and the red point. Suppose we run an algorithm to find the minimum of a function, and suppose that our algorithm returned the green point. Now, the green point is nice since locally it is better than its neighbors. However, as we grow our neighborhood, we see that it isn't the best local minimum. Instead, the red point gives the global minimum of the function. In practice, using either minima could yield very different results. For example, in deep learning, if we have constraints and our algorithm find such a local minimum, it could fail to find a feasible point even though there may exist one. However, for most problems in deep learning, they are unconstrained. Therefore, it isn't really a problem if we don't find the global minimum.

Definition 2.5. A t -level set of a function f is defined as

$$L_t(f) := \{x \in \mathbb{R}^n : x \in \mathbb{R}^n, t = f(x)\}$$

Definition 2.6. A t -sublevel set of a function f is defined as

$$L_t(f) := \{x \in \mathbb{R}^n : x \in \mathbb{R}^n, t \geq f(x)\}$$

Example 2.7. Consider the optimization problem

$$\begin{aligned} \min_x \quad & 0.9x_1^2 - 0.4x_1x_2 - 0.6x_2^2 - 6.4x_2 - 0.8x_2 \\ \text{subject to} \quad & -1 \leq x_1 \leq 2, 0 \leq x_2 \leq 3. \end{aligned}$$

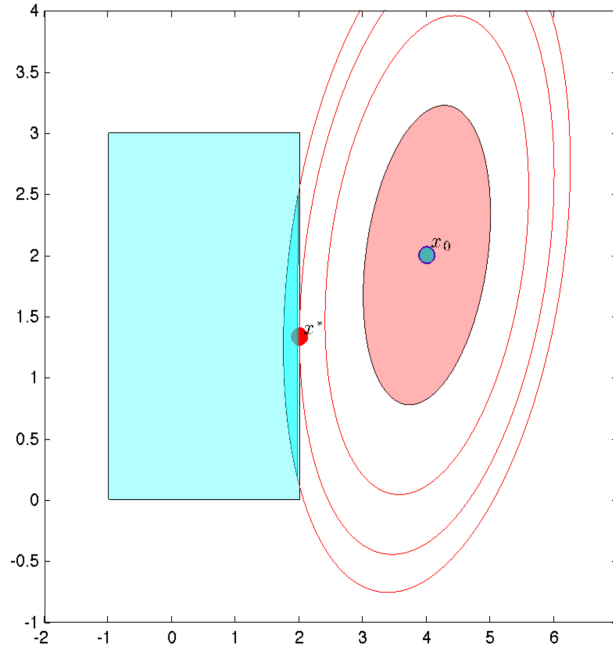


Figure 5: Optimization Problem Definitions Example

- The feasible set is in light blue.
- The 0.1-suboptimal set is in darker blue.
- The unconstrained minimizer is x_0 . This is the solution to the optimization problem without any constraints applied.
- The optimal point is x^* .
- The level sets of the objective function are the red lines.
- The sub-level set is the filled in red ellipse.
- The optimal value $p^* = -10.2667$ is attained by the optimal solution $x_1^* = 2, x_2^* = 1.3333$.

2.2 Convex Problems

Recall the graph in Figure 4 where we had multiple local minima. As we saw, this is a hard case to deal with. However, there is a class of optimization problems that don't face this issue of multiple local minima. This class contains optimization problems where the objective function and constraints involve convex functions.

2.2.1 What does it mean to be convex?

Roughly speaking, a convex function has *bowl-shaped* graph:

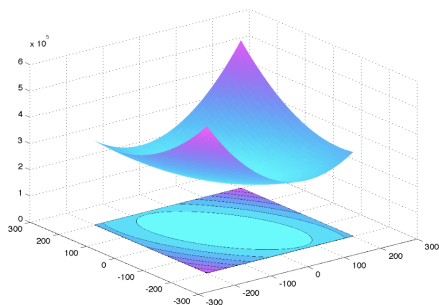


Figure: Convex function.

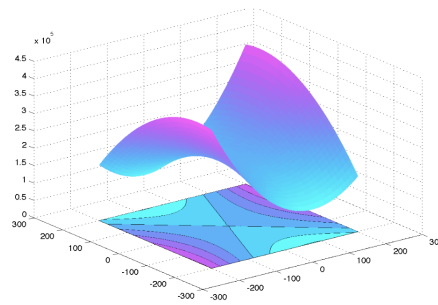


Figure: Non-convex function.

Figure 6: An Example of a Convex and Non-convex Function

Definition 2.8. A set $E \subset \mathbb{R}^n$ is convex if

$$\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in E$$

whenever $\mathbf{x}, \mathbf{y} \in E$ and $0 < \lambda < 1$.

Loosely, this definition is saying that if a function is convex and I draw a line from one point to another point on that function, the line will always lie inside the function.

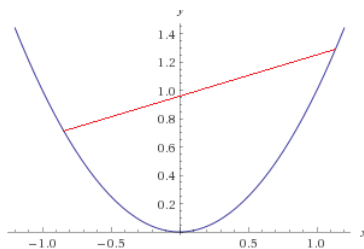


Figure 7: A Convex Function

Notice, in Figure 7, how the red line lies inside the parabola. If this is true for any two points on the function, the function is said to be convex. One of the key features of convex functions is that all local minima are global minima. Recall the function in Figure 3. This function was also convex, with multiple global minima.

When dealing with a convex optimization problem, optimization becomes very reliable. In the past 50 years, many people have been working on solving linear equations at scale. Now, modules and functions in Python and Matlab can reliably solve linear equations. Instead of worrying about whether or not an algorithm in Python found the solution, we have reached the point where we can consider it a reliable black box.

Convex optimization is similar. It is very reliable to solve an optimization problem where we know the objective function and its constraints to involve convex functions. However, not all convex problems are easy to solve. In this course, we will mostly deal with convex optimization.

2.2.2 Special Convex Models

In this course, we will 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)

We will utilize certain software (such as CVX, Yalmip, Mosek, etc.) which contain efficient algorithms, to solve such optimization problems.

2.3 Non-convex Problems

There are three main types of Non-convex optimization problems:

2.3.1 Boolean/Integer Optimization

Suppose that we are an airline company that is trying to allocate resources: pilots, flight attendants, etc.. Now, suppose that we want to send some pilots and flight attendants to airplanes based on demand, traffic, etc. However, notice that there is another layer of complexity in this problem: we cannot send half a pilot to an airplane. Here, we are facing a boolean optimization problem. Solving such problems is hard.

2.3.2 Cardinality-Constrained Problem

Here we seek to bound the number of non-zero elements in a vector variable.

2.3.3 Non-linear Programming

These consist of non-convex problems with differentiable objective function.

References

- [El 18a] Laurent El Ghaoui. Ee127 fall 2018 lecture 1. <https://bcourses.berkeley.edu/courses/1473158/files/73475329/download?wrap=1>, 2018. EE127 Fall 2018 Lecture Notes.
- [El 18b] Laurent El Ghaoui. Optimization models and applications. <http://livebooklabs.com/keepies/c5a5868ce26b8125>, 2018. Livebook visited Spring 2018.
- [Rud76] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.