AM 221: Advanced Optimization Spring 2016

Prof. Yaron Singer *Lecture 1 — Monday, January 25th*

1 Overview

In this course we will cover optimization through the perspective of convex (continuos) and submodular (combinatorial) optimization and emphasize the deep connections between these two areas. Convex optimization is a central tool for solving large-scale problems which in recent years has had a profound impact on statistical machine learning, data analysis, mathematical finance, signal processing, control, approximation algorithms, as well as many other areas. Submodular optimization has had a profound impact on game theory, mechanism design, machine learning, vision, data mining, approximation algorithms, and many other areas. In a single sentence, the premise of this course can be summarized as follows:

Optimization is an elegant mathematical theory which provides fundamental tools for reasoning about and solving problems across a broad range of areas in the data sciences.

The first part of the course will be dedicated the theory of convex optimization and its direct applications. The second part will focus on advanced techniques in combinatorial optimization using machinery developed in the first part. Let's start.

2 Some Examples from Statistical Machine Learning

Let's assume that we want to predict people's height from data on their height and shoe size. We'll collect a set of observations $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i, y_i \in \mathbb{R}$ for all $i \in [m]$, and build a statistical model to make a prediction. That is, we would like to build a model which predicts someone's height given their shoe size.

One popular model for prediction is *linear regression*: we assume that the shoe size is simply a linear function of the height, i.e. there exist some $a, b \in \mathbb{R}$ such that $\texttt{shoe size} = a \cdot \texttt{height} + b$, or (using y to denote shoe size and x as height in cm) $y = ax + b$ and seek to fit the parameters a, b which minimizes the residual sum of squares. Our objective is then to find:

$$a^\star = \text{argmin}_{a \in \mathbb{R}} \sum_{i=1}^{m} (y_i - (a \cdot x_i + b))^2$$

For simplicity let's assume for now that $b = 0$. In this case, to find the parameter a that minimizes the function above we can simply take the derivates of $\sum_{i=1}^{m} (y_i - a \cdot x_i)^2$, in pieces:

$$\begin{aligned} \frac{\partial (y_i - a \cdot x_i)^2}{\partial a} &= 2(y_i - ax_i)(-x_i) \\ &= 2ax_i^2 - 2x_i y_i \end{aligned}$$

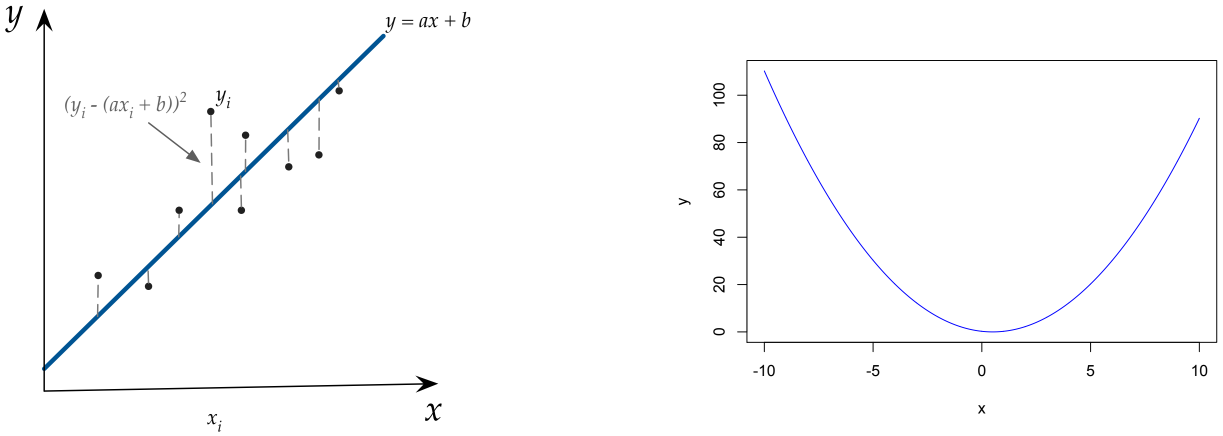

Figure 1: Depiction of linear regression and RSS. On the left we have example data points and the line that best fits these points (minimizes the RSS). On the right we depict an example of a function $y_i = ax_i + b$ where we fix $y_i = 1/2, x_i = 1$ and $b = 0$. Notice that the shape of this curve is convex.

Thus:

$$\frac{\partial \sum_{i=1}^{m} (y_i - a \cdot x_i)^2}{\partial a} = 2 \sum_{i=1}^{m} a x_i^2 - x_i y_i$$

The second derivative here is $\sum_{i=1}^{m} 2x_i^2$ which is positive, thus the critical point is a local minimum, and therefore $a^\star$ is the solution a for which: $\sum_{i=1}^{m}(2ax_i^2 - 2y_i x_i) = 0$. Rearranging, we get that:

$$a^\star = \frac{\sum_{i=1}^{m} x_i y_i}{\sum_{i=1}^{m} x_i^2}$$

Linear regression in multiple dimension. The above example is a very simple case of linear regression which is one of the most well-studied models for prediction and classification. In our example, we can easily imagine a situation where each data point contains more information, e.g. shoe size, weight, gender, age, etc. and the height. More generally, given a d-dimensional point of data $\mathbf{x} \in \mathbb{R}^d$, the model assumes that the output $y \in \mathbb{R}$ is a linear function of $\mathbf{x}$:

$$y = \mathbf{a}^\intercal \mathbf{x} + b = \sum_{j=1}^{d} a_j x_j + b$$

For convenience, it is common to include $b \in \mathbb{R}$ in the vector of coefficients (by considering $\mathbf{x}$ as a $d+1$ dimensional point with the first entry set to 1), and write the linear model as the inner product: $y = \mathbf{a}^\intercal \mathbf{x}$. Given a set of observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ the residual sum of squares function is defined as:

$$RSS(\mathbf{a}) = \sum_{i=1}^{m} (y_i - \mathbf{a}^\intercal \mathbf{x})^2$$

The goal is then fit a vector of parameters $\mathbf{a}$ which minimizes the residual sum of squares:

$$\mathbf{a}^\star \in \text{argmin}_{\mathbf{a} \in \mathbb{R}^d} RSS(\mathbf{a})$$

LASSO: Convex optimization under constraints. In a similar manner to the way we found the optimal solution to the example above, the solution has a closed form solution here too. In practice, more sophisticated methods are used such as the Least Absolute Shrinkage and Selection Operator (LASSO) where we seek a solution that enforces some notion sparsity:

$$\begin{aligned} \min \sum_{i=1}^{m} (y_i - \mathbf{a}^{\intercal}\mathbf{x})^2 \\ \text{s.t.} \sum_{j=1}^{d} |a_j| \leq t \end{aligned}$$

The LASSO helps improve prediction accuracy by setting some features to 0, and helps interpretation as it intuitively enforces a smaller number of features. This is an example of an optimization problem *under constraints*: we seek a solution $a = (a_1, \ldots, a_d)$ which has the property of being optimal under the constraint that $\sum_{j=1}^{d} |a_j| \leq t$. For the LASSO method the optimal solution can no longer be expressed in closed form. In order to apply such techniques we must develop algorithms that fit the parameters appropriately. More importantly, if we want to design methods such as LASSO, we must understand what kind of statistical models one can fit parameters for.

Sparse regression: combinatorial optimization. Another example, similar to LASSO, is that of *sparse regression.* Here the goal is to fit the best hyperplane of dimension at most t that best fits the data. Formally, the objective is then:

$$\begin{aligned} \min \sum_{i=1}^{m} (y_i - \mathbf{a}^{\intercal}\mathbf{x})^2 \\ \text{support}(\mathbf{a}) \subseteq S \\ \text{s.t. } |S| \leq t \end{aligned}$$

Where support($\mathbf{a}$) indicates the non-zero entries of the vector $\mathbf{a}$. The goal is essentially to select the best set of features of size S that minimize the RSS functions. This is an example of a *combinatorial optimization* problem, or an *integer program.* A feasible solution is one which has 1s in indices that correspond to S and 0s in indices that do not correspond to S.

3 Optimization Problems

The example of fitting parameters is a special case of an *optimization problem*: minimizing (or maximizing) a function f under some constraints. We usually write optimization problems as:

$$\begin{aligned} \text{min (or max) } & f(\mathbf{x}) \\ s.t. \; & g_1(\mathbf{x}) \leq b_1 \\ & . \\ & . \\ & . \\ & g_m(\mathbf{x}) \leq b_m \end{aligned}$$

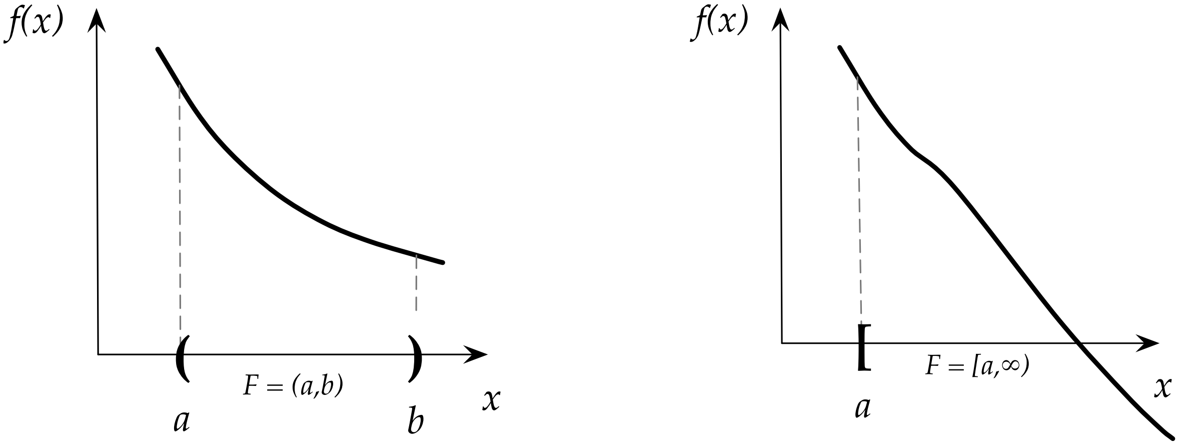

Figure 2: A depiction of open (left) and unbounded (right) feasible regions.

where $\{g_i(\mathbf{x})\}_{i=1}^m$ encode the constraints. In the LASSO case, there was only one constraint $g(\mathbf{x}) = \sum_{j=1}^d x_j \leq t$. The set of points which respect the constraints is called the *feasible set.*

Definition. *For a given optimization problem the set $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n \ : \ g_i(\mathbf{x}) \leq b_i, \forall i \in [m]\}$ is called the **feasible set**. A point $\mathbf{x} \in \mathcal{F}$ is called a **feasible point**, and a point $\mathbf{x}' \notin \mathcal{F}$ is called an **infeasible point**.*

Ideally, would would like to find optimal solutions for the optimization problems we consider. Let's define what we mean exactly.

Definition. *For a* maximization *problem $\mathbf{x}^\star$ is called an **optimal solution** if $f(\mathbf{x}^\star) \geq f(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{F}$. Similarly, for a* minimization *problem $\mathbf{x}^\star$ is an optimal solution if $f(\mathbf{x}^\star) \leq f(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{F}$.*

Does an optimization problem always have an optimal solution?

Consider a few examples depicted in Figure 2:

(a) $\mathcal{F} = \emptyset$

(b) $\mathcal{F} = (a, b)$

(c) $\mathcal{F} = [a, \infty)$

In the first example the feasible region (set) is empty and hence the optimal solution cannot exist. In the second case the region is open and therefore the optimal solution cannot be obtained since for every point we will choose, there will always be another point closer to the minimum or maximum. Lastly, in the third example, the region in unbounded, thus there will always be another point that can be closer to the minimum or maximum of the function.

Definition. *If $\mathcal{F} = \emptyset$ we say that the optimization problem is* ***infeasible****. If $\mathcal{F} \neq \emptyset$ we say the optimization problem is* ***feasible****.*

Definition. *A set $\mathcal{F}$ is* ***closed*** *if it contains all its limit points.*

Definition. *A set $\mathcal{F}$ is* ***bounded*** *if $\exists \epsilon > 0$ s.t. $\mathcal{F} \subseteq \mathcal{B}_\epsilon$ (ball in $\mathbb{R}^n$ with radius ϵ). If there is no such ϵ then $\mathcal{F}$ is called* ***unbouned****.*

Definition. *If for any $\lambda \in \mathbb{R}$ there exists $\mathbf{x} \in \mathcal{F}$ s.t. $f(\mathbf{x}) \geq \lambda$, then we say that the maximization problem is* ***unbounded****. Similarly, for a minimization problem we say it is unbounded if for any $\lambda \in \mathbb{R}$ there exists $\mathbf{x} \in \mathcal{F}$ for which $f(\mathbf{x}) \leq \lambda$.*

3.1 Sufficient Condition for Optimality

The following theorem, which is a fundamental theorem in real analysis, gives us a sufficient (though not necessary) condition for optimality. Before stating the theorem, let's first recall the Bolzano-Weierstrass theorem from real analysis (which you will prove in section this week).

Theorem. *(Bolzano-Weierstrass) Every bounded sequence in $\mathbb{R}^n$ has a convergent subsequence.*

Theorem (Weierstrass)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\mathcal{F} \subseteq \mathbb{R}^n$ be nonempty, bounded, and closed. Then, the optimization problem $\min f(\mathbf{x}) : \mathbf{x} \in \mathcal{F}$ has an optimal solution.*

Proof. Let α be the infimum of f over $\mathcal{F}$ (i.e. the largest value for which any point $x \in \mathcal{F}$ respects $f(x) \geq \alpha$), and for some $\epsilon \in (0, 1)$ define:

$$\mathcal{F}^k := \{x \in \mathcal{F} : \alpha \leq f(x) \leq \alpha + \epsilon^k\}.$$

Notice that $\mathcal{F}^k$ is bounded, for any k. By the Bolzano-Weierstrass theorem we know that there is a convergent subsequence. Now, consider a convergent subsequence of points $\{x^k\}_{k=1}^\infty$ whose limit is some $\bar{x}$. Since $\mathcal{F}$ is closed we know that $\bar{x} \in \mathcal{F}$. Since f is continuous we have that $\lim_{k\to\infty} f(x^k) = f(\bar{x})$. The optimal solution is $\bar{x}$. □

4 Convex Optimization

Before we conclude, let's briefly discuss one more concept. If we take a look at the RSS function we aim to minimize in regression models, we will see it follows a particular structure. For convenience, we plot such a function in Figure1. As we can see, the terms $\{(y_i - (ax_i + b))^2\}_{i=1}^m$ of the RSS function have a convex shape – any two points on the curve sit below the line connecting them. Appropriately, functions that have such structure are called *convex functions*. Some argue that convex functions and their combinatorial analogues submodular functions are the most general classes of functions that we know how to optimize. This perspective aligns with the agenda of this course where we'll develop a rich theory and impressive arsenal of techniques for solving convex and later submodular optimization problems.

So why convex functions? In a single sentence: for convex functions a local minimum is also a global minimum, and this fact makes searching for an optimal solution computationally feasible.

5 Roadmap

We will begin by introducing basic concepts from convex analysis and prove fundamental properties of convex sets in the next lecture. We will then continue to linear optimization (which is a special case of convex optimization), and cover fundamental concepts like duality, and describe algorithms for solving linear optimization problems (roughly 3 to 4 weeks). We will then generalize the concepts and develop algorithms for convex optimization problems (roughly 4 weeks). In the last part of the course, we will transition into combinatorial optimization, and develop a theory for combinatorial optimization, a majority of which is largely based on the machinery we developed for convex optimization.