EE C227C Convex Optimization and Approximation Lecture 1 - 01/18/2021

Lecture 1: Introduction

Lecturer: Jiantao Jiao *Scribe: Donghao Ying*

In this lecture, we give basic introductions to the course. One can also refer to chapter 1.1 in [1].

1 Formulation

A general nonlinear optimization problem can be written as:

$$\begin{aligned} &\min\ f_0(x) && (1)\\ &\text{s.t. } f_j(x)\ \&\ 0,\ j=1,2,3,\ldots,m && (2)\\ &\quad\ x\in Q, && (3)\end{aligned}$$

where $\& \in \{\geq, \leq, =\}$. The set Q is referred to as the basic feasible set. We define $f(x) := (f_1(x), \ldots, f_m(x))$ as the vector of functional constraints, and we remark that the entire feasible region can be expressed as

$$\{x \in Q \mid f_j(x)\ \&\ 0,\ j=1,2,3,\ldots,m\}. \tag{4}$$

The problem is said to be *smooth*, if every $f_j(x)$, for $j \in [m]\cup\{0\}$, is differentiable. Otherwise, we say that the problem is *non-smooth*.

2 Application

Nonlinear optimization is almost universal. Below, we give several modeling examples.
Example 1 (Fitting)**.** Suppose we want to find $x \in \mathbb{R}$ such that $f_j(x) = a_j \in \mathbb{R}$ for all $j \in [m]$. We can define the following problem

$$\mathrm{OPT} = \min_{x\in\mathbb{R}^n} \sum_{j=1}^{m} \left(f_j(x) - a_j\right)^2, \tag{5}$$

so that $\mathrm{OPT} = 0$ indicates a perfect fit can be found. Otherwise, the problem is infeasible.
Example 2 (Integer optimization)**.** The integral constraints can be modeled using trigonometric functions. For example,

$$\begin{aligned} &\min\ f_0(x) && (6)\\ &s.t.\ \sin\left(\pi x^{(i)}\right) = 0,\ 1\leq i\leq n, && (7)\end{aligned}$$

where we denote $x = \left(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\right)$.

During 1950's, people are excited about nonlinear optimization. However, in 1970's, it was shown that even for Lipschitz functions, the problem is not solvable in a precise sense. Therefore, suitable evaluation metrics are required to evaluate the performance of numerical methods.

3 Performance Evaluation of Numerical Methods

We first remark that, a well-defined performance evaluation for a method $\mathcal{M}$ should not only work for a particular problem P, but a class of problems $\mathcal{P}$.

Definition 3 (Model)**.** *The model of problem $\mathcal{P}$, denoted by Σ, is defined as the known part of $\mathcal{P}$.*

Definition 4 (Oracle)**.** *The oracle, denoted by $\mathcal{O}$, answers questions of the method $\mathcal{M}$.*

In practice, there are usually three kinds of oracles:

- Zero-order oracle, which sends back $f(x)$ given x.
- First-order oracle, which sends back $(f(x), \nabla f(x))$ given x.
- Second-order oracle, which sends back $\big(f(x), \nabla f(x), \nabla^2 f(x)\big)$ given x.

In general, the components of a problem $\mathcal{P}$ can be represented by a three-tuple $(\Sigma, \mathcal{O}, \mathcal{M})$. The performance of $\mathcal{M}$ on $\mathcal{P}$ is defined as the total amount of *computational effort* required by $\mathcal{M}$ to *solve* $\mathcal{P}$. To be specific, *solve* refers to finding an approximate solution with accuracy ε. Hereby, we define the *stopping criterion*, denoted by τ_ε, such that we stop the computation when τ_ϵ is true.

To give a definition for the computational efforts, we need to first introduce what is a *Generic Iterative Algorithm.* A generic iterative algorithm consists of the following parts:

- Input: starting point x_0, accuracy $\varepsilon > 0$.
- Initialization: $k = 0$, information set $\mathcal{F}_{-1} = \emptyset$.
- Loop:
  1. Call oracle at point x_k.
  2. Update the information set $\mathcal{F}_k = \mathcal{F}_{k-1} \cup (x_k, \mathcal{O}(x_k))$.
  3. Apply method $\mathcal{M}$ to $\mathcal{F}_k$ and generate a new point x_{k+1}.
  4. Check τ_ε. If yes, output $\bar{x}$. Otherwise, set $k = k + 1$ and go to step 1.

Now, we define the computational efforts by the following two complexity measures.

Definition 5 (Analytical complexity)**.** *The analytical complexity, which is also called the information based complexity, is defined as the number of calls to the oracle during the computation.*

Definition 6 (Arithmetical complexity)**.** *The arithmetical complexity is defined as the number of arithmetic operations, including the work of oracle and the work of the method, which is necessary to solve the problem up to accuracy ε.*

4 Summary

In this lecture, we give basic introductions to the course, including the generic formulation of a nonlinear optimization problem, applications of nonlinear optimization, and the performance evaluation of numerical methods. In the next lecture, we will

- Show the complexity of minimizing the class of Lipschitz continuous functions.
- Introduce *convexity.*

References

[1] Y. Nesterov *et al.*, *Lectures on convex optimization.* Springer, 2018, vol. 137.