CMPUT 675: Optimization and Decision Making under Uncertainty

Shuai Liu
instructor: Xiaoqi Tan

September 2021

Contents

1 Lec 1: Convex Optimization Pt.1

1.1 Mathematical Optimization

Mathematical optimization takes the following form:

$$\min_{x} f(x)\ , s.t. x \in C \tag{1}$$

where C is called feasible set/region or decision space, $x \in \mathbb{R}^n$ is a vector called **decision variables** whose value can be changed overtime. The **solution** to the problem x^* is also called **optimal point**. The terminology should be paid attention that for a optimization problem, there doesn't exist a concept of "optimal solution" - if there is a existing solution then it should automatically be an optimal point. $f(x^*)$ is called an optimal value or object.

1.2 Continuous optimization & Discrete Optimization

In a **Continuous Optimization Problem**, the decision variables are continuous and the feasible set $C \subset \mathbb{R}^n$. If $C = \mathbb{R}^n$, then the problem is unconstrained; if f is linear and C is a polyhedral, then the problem is a Linear Programming(LP), otherwise is a Non-linear Programming(NLP).
In a **Discrete Optimization Problem**, some of the decision variables are discrete. Decision variables are integers are sometimes called integer programming; problems with emphasis on graphs and matroids are called combinatorial optimization problem and something mixed are called mixed-integer programming.

1.3 3 Pillars of Optimization

1.3.1 Modelling

There are two types of modelling problems: tractable and practical. It mainly depends on the task you are dealing with: for example, a deep learning task. This is not what we'll focus on in this course.

1.3.2 Characterization of Minima(Maxima)

There are a lot of stuffs in this pillar including the optimal conditions, Lagrange Multpliers, Duality, Sensitivity, etc. This pillar and the next one would be the main focus of this course.

1.3.3 Iterative Algorithms

There is a natural question: why we need iterative algorithms? Wouldn't it be better if we can directly solve the problem. The reason is in many cases of optimization, the problems are generally very hard or intractable and only a few could be solved, which means there's no closed form solution that we can immediately obtain. Here comes iterative algorithms: we need to adaptively search for the solution and try to get as better results as possible based on our current "solution".
There are multiple well-known iterative algorithms: GD, Approximation Algorithms and Dual or Primal-Dual Algorithms.

1.4 Convex Sets

A set $C \subset \mathbb{R}^n$ is a convex set if $\forall x, y \in C, \forall \alpha \in [0,1], \alpha x + (1-\alpha)y \in C$. Intuitively, if someone draw a line between two points in a convex set, then the whole line lies within that set. There are multiple operations over sets could preserve the convexity and here we emphasize the intersection operation. Affine sets $\{x : Ax = b\}$ are convex sets and a notable fact is that no other equality constraints are convex except affine sets. Sets of the form $\{x : Ax \preceq b\}$ are polyhedral and are also convex sets.

1.5 Convex Functions

A function $f : C \to \mathbb{R} \ \ s.t. \forall \alpha \in [0,1], \forall x, y \in C, f(\alpha x + (1-\alpha)y)) \geq \alpha f(x) + (1-\alpha)f(y)$, where C is a convex set is a convex function. So how to decide if a function is convex? Of course the definition could do that. But there are other ways that may be more convenient in some sense. First, the feasible set **must be convex**. Then, If the function is differentiable, gradient inequation $f(y) \geq f(x) + \nabla f(x)^T(y-x)$ could also decide a convex function. If the function is twice differentiable, a Positive Semi-definite(PSD) hessian could also decide a convex function: $\nabla^2 f(x) \succeq 0$.

However, even if with all these above, it could still be complicated to prove convexity. People often use some operations that preserve convexity to make things easier. For example, non-negative weighted sum: $g(x) = \sum_i \lambda_i f_i(x)$, composition with affine functions $g(x) = f(Ax)$ and pointwise maximum $g(x) = \sup_{i \in I} f_i(x)$ are all operations preserving convexity.

Compared to sets, we are more familiar with functions. Therefore, in convex optimization, there are operations helping us switch between convex sets and convex functions: epigraphs and level sets. An epigraph of a function f is the part in the space that is upper to f, formally defined as $\{(y,t) | y \geq f(t), t \in dom(f)\}$. A function f is convex if and only if the epigraph of $f : epi(f)$ is a convex set. Level sets are sets of points in the domain whose function values are less than or equal to a specific value: $C_\beta = \{x | f(x) \leq \beta\}$. All level sets of a convex function are convex sets. Note that the other direction doesn't hold, i.e. if all level sets of a convex function are convex sets, then the function is not necessarily convex function.

1.6 Convex Optimization

The standard form of convex optimization looks like:

$$\min_{\substack{x \\ Ax=b \\ f_i(x) \leq 0, i=1,2,3,\ldots,m}} \quad f_0(x) \tag{2}$$

where $f_i(x), i = 0, 1, 2, \ldots m$ are convex functions. It is always been taken for granted that the feasible set is a convex set while we'd like to make it clear why it's convex. Since $f_i(x) \leq 0$ are all epigraphs of $f_i(x)$, which are convex functions, they are all convex sets. Also, the affine sets are all convex sets. As is mentioned before, the operation of intersection over sets could preserve the convexity. The feasible set is thus a convex set.

1.6.1 Linear Programming

As is mentioned before, linear programming takes the form of

$$\min_{\substack{x \\ Ax=b \\ Gx \preceq d}} c^T x + d \tag{3}$$

We also give the standard form of

$$\min_{\substack{x \\ Ax=d \\ x\succeq 0}} c^T x \tag{4}$$

Here is a proof of why all LPs can be translated into standard form: The proof consists of two parts. Part 1 focuses on eliminating equality constraints. Part 2 demonstrates how to eliminate free variables such that all entries of x could be greater than or equal to 0.
Part 1: The constraint $Ax = b$ could be rewritten as $Ax \preceq b, Ax \succeq b$ and we can revert the direction of $\succeq$ by multiplying both sides with -1, we get $Ax \preceq b, -Ax \preceq -b$.
Part 2: Any real number can be written as the difference of two non-negative numbers. Therefore we can define $x^+ \succeq 0$ and $x^- \succeq 0$ such that $x = x^+ - x^-$. It turns out that LP is a special example of NLP. The key distinct between these two problems seems to be linear and non-linear while in recent years, it is revealed it's actiually convex and non-convex instead.

1.6.2 Why Convexity is so important

Convex functions have the nice property that there is **no** local minima which is not global. There are some techniques that can convexify a non-convex function but no general method is found, i.e. convexify a non-convex function is task-specific. If an optimization problem is proved to be convex, then it is guaranteed a solution could be found. Convex sets also exhibits nice properties: there is no empty relative interior in it and all directions are feasible at any point in the set. That explains why iterative algorithms are widely used: these two nice properties guarantee the iterative algorithms working well since the algorithm could start from any interior point and is capable of searching in any direction until one of the solutions is found.

2 Lec 2: Convex Optimization Pt.2

2.1 Different Forms of Convex Programs

The canonical form of convex program is presented previously. There are several other forms of convex programs.

2.1.1 Epigraph Form

The epigraph form looks like

$$\min_{\substack{x,t \\ f_0(x)\leq t \\ f_i(x)\leq 0, i=1,2,3,\ldots,m \\ Ax=b}} \quad t$$

2.1.2 Unconstrained form

$$\min_x f_0(x) + \mathbb{I}_C(x)$$

2.1.3 Convex Program with a feasible set

$$\min_{x\in C} f_0(x)$$

In general, the unconstrained form is commonly used in convex optimization. It is easier to solve unconstrained problems in the sense that for unconstrained convex functions, we have

$$x^* = \arg\max_x f_0(x) \longleftrightarrow \nabla f_0(x^*) = 0 \tag{5}$$

So the next problem is if we can bridge unconstrained problems with constrained ones?

Proposition 2.1. For constrained problems, we have

$$x^* = \arg\max_{x \in C} f_0(x) \longleftrightarrow \nabla f_0(x^*)^T (x - x^*) \geq 0, \forall x \in C \tag{6}$$

Proof. For the sufficiency, assume $\nabla f_0(x^*)^T(x - x^*) \geq 0$, then $\forall x \in C$, we have $f_0(x) \geq f_0(x^*) + \nabla f_0(x^*)^T(x - x^*) \geq f_0(x^*)$ by the first-order condition of convex function
For necessity, assume x^* to be a minimum point of f_0 within the feasible set C and $\exists x'$ s.t. $\nabla f_0(x^*)^T(x' - x^*) \leq 0$. we already know that

$$\begin{aligned} f_0(x') &\geq f_0(x^*) \\ f_0(x') + \nabla f_0(x^*)^T(x - x^*) &\geq f_0(x^*) \end{aligned}$$

which contradicts to the first-order condition of convex functions. □

The intuition behind the above proposition is all the feasible directions to the minimum point within the feasible set from x^* must be aligned with gradient. However, this proposition doesn't actually well characterizes the relation between the objective function and feasible set(the feasible set is still in the conditions!).And that's where we need the unconstrained form. However, we still have the problem that the indicator function is not differentiable so we need to soften it. Here comes the need for Lagrange Multipliers and Relaxations.

2.2 Lagrange Multipliers and Relaxations

For the canonical form of convex program (2), we assign the so called **Lagrange Multipliers** to each of the constraints:

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \nu^T (Ax - b), \lambda_i \geq 0 \text{ for } i = 1, 2, ..., m \tag{7}$$

And a relaxed problem of the original problem (2) is $\min_x \mathcal{L}(x, \lambda, \nu) = g(\lambda, \nu)$. Let $p^* = f_0(x^*)$. So what's the relation between p^* and $g(\lambda, \nu)$?

2.3 Dual Problem

Proposition 2.2. The relation between p^* and $g(\lambda, \nu)$ is $\forall \lambda \geq 0$ and ν, $g(\lambda, \nu)$ forms a lower bound for p^*

Proof.

$$\begin{aligned} g(\lambda, \nu) &= \inf_x \mathcal{L}(x, \lambda, \nu) \\ &= \inf_x f_0(x) + \sum_i \lambda_i + f_i(x) + \nu^T (Ax - b) \\ &\leq \inf_{x \in C} f_0(x) + \underbrace{\sum_i \lambda_i f_i(x)}_{\lambda_i \geq 0, f_i(x) \leq 0, \lambda_i f_i(x) \geq 0} + \underbrace{\nu^T (Ax - b)}_{Ax - b = 0} \\ &\leq \inf_{x \in C} f_0(x) \\ &= f_0(x^*) \end{aligned}$$

□

That explains why we artificially impose $\lambda_i \geq 0$. Since for any $\lambda \geq 0$ and $\nu, g(\lambda, \nu)$ forms a lower bound on p^*, we can optimize over $g(\lambda, \nu)$ and then try to merge the gap between p^* and $g(\lambda, \nu)$. That leads to the formulation of **dual problem**:

$$d^* = \max_{\lambda \geq 0} g(\lambda, \nu) \tag{8}$$

Also, the original problem 2 is called the **primal problem**.

2.4 Primal and Dual Problem

An exciting fact is that the dual problem is always convex even if the original one is not since the minimum of linear functions is concave and maximum of linear functions is convex. Weak duality stands for $d^* \leq p^*$ and strong duality is $d^* = p^*$. Also, strong duality holds for most convex problems:

Proposition 2.3 (Slater's Condition(Strictly feasible))**.** If a convex optimization problem is strictly feasible, i.e. $\exists x \in relint(C)$ $s.t.$ for all the nonlinear constraints $f_i(x) < 0$ and $Ax - b = 0$, then strong duality holds.

Note that the Slater's condition only implies strong duality for **convex problems**, the other direction doesn't necessarily holds. Here we introduce KKT Conditions that is a necessary and sufficient condition for strong duality.

2.5 KKT Conditions

There are 5 conditions included in KKT. The first two are primal feasibility, i.e., $f_i(x) \leq 0, Ax = b$, the third is dual feasibility $\lambda_i \geq 0$. The remaining two are complementary slackness and stationarity.

2.5.1 Complementary Slackness

$$\begin{aligned}
f_0(x^*) = g(\lambda^*, \nu^*) &= \inf_x \mathcal{L}(x, \lambda^*, \nu^*) \\
&= \inf_x f_0(x) + \sum_i \lambda_i^* + f_i(x) + \nu^{*T}(Ax - b) \\
&\leq \inf_{x \in C} f_0(x) + \underbrace{\sum_i \lambda_i^* f_i(x)}_{\lambda_i^* \geq 0, f_i(x) \leq 0, \lambda_i^* f_i(x) \geq 0} + \underbrace{\nu^{*T}(Ax - b)}_{Ax - b = 0} \\
&\leq \inf_{x \in C} f_0(x) \\
&= f_0(x^*)
\end{aligned}$$

So the inequations should be tight, and that implies

$$\lambda_i^* f_i(x) = 0, i = 1, 2, ...m \tag{9}$$

2.5.2 Stationarity

$f_0(x^*) = g(\lambda^*, \nu^*) = \inf_x \mathcal{L}(x, \lambda^*, \nu^*)$ implies that x^* is the minimizer of $\mathcal{L}(x, \lambda^*, \mu^*)$. Therefore we have $\nabla_x \mathcal{L}(x, \lambda^*, \nu^*) = 0$,i.e.

$$\nabla f_0(x^*) + \sum_i \lambda_i^* \nabla f_i(x^*) + A^T \nu^* = 0 \tag{10}$$

Note that in the context of convex problems, strong duality is equivalent to KKT conditions, i.e. necessity and sufficiency. But for non-convex problems, sufficiency and necessity are considered separately.

1. sufficiency: If (x^*, λ^*, ν^*) satisfies KKT conditions for an optimization problem, then x^* is an optimal point.

2. necessity: If x^* is a minimizer for an optimization problem, then $\exists(\lambda^*, \nu^*)$ that satisfies KKT conditions under some regulairity conditions(e.g. Slater's condition for convex problems).