Optimization Cookbook

Richard Chen*

September 9, 2025

This document is meant to be used as a reference sheet for Economics PhD students on convex optimization. Hopefully, it gives stronger intuition for the KKT conditions and common optimality criteria. It also corrects a common but incorrect interpretation of Newton's method. Any feedback and corrections in this document would be greatly appreciated.

*University of Chicago, Harris School of Public Policy, 1307 E 60th St, Chicago, IL 60637, USA E-mail: `richardchen@uchicago.edu`

Contents

1 Introduction

Much of Economics boils down to optimization problems: consumers maximize utility, firms maximize profit, and social planners maximize social welfare. These (often constrained) optimization problems are usually convex because they arise naturally, for tractability purposes, or both.

To solve these optimization problems, students are taught how to form Lagrangians and use the Karush-Kuhn-Tucker (KKT) conditions to find solution candidates. There isn't enough time in a standard Microeconomics class to give an in-depth discussion of these tools. Nevertheless, a student can get by quite well without going through tedious derivations to prove optimality conditions. However, the lack of intuition (or misgiven intuitions) for these optimization tools may cause students to misapply these techniques both in theory and in empirical settings.

This document has 2 main goals.

1. Give stronger intuition on the KKT conditions

2. Correct a common (but incorrect) interpretation of Newton's method.

1.1 Motivation

What I found most frustrating in my classes was the vagueness of the KKT conditions. The KKT theorem is our workhorse in solving constrained optimization problems in the first year. However the KKT conditions are presented with no proof as to why they define the optimum.

The optimality conditions for an unconstrained problem make intuitive sense - the gradient should be 0 at the optimum. However for a constrained problem, why does taking FOC's of the Lagrangian work? Consider the generic constrained optimization problem

$$\begin{aligned} &\min_x f(x) \\ \text{s.t.} \quad &g(x) = 0 \\ &h(x) \leq 0 \end{aligned}$$

where f(x) and h(x) are convex, and g(x) is an affine function. The Lagrangian for this optimization problem is

$$\mathcal{L}(x, \lambda, \nu) = f(x) + \lambda g(x) + \nu h(x) \tag{1}$$

Hopefully this is nothing new so far. Now to get the optimum for the original problem, we are told to take the FOC of $\mathcal{L}$ with respect to x. This, combined with several other conditions, supposedly give us the optimum. The conditions are

1. Primal Constraints:

$$\begin{aligned} f_i(x) &\leq 0, \quad i = 1, ..., m \\ h_i(x) &= 0, \quad i = 1, ..., p \end{aligned}$$

2. Dual Constraints:

$$\lambda \gtrsim 0$$

3. Complementary Slackness:

$$\lambda_i f_i(x) = 0, \quad i = 1, ..., m$$

4. Lagrangian Optimality:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$$

Often, the λ and ν terms are portrayed as penalty terms for the equality and inequality constraints.

Several things confused me.

- Why does λ have to be weakly positive?
- Why does complementary slackness have to hold?
- Why do the equality constraints have to be affine? Why can't they be any other kind of function?

- If λ, ν are merely penalty terms, how does the FOC of $\mathcal{L}(x, \lambda, \nu)$ **guarantee** that we get the correct optimum? With this interpretation, it seems like we've recast our constrained optimization problem as an unconstrained optimization problem with soft penalties, which means there should be no guarantee that the constraints will hold.

These questions were my motivation in writing this document. I wanted to pin down exactly why KKT works, and give more intuition on when and why it's appropriate to apply. Sections 2-4 walk through and build up to the KKT (although I skip the actual KKT proof its hard and not much is gained from doing it).
Section 5 discusses how to analytically solve these optimization problems. Section 5.3 corrects a common misconception of Newton's Method.

1.2 Mathematical Objects

There are several mathematical objects used in this document which one may not be familiar with. I encourage readers to check the Appendix Section A which give the definitions of these objects. Appendix Section B also gives a formal definition of convexity and how to determine if a function is convex, which may be helpful for understanding the main body of this document.

2 Optimization Problems

We define the standard form of a "primal" optimization problem as the following.

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{s.t.} \quad f_i(x) \leq 0, & \quad i = 1, ..., m \\ h_i(x) = 0, & \quad i = 1, ..., p \end{aligned}$$

- $x \in \mathbb{R}^n$ is the optimization variable (or "primal" variable)
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function
- $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., m$, are the inequality constraint functions
- $h_i : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., p$, are the equality constraint functions

Canonically, optimization problems are framed as minimization problems. One can easily recast it as a maximization by negating the objective function.
We define the optimal value (solution) as $p^* = \inf\{f_0(x) | f_i(x) \leq 0, i = 1, ..., m; h_i(x) = 0, i = 1, ..., p\}$.
The optimal value p^* must lie in the feasible region of the problem. The feasible region is defined through the implicit and explicit constraints of the problem. $f_i(x)$ and $h_i(x)$ act as explicit constraints on our problem. Occasionally, there are also implicit constraints that arise from the domain of our functions. Usually when we refer to constraints, we refer to the explicit constraints.

Example 2.1. Consider the following.

$$\min_x f_0(x) = -\sum_{i=1}^{k} \log(b_i - a_i^T x) \tag{2}$$

This problem has no explicit constraints, but has an implicit constraint. The quantity $b_i - a_i^T x$ must be strictly positive to satisfy the domain of the log function. Thus, the feasible region is any x that satisfies $b_i - a_i^T x > 0$. △

For housekeeping, we define $p^* = \infty$ when the problem is infeasible (the feasible region is empty), and $p^* = -\infty$ when the problem is unbounded below.
We define an x to be locally optimal if $\exists R > 0$ such that x is optimal for

$$\begin{aligned} &\min_z f_0(z) \\ \text{s.t.} \quad & f_i(z) \leq 0, \quad i = 1, ..., m \\ & h_i(z) = 0, \quad i = 1, ..., p \\ & ||z - x||_2 \leq R \end{aligned}$$

2.1 Convex Optimization Problems

We focus on convex optimization problems. We can represent these in the standard form with some slight restrictions.

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{s.t.} \quad f_i(x) \leq 0, & \quad i = 1, ..., m \\ a_i^T x = b_i, & \quad i = 1, ..., p \end{aligned}$$

where the functions $f_i(x), i = 0, 1, ..., m$ are convex, and the equality constraints are affine.

Why do the equality constraints have to be affine? By definition, a convex optimization problem must have a convex feasible region. This explains why the inequality constraints must be convex: $\alpha-$sublevel spaces of convex functions are convex sets, and the intersection of convex sets is convex (see Appendix Section B). Thus, absent of any equality constraints, if all of the inequality constraints are composed of convex functions, the feasible region will be convex.

Now if we add equality constraints, this is akin to drawing a “surface” in the convex feasible region defined by the inequality constraints. If we allow for non-affine functions, this may make it such that the feasible space is no longer convex.

For visualization, consider a sphere as the feasible region defined by inequality constraints. Now consider adding an equality constraint. This can be visualized by slicing the sphere with a plane. The feasible region is now the plane that intersects with the sphere. A plane is convex.

If instead we added a parabolic constraint (as an example of a non-affine equality constraint), the feasible region is now the paraboloid formed by the intersection of the sphere and the parabola, which is a non-convex shape. See Appendix Section B for a discussion on how to determine if a region is convex or not.

One important thing to keep in mind is that we really just need the feasible region to be convex. Usually this is satisfied if the objective function and inequality functions are convex and the equality constraints is affine. These technically aren’t necessary. Consider the following optimization problem.

$$
\begin{aligned}
\min f_0(x) &= x_1^2 + x_2^2 \\
\text{s.t.} \quad f_1(x) &= \frac{x_1}{1 + x_2^2} \le 0 \\
h_1(x) &= (x_1 + x_2)^2 = 0
\end{aligned}
$$

The objective function is convex. The inequality constraints are not convex functions, nor is the equality constraint affine. However note the feasible set, which is $\{(x_1, x_2) | x_1 = -x_2 \le 0\}$. This set is convex, so this is a convex problem! Well, kind of. It can be rewritten as an equivalent but not identical convex problem

$$
\begin{aligned}
\min f_0(x) &= x_1^2 + x_2^2 \\
\text{s.t.} \quad x_1 &\le 0 \\
x_1 + x_2 &= 0
\end{aligned}
$$

One of the main reasons we focus on convex optimization problems is because of their optimality conditions.

Proposition 1. *Any locally optimal point of a convex problem is globally optimal.*

Proof. Suppose x is locally optimal, which means $\exists R > 0$ such that $\forall z$ feasible, $||z - x||_2 \le R \Rightarrow f_0(z) \ge f_0(x)$.
Now suppose x is not globally optimal, which means there is a feasible y such that $f_0(y) < f_0(x)$. Because x is locally optimal, y must not exist in the neighborhood defined by R. Consider a new variable $z' = \theta y + (1 - \theta)x$ where $\theta = \frac{R}{2||y-x||_2}$. Note that $||y - x||_2 > R$ because x is locally optimal, which means that $\frac{1}{2} > \theta > 0$. Additionally, $||z' - x||_2 = R/2 < R$, and by the convexity of the feasible set, z' must also be feasible. By the convexity of f_0, we also have

$$
f_0(z') = f_0(\theta y + (1 - \theta)x) \le \theta f_0(y) + (1 - \theta) f_0(x) < f_0(x) \tag{3}
$$

which contradicts our earlier assumption that x was locally optimal. Thus there cannot exist any feasible y where $f_0(y) < f_0(x)$, and x must be globally optimal. □

Notice how important it was that the feasible region was convex. Because of that, we could use the definition of convexity to prove that a locally optimal point is globally optimal. If the feasible region wasn't convex, then we wouldn't be able to construct those new points to show the contradiction!

2.1.1 Optimality Criterion

Consider Equation 18 from Appendix section B. This can be rewritten as an optimality criterion for a convex optimization problem.

Proposition 2. *Suppose X denotes the feasible set. x is optimal for a convex optimization problem iff $x \in X$*

$$\nabla f_0(x)^T(y-x) \geq 0 \quad \forall y \in X \tag{4}$$

Proof. Suppose $x \in X$ and x is optimal but $\exists y \in X$ such that $\nabla f_0(x)^T(y-x) < 0$. Consider the point $z(t) = ty + (1-t)x$ where $t \in [0,1]$ is a parameter. $z(t)$ is a line segment from x to y, both of which are feasible. The feasible set is convex, which means the entire line segment is feasible. For some small positive t,

$$\frac{d}{dt} f_0(z(t))|_{t=0} = \nabla f_0(x)^T(y-x) < 0 \tag{5}$$

via the chain rule. So for small positive t, we have shown $f_0(z(t)) < f_0(x)$ □

We can use Proposition 2 as the basis to derive many different optimality criteria.

Example 2.2 (Unconstrained Problems). We can derive the optimality condition we're familiar with for unconstrained optimization problems using Equation 4 as our base. Suppose we have an unconstrained problem where x is optimal. For all feasible y, it must be that $\nabla f_0(x)^T(y-x) \geq 0$. f_0 is differentiable, so the domain is open by definition, which means all y close to x are also feasible. Let $y = x - t\nabla f_0(x)$ where $t \in \mathbb{R}$. For a small positive t, y is feasible, and we can rewrite the optimality criterion as

$$\nabla f_0(x)^T(y-x) = \nabla f_0(x)^T(x - t\nabla f_0(x) - x) = -t||\nabla f_0(x)||_2^2 \geq 0$$

where it's obvious that for this condition to hold, we must have $\nabla f_0(x) = 0$. △

Example 2.3 (Equality Constrained Problems). Consider problems that only have equality constraints.

$$\begin{aligned} &\min f_0(x) \\ &\text{s.t.} \quad Ax = b \end{aligned}$$

The feasible set is affine, and the optimality condition for a feasible x is

$$\nabla f_0(x)^T(y - x) \geq 0$$

which must hold for all y in the feasible affine set. x is feasible, so every feasible y must have the form $y = x + v$ for some $v \in \mathcal{N}(A)$. This is more obvious if you write it out. $Ax = b$ must hold. y is feasible, so the following must hold.

$$\begin{aligned} Ay &= b \\ A(x + v) &= b \\ Ax + Av &= b \\ b + Av &= b \end{aligned}$$

The last line shows that for y to be feasible, $Av = 0$ must hold, which means v is in the null space of A. So we can rewrite the optimality condition.

$$\nabla f_0(x)^T(x + v - x) = \nabla f_0(x)^T v \geq 0 \quad \forall v \in \mathcal{N}(A)$$

Note that $\mathcal{N}(A)$ is a vector subspace, which means that for any vector $v \in \mathcal{N}(A)$, the vector $-v \in \mathcal{N}(A)$ also holds. So if $\nabla f_0(x)^T v \geq 0$ for all v, then we must have $\nabla f_0(x)^T v = 0$ for all $v \in \mathcal{N}(A)$. This means that $\nabla f_0(x) \perp \mathcal{N}(A)$.
Now we can use the property that $\mathcal{N}(A)^\perp = \mathcal{R}(A^T)$, so $\nabla f_0(x) \in \mathcal{R}(A^T)$. In other words, there exists a $v \in \mathbb{R}^p$ such that

$$\nabla f_0(x) + A^T v = 0$$

Together with $Ax = b$, these are the classical Lagrange multiplier optimality conditions.[1]. Also note that $A^T v = 0$ must hold, which means that $\nabla f_0(x) = 0$ must also hold, which is the same optimality criterion for an unconstrained problem. △

[1] If you write the Lagrangian $L(x, \nu) = f_0(x) + \nu(Ax - b)$, taking the FOC gives the equation $\nabla f_0(x) + A^T v = 0$

Example 2.4 (**Nonnegative Orthant**)**.** Consider the problem

$$\begin{aligned} &\min f_0(x) \\ &\text{s.t.} \quad x \succsim 0 \end{aligned}$$

The optimality criterion can be written as

$$\begin{aligned} &x \succsim 0 \\ &\nabla f_0(x)^T(y-x) \geq 0 \quad \forall y \succsim 0 \end{aligned}$$

The term $\nabla f_0(x)^T y$ is unbounded below unless $\nabla f_0(x) \succsim 0$. This is one condition. Now consider the second term, $-\nabla f_0(x)^T x \geq 0$. We must have $x \succsim 0$ from the original statement and $\nabla f_0(x) \succsim 0$ from earlier the condition on $\nabla f_0(x)^T y$, so for this last condition to hold, we must have $\nabla f_0(x)^T x = 0$.

$$\sum_{i=1}^{n} (\nabla f_0(x))_i x_i = 0$$

Each term in this sum is the product of 2 nonnegative numbers, so each term must be 0, so this can be rewritten as

$$\nabla f_0(x)_i x_i = 0 \quad \forall i = 1, ..., n$$

This is the (hopefully!) familiar complementary slackness condition. △

3 Duality

This section explains why we use a Lagrangian in the first place. Consider our standard form problem.

$$\begin{aligned} &\text{minimize} \quad f_0(x) \\ &\text{s.t.} \quad f_i(x) \leq 0, \quad i = 1, ..., m \\ &\quad h_i(x) = 0, \quad i = 1, ..., p \end{aligned}$$

We have not restricted this to be convex yet. We can construct the Lagrangian $\mathcal{L}$: $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$.

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

Hopefully this looks familiar. The vectors λ, ν are the dual variables associated of the standard form problem.

Now we can introduce the Lagrange dual function (aka dual function) g: $\mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ which is the minimum value of the Lagrangian over x for $\lambda \in \mathbb{R}^m$, $\nu \in \mathbb{R}^p$

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} (f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{6}$$

As a formality, we say define the dual function to be $-\infty$ when the Lagrangian is unbounded below in x. Note that the dual function is the pointwise infimum of a family of affine functions of λ, ν, so even when the original problem is not convex, the dual function is always concave.

Denote p^* as the optimum of the standard form problem. The dual function yields a lower bound on p^*.

$$g(\lambda, \nu) \leq p^* \quad \forall \lambda \succsim 0, v \tag{7}$$

This is easy to see. Suppose $\tilde{x}$ is feasible. This means $f_i(\tilde{x})$ and $h_i(\tilde{x}) = 0$, and $\lambda \succsim 0$. Thus

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} \nu_i h_i(\tilde{x}) \leq 0$$

This naturally leads to

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) \leq f_0(\tilde{x})$$

We can interpret the Lagrangian and lower bound property as a linear approximation. Note that because we want to use this dual function as a lower bound, we need $\lambda \succsim 0$ to hold because we are restricting the inequality constraint $f_i(x)$ to be negative, and we want the quantity $\lambda \cdot f_i(x)$ to be negative.

This is the motivator for why we restrict nonnegativity of λ. We don't have the same restriction for the equality constraint multiplier ν because those constraints should add up to 0, so the sign doesn't matter.

Note that the dual function was constructed without any assumptions about convexity for the primal problem.
Let's walk through some examples of constructing dual functions.

Example 3.1 (**Least-norm solution of linear equations**)**.** Consider the problem

$$\begin{aligned} &\min x^T x \\ &\text{s.t.} \quad Ax = b \end{aligned}$$

The Lagrangian is $L(x,\nu) = x^T x + \nu^T(Ax - b)$, and the dual function will be expressed as

$$g(\nu) = \inf_x x^T x + \nu^T(Ax - b)$$

We denote this quantity as f(x) and now use the optimality conditions derived above for an unconstrained optimization problem.

$$\begin{aligned} \nabla f(x) &= 0 \\ 2x + A^T\nu &= 0 \\ x &= -\frac{1}{2}A^T\nu \end{aligned}$$

We can replace this into the dual function to write

$$\begin{aligned} g(\nu) = \inf_x L(-\frac{1}{2}A^T\nu, \nu) &= -\frac{1}{2}(A^T\nu)^T(-\frac{1}{2})(A^T\nu) + \nu^T(A(-\frac{1}{2}A^T\nu - b)) \\ &= \frac{1}{4}\nu^T AA^T\nu - \frac{1}{2}\nu^T AA^T\nu - \nu^T b \\ &= -\frac{1}{4}\nu^T AA^T\nu - b^T\nu \quad \forall\nu \end{aligned}$$

So we have found a lower bound for p^* △

Example 3.2 (**Standard Form Linear Program**)**.** Consider the problem

$$\begin{aligned} &\min c^T x \\ &\text{s.t.} \quad Ax = b \\ &x \succsim 0 \end{aligned}$$

The Lagrangian is given by $L(x, \lambda, \nu) = c^T x + \nu^T(Ax - b) - \lambda^T x = -b^T\nu + (c + A^T\nu - \lambda)^T x$. The dual function $g(\lambda, \nu)$ is the infimum of this which we will have to split into cases. Notice that if $c + A^T\nu - \lambda) \neq 0$, the Lagrangian is unbounded from below. In this case, the infimum will be $-\infty$. If $c + A^T\nu - \lambda) = 0$, the infimum will be $-b^T\nu$.

We are flexible with what the variables ν, λ are, but λ in particular has to be nonnegative. This means that when $c + A^T\nu \succsim 0$, we can find a λ to make the term $c + A^T\nu - \lambda$ equal to 0. If not, then the problem will be unbounded. These observations give us the dual function

$$g(\lambda, \nu) = \begin{cases} -b^T\nu & c + A^T\nu \succsim 0, \\ -\infty & \text{otherwise} \end{cases}$$

△

3.0.1 The Dual Problem

Now we can introduce the Lagrange Dual Problem.

$$\begin{aligned} &\max g(\lambda, \nu) \\ &\text{s.t.} \quad \lambda \succsim 0 \end{aligned}$$

The solution to this problem gives the best lower bound on p^*. Regardless of the convexity of the primal problem, the Lagrange dual problem is always concave in λ and ν because it is the infimum of affine functions. This means it is always a convex optimization problem, and we denote its optimal value with d^*.

We make a distinction between weak and strong duality.

Weak Duality: $d^* \leq p^*$

Weak Duality holds for both convex and nonconvex problems and it can be used to find nontrivial lower bounds for hard problems.

Strong Duality: $d^* = p^*$

This doesn't hold in general, but almost always holds for convex problems. The conditions that guarantee strong duality in convex problems are called constraint qualifications.

3.1 Slater's Condition

One simple (relative) one that we focus on is Slater's Constraint Qualification, which states that $\exists x \in \mathbf{relint}\mathcal{D}$ (relint is the relative interior) such that

$$f_i(x) < 0 \quad i = 1, ..., m, \quad Ax = b$$

This point is strictly feasible. Slater's theorem states that if this point exists, then strong duality holds.

The proof for Slater's Theorem is quite complex...I will not go into it. Note that if an inequality constraint is affine, then it is allowed to hold with equality and the theorem will still hold.

Let's restate the primal problem, dual function, and dual problem for Linear Programs as an example. Recall that the primal problem is

$$\begin{aligned} &\min c^T x \\ \text{s.t.} \quad & b \succsim Ax \end{aligned}$$

The dual function is given by

$$g(\lambda) = \inf_x ((c + A^T\lambda)^T x - b^T\lambda) = \begin{cases} -b^T\lambda & A^T\lambda + c = 0, \\ -\infty & \text{otherwise} \end{cases}$$

The dual problem is

$$\begin{aligned} &\max -b^T\lambda \\ \text{s.t.} \quad & A^T\lambda + c = 0 \\ & \lambda \succsim 0 \end{aligned}$$

Slater's condition tells us that $p^* = d^*$ if $b \succ A\tilde{x}$ for some $\tilde{x}$. Restated more simply: Slater's condition tells us that if there exists a point in the relative interior (not on the boundary) of the feasible region, then strong duality holds. If strong duality holds, that means the solution to the Lagrange dual problem gives us the best possible lower bound of the primal problem. **This is why taking the FOC of the Lagrangian in KKT works and gives us the correct primal solution**.

In practice, why do we like the dual function? There are several reasons. One is that if

I give you a feasible x for the primal problem, how do you verify that it's optimal? You have to check that it's better than every feasible x. However if I also give you the solution to the dual problem, λ^*, you can simply check that $b^T\lambda^* = c^Tx^*$. If that's true, you're optimal.
One quick example.

Example 3.3. Consider the Quadratic Problem.

$$\begin{gathered} \min x^T P x \\ \text{s.t.} \quad b \succsim Ax \end{gathered}$$

where P is a positive definite matrix. The Lagrangian is given by $L(x, \lambda) = x^T P x + \lambda^T (Ax - b)$. The dual function is

$$g(\lambda) = \inf_x L(x, \lambda) = \inf_x (x^T P x + \lambda^T (Ax - b))$$

Note that in this case, the Lagrangian is convex in x and is unconstrained, which means we can apply the FOC optimality condition for the dual function.

$$\begin{aligned} \nabla L(x, \lambda) &= 0 \\ 2Px + A^T\lambda &= 0 \\ x &= -\frac{1}{2}P^{-1}A^T\lambda \\ g(\lambda) = \inf_x ((-\frac{1}{2}P^{-1}A^T)^T P(-\frac{1}{2}P^{-1}A^T) &+ \lambda^T(A(-\frac{1}{2}P^{-1}A^T) - b)) \\ &= -\frac{1}{4}\lambda^T A P^{-1} A^T \lambda - \lambda^T b \end{aligned}$$

The dual problem is thus

$$\begin{gathered} \max_\lambda g(\lambda) = -\frac{1}{4}\lambda^T A P^{-1} A^T \lambda - \lambda^T b \\ \text{s.t.} \quad \lambda \succsim 0 \end{gathered}$$

Under Slater, there exists $\tilde{x}$ such that $A\tilde{x} < b$, so $d^* = p^*$ △

4 Karush-Kuhn-Tucker (KKT) Conditions

The KKT theorem basically states that the following 4 KKT conditions are sufficient conditions for x^* to be a solution to a given optimization problem.

1. Primal Constraints:

$$\begin{aligned} f_i(x) \leq 0, &\quad i = 1, ..., m \\ h_i(x) = 0, &\quad i = 1, ..., p \end{aligned}$$

2. Dual Constraints:

$$\lambda \succsim 0$$

3. Complementary Slackness:

$$\lambda_i f_i(x) = 0, \quad i = 1, ..., m$$

4. Lagrangian Optimality:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$$

Notice that we have touched on all of these conditions in the previous sections.

1. Primal constraints are just restated. Nothing fancy here.

2. Dual constraints comes from an argument similar to the discussion centered on Equation 6

3. Complementary Slackness comes from an argument similar to the discussion about the non-negative orthant in section 2.4

4. Lagrangian Optimality comes from the fact that strong duality holds, so solving the Lagrangian dual problem gives us the same $d^* = p^*$ as the original primal problem, which means the FOC for the Lagrangian dual problem must hold.

The actual proof of the KKT conditions is a little bit of a hassle, but I believe we have built up all of the intuition for every condition. So how can you check when to use KKT? You can more or less use the following steps.

1. Check if the problem is convex (convex feasible region and convex objective function).
2. Check if strong duality holds (can a point exist in the interior of the feasible convex region as defined by the constraints).
3. If strong duality holds, the KKT conditions are sufficient to prove the optimality of a point x^*
4. Do algebra or what not to solve for the optimum.
5. Celebrate

5 Descent Methods

Many convex optimizations we look at will not have analytic solutions. Instead we must use iterative methods to find the optimum p^*. All descent methods have the same general form.

$$x^{k+1} = x^k + t^k \Delta x^k \tag{8}$$

where k denotes a step number in the sequence, and x^k forms a sequence towards the global optimum. t^k is commonly known as the step length or step size. Δx^k is commonly known as the step (or "search") direction. Because these are descent methods, we have

$$f(x^{k+1}) < f(x^k)$$

Convexity tells us $\Delta f(x^k)^T(y - x^k) \geq 0$ from Equation 4, which implies that $f(y) \geq f(x^k)$. This gives us a condition for the search direction in a descent method, namely

$$\nabla f(x^k)^T \Delta x^k < 0 \tag{9}$$

This is also clear from taking a Taylor expansion at x^{k+1} and noting that in order to be a descent method, we must have $f(x^{k+1}) < f(x^k)$.

$$f(x^{k+1}) \approx f(x^k) + \nabla f(x^k)^T (x^{k+1} - x^k)$$
$$f(x^{k+1}) - f(x^k) \approx \nabla f(x^k)^T \Delta x^k < 0$$

A geometric interpretation is that the search direction must make an acute angle with the negative gradient.

I introduce a general descent method algorithm.

Given a starting point $x \in dom f$
Repeat

1. Determine a descent direction Δx
2. Line search: Choose a step size $t > 0$
3. Update: $x = x + t\Delta x$

until a stopping criterion is reached. A typical stopping criterion is $||\nabla f(x)|_2 \leq \eta$ where η is some small number (something like 1e6), because a necessary and sufficient condition for a convex problem is for the gradient to be 0 as we showed previously.

5.1 Line Search Methods

How do we determine the step size of the descent algorithm? Almost all open source optimization libraries include a line search algorithm such that you, as the user, don't have to think about it. You can read this section in case you're just curious about what's going on behind the scenes.

One line search method is the exact line search, where t is chosen to minimize f along the ray $\{x + t\Delta x | t \geq 0\}$.

$$t = \text{argmin}_{s \geq 0} f(x + s\Delta x)$$

This is generally not used because if the computational cost of solving this sub-optimization problem is not low compared to computing the search direction, it's too costly. There are some cases where the minimizer can be found analytically, or others where it can be computed efficiently.

More commonly, line searches are inexact where a step length is chosen to approximately minimize f. There are many inexact line search methods, one of which is the backtracking line search.

Define constants, $0 < \alpha < 0.5$ and $0 < \beta < 1$.

```
Given a descent direction $\Delta x$ for f at $x \in \mathrm{dom} f$, $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$, t=1
while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$ do
    $t = \beta t$.
end while
```

Δx is a descent direction and we have that $\nabla f(x)^T \Delta x < 0$, so for small enough t, we have

$$f(x + t\Delta x) \approx f(x) + t\nabla f(x)^T \Delta x < f(x) + \alpha t \nabla f(x)^T \Delta x$$

So backtracking line search eventually terminates

5.2 Gradient Descent

A natural choice for the search direction is the negative of the gradient: $\Delta x = -\nabla f(x)$. This satisfies our condition for search directions

$$\begin{gathered} \nabla f(x^k)^T \Delta x^k < 0 \\ -\nabla f(x^k)^T \nabla f(x^k) < 0 \\ -||\nabla f(x^k)||_2 < 0 \end{gathered}$$

I skip the proof for convergence analysis for exact line searches, but present the bound showed in Boyd's Convex Optimization textbook. They show that $f(x^k) - p^* \leq \varepsilon$ must be satisfied after at most

$$\frac{log(f(x^0) - p^*)/\varepsilon}{log(1/c)}$$

iterations of the gradient method, where $c = 1 - m/M < 1$, m and M are the lower and upper bounds of the Hessian of f respectively.

The gradient method requires a large number of iterations when the Hessian of f has a large condition number (M/m) close to the optimum. The bound shows the error converges to zero at least as fast as a geometric series, and is called linear convergence because the error lies below a line on a log-linear plot of error vs. iteration number.

For the backtracking line search, the convergence analysis shows that

$$f(x^k) - p^* \leq c^k(f(x^0) - p^*)$$

where $c = 1 - \min\{2m\alpha, 2\beta/M\} < 1$. So we get convergence at least as fast as a geometric series with an exponent that depends on the condition number M/m. This convergence is at least linear.

One good take-away is that the condition number has a very strong influence on convergence rate. It's hard to get the exact bounds m and M, but you can maybe approximate them, and it could be helpful in seeing how you can bound the number of iterations required given a Hessian.

Some other takeaways are that

- gradient methods often show approximately linear convergence
- The backtracking parameters α, β have noticable but not dramatic effect on the convergence. Exact line search can improve convergence of the gradient method at times, but the effect is not large.
- Convergence rate depends greatly on the condition number of the Hessian.

5.3 Newton's Method

For $x \in dom f$, we denote

$$\Delta x_{nt} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

as the Newton step. Because this is a convex problem (and we have restricted our analysis to strongly convex functions in this section), we know that $\nabla^2 f(x)$ is positive definite which implies

$$\nabla f(x)^T \Delta x_{nt} = -\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) < 0$$

So unless $\nabla f(x) = 0$, the Newton step is a descent direction.

Important note: A common misinterpretation of Newton's method is that it is a "type" of gradient descent method, where the step size is determined by the Hessian. **This is incorrect.**

The Hessian provides an amendment to the step direction and while there is a connection (note that when the Hessian is the identity, it reduces to gradient descent), this is a different method for finding step directions. Line searches are still necessary.

Newton's method is divided into two phases: the damped phase, and the quadratically convergent phase. In both phases, the update direction is defined as

$$\Delta x_{nt} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

such that Newton's method updates each step with the following[2]

$$x_{t+1} = x_t + \lambda \Delta x_{nt}$$

At each iteration regardless of the phase, the step direction is strictly Δx_{nt}, and the step size is λ, which must be solved for using a backtracking line search.

In the damped phase, $\lambda \neq 1$ and you still have to do a line search. When x_t is close enough to the optimum, we enter the quadratically convergent phase and $\lambda = 1$ because of the second order approximation (described below) and Δ_{nt} is an exact approximation for the solution.

However, the interpretation should still be that the step direction is $\Delta x_{nt} = -\nabla^2 f(x)^{-1} \nabla f(x)$, while the step size is $\lambda = 1$.

Another way of thinking about it is that the Newton step multiplies the gradient with the inverse Hessian, and not the norm of the inverse Hessian. So the inverse Hessian changes the direction of the gradient and the term $-\nabla^2 f(x)^{-1} \nabla f(x)$ is an entirely different update direction altogether compared to $-\nabla f(x)$. This is why another constant, λ is needed for the step size.

Minimizer of second-order approximation. We write the second order Taylor approximation of f at x.

$$f(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

This is a convex quadratic function of v and is minimized when $v = \Delta x_{nt}$. This interpretation provides some nice insight. When f is quadratic, $x + \Delta_{nt}$ is an exact minimzer of f. Because we used a taylor approximation, when x is close to x^*, $x + \Delta x_{nt}$ is a pretty good

[2] Boyd - Convex Optimization, Section 9.5.2

guess of x^*, and not so much when far from the optimum.

Consider the optimality condition

$$\nabla f(x^*) = 0 \tag{10}$$

$$\nabla f(x+v) \approx \nabla f(x) + \nabla^2 f(x) v = 0 \tag{11}$$

In the last line, we use a Taylor approximation and again get intuition that when close to the optimum, the newton step is what must be added so the linearized optimality condition holds. This is exactly where we enter the quadratically convergent phase of Newton's method, such that the step size is 1. If we weren't close enough to the optimum to do a valid Taylor approximation, then we would be in the damped phase and would need to do a line search.

Convergence Analysis: I skip convergence analysis right now. It's annoying. Takeaways:

- Convergence of Newton's method is very fast and quadratic close to the optimum. Once the quadratic convergence phase is reached, at about 6 iterations should be enough to produce a very high accuracy solution
- Newton's method is affine invariant and insensitive to the choice of coordinates
- Newton's method scales well with problem size (IN TERMS OF STEP SIZE, NOT COMPUTATION)
- Good performance of Newton's method is not dependent on the choice of algorithm parameters

Appendix

A Basic Objects

I introduce mathematical objects commonly used in optimization problems.
Lines: A line through x_1, x_2 is defined as

$$x = \theta x_1 + (1 - \theta)x_2 \tag{12}$$

where $\theta \in \mathbb{R}$. Lines are useful in defining affine sets.
Affine Sets: An affine set is defined to contain the line through any two distinct points in the set. An example of an affine set would be $\{x \in \mathbb{R}^n | Ax = b\}$. A minor distinction (that's not very important) is that a vector space is an affine set, but an affine set is not necessarily a vector space. Vector spaces must pass contain the origin (contains the zero vector).
Line Segment: A line segment is the same as a line except θ is restricted to be in $[0, 1]$. With these definitions, we can now define a convex set.
Convex Sets: A convex set contains the line segment between any two points in the set. So a set C is convex if and only if

$$x_1, x_2 \in C \Rightarrow \theta x_1 + (1 - \theta)x_2 \in C \tag{13}$$

for $\theta \in [0, 1]$. One question you may have is "Why is convexity defined with a line segment? Why not with some other object like a curve, or part of a parabola, or something even more wacky?" I don't have the most satisfactory answer to this other than the observation that there is an intimate relationship between convexity and the triangle inequality. And the triangle inequality is a very helpful property when we create sets.
Cones: A cone is defined with the following equation.

$$x = \theta_1 x_1 + \theta_2 x_2 \quad \theta_1, \theta_2 \geq 0 \tag{14}$$

Notice that this implies the origin must lie in the cone. In practice, it's not really necessary to understand what a cone is.

Dual Cones: The dual cone of a cone K is defined as

$$K^* = \{y | y^T x \geq 0, \quad \forall x \in K\} \tag{15}$$

We are interested in dual cones because a dual cone is always convex even when the original cone K isn't.

Hyperplanes: A hyperplane is a set of the form

$$\{x | a^T x = b\} \quad a \neq 0 \tag{16}$$

where a represents the normal vector that defines the hyper plane. A hyperplane defines 2 half spaces, one above and below the hyper plane.

Halfspaces: A halfspace is a set of the form

$$\{x | a^T x \leq b\} \quad a \neq 0 \tag{17}$$

Notice that the half-space is defined by the hyperplane $a^T x = b$. A half-space is always convex. Hyperplanes and halfspaces are pretty important concepts.

Epigraphs: The epigraph of a function f is defined as the following.

$$\textbf{epi} f = \{(x, t) \in \mathbf{R}^{n+1} | x \in \textbf{dom} f, f(x) \leq t\}$$

Conceptually, this is just everything that's "above" a function.

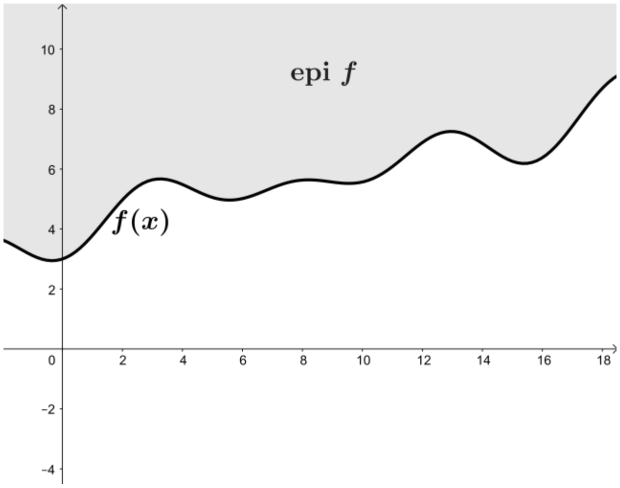

Figure 1: Visualization of the epigraph of a function f. Screenshot from Wikipedia.

B Convexity

We focus on convex problems and need to equip ourselves with a toolkit for identifying convex problems (i.e. identifying convex functions).

B.1 Is it Convex?

Here are some useful tricks for identifying a convex function. The most obvious and commonly used one is checking if the secant line between two points lies completely above the function.
A very useful one is the fact that a function is convex if and only if the epigraph is a convex set. The most common way of checking if a set is convex is applying the definition of convexity: checking if it contains all line segments of its points. To do this, choose two arbitrary points and check if the line segment lies completely in the set or not.
One is restricting the function to a line.

Proposition 3. *$f : \mathbb{R}^n \to \mathbb{R}$ is convex iff the function $g : \mathbb{R} \to \mathbb{R}$*

$$g(t) = f(x = tv) \quad dom(g) = \{t | x + tv \in dom(f)\}$$

is convex in t for any x in the domain of f,$v \in \mathbb{R}^n$

Proposition 4. ***First-Order Condition**: If f is differentiable, then f is convex if and only if the domain of f is convex, and*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \tag{18}$$

I skip the proof for brevity.
Second-Order Condition:

Proposition 5. *If f is twice differentiable with a convex domain, then f is convex if and only if*

$$\nabla^2 f(x) \gtrsim 0 \quad \forall x \in \mathbf{dom} f$$

In other words, the first-order Taylor approximation of a function is always a global underestimator of the function if the function is convex.

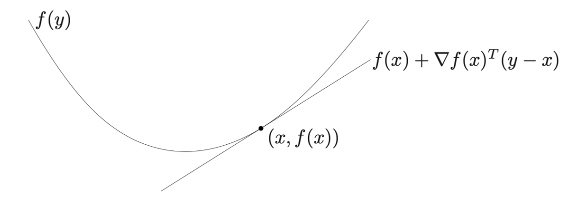

Figure 2: Visualization of convexity for a differentiable function. Screenshot taken from Boyd.

The $\alpha-$sublevel set of a function is defined as

$$C_\alpha = \{x \in \text{dom} f | f(x) \leq \alpha\} \tag{19}$$

Sublevel sets of convex functions are convex. If all of the sublevel sets of a function are convex, the function is not necessarily convex.

B.2 Operations preserving convexity

1. Nonnegative weighted sum of convex functions
2. Intersection of an arbitrary number of convex sets is convex
3. The point-wise max of convex functions is convex
4. The point-wise sup of convex functions is convex.
5. Norms are always convex

C The Supporting Hyperplane Theorem

A hyperplane can be defined in several ways. A vector $\vec{p} \in \mathbb{R}^n$ and a point $c \in \mathbb{R}$ completely defines the Hyperplane $H_{p,c} = \{z \in \mathbb{R}^n ; \vec{z} \cdot \vec{p} = c\}$.

Theorem 1 (Supporting Hyperplane Theorem)**.** *If Y is a nonempty, convex subset of $\mathbb{R}^n$, and there is a point y^o on the boundary of Y, then there is a vector $\vec{p} \neq 0$ such that*

1. *$\forall y \in Y$, $\vec{p} \cdot y \leq \vec{p} \cdot y^0$*
2. *$\forall y \in intY$, $\vec{p} \cdot y < \vec{p} \cdot y^0$*

The intuition behind this theorem is that for any convex set, for any point on the boundary, there exists a hyperplane such that the entire convex set lies on either the upper or lower half-space of the hyperplane. As stated above, a hyperplane is defined completely by a normal vector and a point on the hyperplane. By convention, this normal vector points away from the convex set (so that the convex set lies in the lower half-space).

I personally find this theorem to be confusing because of what I claim to be the abuse of vector notation. My main point of contention was how can you take the dot product of the normal vector $\vec{p}$ with a point on the boundary? That didn't make any sense to me. The statement of this theorem is correct; there's nothing wrong with it. But the way many sources usually state it was confusing to me.
I rewrite the statement like so, which I find much more intuitive.

$$\vec{p} \cdot (y - y^0) \leq 0$$
$$||\vec{p}|| * ||y - y^0|| \cos\theta \leq 0$$

The term $y - y^0$ is obviously a vector, and is pointing from y^0 to y. Then I use the geometric definition of a dot product. Consider the figure below.

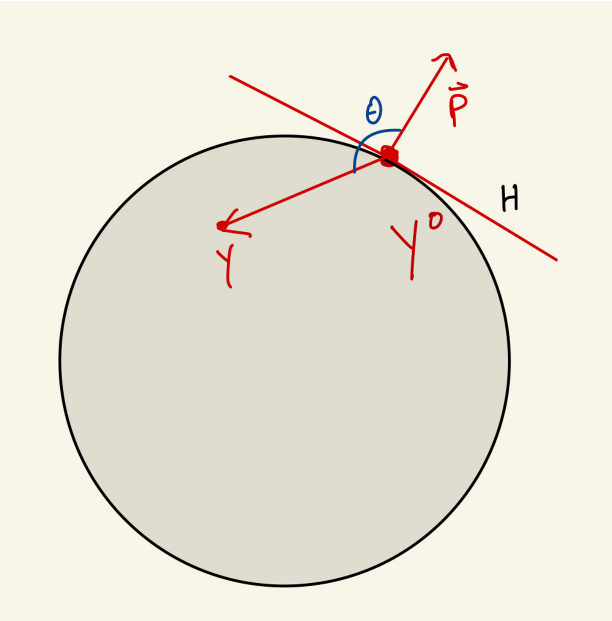

Figure 3: Convex set in grey, with an interior point y and boundary point y_0. The $\vec{p}$ vector is normal to the Supporting Hyperplane.

Consider the angle between the vector $\vec{p}$, and the vector $y - y^0$. The cosine of any angle between 90^o and 270^o will be negative, so the inequality from the Supporting Hyperplane Theorem holds.

With this geometric intuition, we can return to the original theorem and notice that as long as we define y and y^0 to be vectors in reference to any common point in space, the inequality will hold as well. It just becomes more clear when we subtract the points from each other and look at the vector defined by their difference.

Fun Tidbit: This implies that a convex set is completely defined by the union of half spaces that define the boundary of the set.