Convex Optimization Overview

Conner DiPaolo

Contents

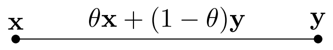


Figure 1: **Convex Combination**. The line segment between $\mathbf{x}$ and $\mathbf{y}$ above represents every possible combination $\{\theta\mathbf{x} + (1-\theta)\mathbf{y} : 0 \leq \theta \leq 1\}$.

1 Introduction

Convex optimization in a large way influences the way people think about and phrase machine learning problems. Almost all problems we will see in our studies are developed or can be viewed as optimization problems. Some problems, like the Support Vector Machine you will all see in the coming weeks, are almost entirely based in the heart of convex optimization.

We don't plan on bringing you all up to speed completely on the art of Convex Optimization, but hopefully in two sections you will know enough to be able to think about problems in new, interesting ways that will aid your studies in and out of machine learning.

The only real prerequisite for this material is a strong confidence in linear algebra and familiarity with matrix calculus. Note that much of this material stems from Boyd and Vandenberghe's insanely influential textbook *Convex Optimization.* If you are interested in the topic or need a more in-depth resource, check out the book. It's free online[1].

2 Convex Sets

The first step into examining convexity is defining what a **convex set** is when given, for example, a subset of the real numbers, or the set of matrices.

Definition 2.1 (Convex Combination)**.** *In the $n = 2$ case, the convex combination of points $\mathbf{x}$ and $\mathbf{y}$ in an affine space is*

$$\theta\mathbf{x} + (1-\theta)\mathbf{y}$$

where $0 \leq \theta \leq 1$. Intuitively, this is the line segment between $\mathbf{x}$ and $\mathbf{y}$ (consider $\theta = 0$ and $\theta = 1$), as seen in Figure 1. More generally, a convex combination of points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ in an affine space (vector spaces included) is the combination

$$\boldsymbol{\theta}_1\mathbf{x}_1 + \boldsymbol{\theta}_2\mathbf{x}_2 + \cdots + \boldsymbol{\theta}_n\mathbf{x}_n = \boldsymbol{\theta}^\top\mathbf{x}$$

where $\theta_i \geq 0$ and $\mathbf{1}^\top\theta = 1$. That is, $\boldsymbol{\theta}$ lies on the standard probability simplex.

With this definition of a convex combination we can define a convex set:

Definition 2.2 (Convex Set)**.** *A set C is convex if, given $\mathbf{x}, \mathbf{y} \in C$, every convex combination of $\mathbf{x}$ and $\mathbf{y}$ is still in C. Mathematically,*

$$\theta\mathbf{x} + (1-\theta)\mathbf{y} \in C. \qquad (0 \leq \theta \leq 1)$$

See Figure 2 for an illustration. Intuitively, this means that every line segment between any two points in C is contained entirely within C.

2.1 Examples of Convex and Non-Convex Sets

Example 2.1 (All of $\mathbb{R}^n$)**.** *$\mathbb{R}^n$ is convex.*

Proof. As a vector space, for any $\theta_1, \theta_2 \in \mathbb{R}$ and $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$,

$$\theta_1\mathbf{x} + \theta_2\mathbf{y} \in \mathbb{R}^n.$$

Thus, restricting $\theta_2 = 1 - \theta_1$ and $0 \leq \theta_1 \leq 1$ does not change this fact, and any convex combination is also in $\mathbb{R}^n$, making the set convex by definition. ■

[1] `http://stanford.edu/~boyd/cvxbook/`

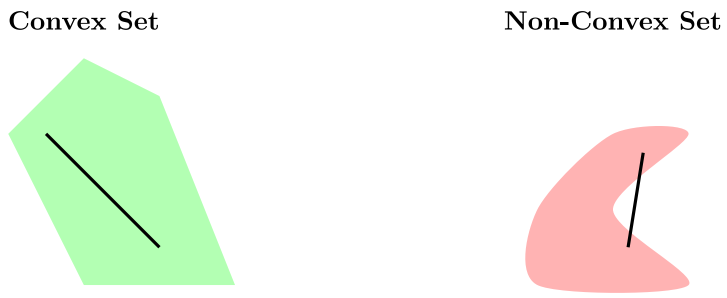


Figure 2: **Set convexity.** Intuitively, a set is convex if a line drawn between any two points within the set lies completely in the set. The convex set to the left is a *convex hull* of it's vertices, meaning the set is constructed as all possible convex combinations of its vertices. These sets are subsets of $\mathbb{R}^2$.

Example 2.2 (The Non-Negative Orthant $\mathbb{R}^n_+$)**.** *The set of all vectors*

$$\{\mathbf{x} : \mathbf{x} \in \mathbb{R}^n \text{ and } \mathbf{x}_i \geq 0\}$$

is convex.

Proof. Left as an exercise to the reader. ∎

Example 2.3 (Closed Intervals in $\mathbb{R}$)**.** *Let $C = [a, b]$ be a subset of the real numbers where $a \leq b$. Then C is convex.*

Proof. Suppose, without loss of generality, that $x_1 \leq x_2$ where $x_1, x_2 \in [a, b]$. Now let $0 \leq \theta \leq 1$. Then

$$\theta x_1 + (1-\theta)x_2 \leq \theta x_2 + (1-\theta)x_2 = x_2$$

because $\theta x_1 \leq \theta x_2$. Similarly,

$$\theta x_1 + (1-\theta)x_2 \geq \theta x_1 + (1-\theta)x_1 = x_1$$

because $(1-\theta)x_2 \geq (1-\theta)x_1$. Thus

$$a \leq x_1 \leq \theta x_1 + (1-\theta)x_2 \leq x_2 \leq b,$$

and hence

$$\theta x_1 + (1-\theta)x_2 \in [a, b].$$

Therefore, by the definition of convexity, $[a, b] \subseteq \mathbb{R}$ is convex for any $a \leq b$. ∎

Example 2.4 (The Set of All Complex Hermetian Matrices)**.** *Let $C = \{A : A \in \mathbb{C}^{n\times n} \text{ and } A^* = A\}$. C is convex.*

Proof. Let $A, B \in \mathbb{C}^{n\times n}$ be Hermitian matrices and $0 \leq \theta \leq 1$. Then

$$\begin{aligned}
(\theta A + (1-\theta)B)^* &= (\theta A)^* + [(1-\theta)B]^* && (1)\\
&= \theta A^* + (1-\theta)B^* && (2)\\
&= \theta A + (1-\theta)B && \text{(because } A^* = A \text{ and } B^* = B\text{)} \quad (3)
\end{aligned}$$

Thus every convex combination of Hermitian matrices is Hermitian, and by the definition of convexity the set is convex. ∎

Example 2.5 (The Set of All Real Symmetric Matrices)**.** *Let $C = \{A : A \in \mathbb{R}^{n\times n} \text{ and } A^\top = A\}$. C is convex.*

Proof. Left as an exercise to the reader. ■

Example 2.6 (The Set of All Linear Matrix Inequalities). *Let A_i and B be symmetric $n \times n$ matrices and $\mathbf{x} \in \mathbb{R}^n$. Let $C = \{\mathbf{x} : A(\mathbf{x}) \preceq B\}$ where $A(\mathbf{x}) = \mathbf{x}_1 A_1 + \cdots + \mathbf{x}_k A_k$. C is convex.*

Proof. Let $0 \leq \theta \leq 1$ and $\mathbf{x}, \mathbf{y} \in C$. Then

$$\begin{aligned} A(\theta\mathbf{x} + (1-\theta)\mathbf{y}) &= \sum_i \left[\theta\mathbf{x}_i + (1-\theta)\mathbf{y}_i\right] A_i \\ &= \theta \sum_i \mathbf{x}_i A_i + (1-\theta) \sum_i \mathbf{x}_i A_i \\ &= \theta A(\mathbf{x}) + (1-\theta) A(\mathbf{y}) \\ &\leq \theta B + (1-\theta) B \\ &= B. \end{aligned}$$

Thus any convex combination of elements of C is contained in C and by definition C is convex. ■

Example 2.7 (The Space of Probability Distributions). *Let $\mathcal{P}$ be the space of continuous probability distributions over $\mathbb{R}^n$. That is, every element of $\mathcal{P}$ defines a unique probability density function $\mathbb{P}(x) \geq 0$ such that*

$$\int_{\mathbb{R}^n} \mathbb{P}(x) dx = 1.$$

$\mathcal{P}$ is convex.

Proof. Let f and h be valid probability distributions from $\mathcal{P}$. That is,

$$f, h \in \left\{ \mathbb{P}(x) : \int_{\mathbb{R}^n} \mathbb{P}(x) dx = 1 \text{ and } \mathbb{P}(x) \geq 0 \right\}.$$

Now let $0 \leq \theta \leq 1$. Then

$$\theta f(x) + (1-\theta) h(x) \geq 0$$

as a positive combination of positive functions. Similarly,

$$\begin{aligned} \int_{\mathbb{R}^n} \left[\theta f(x) + (1-\theta) h(x)\right] dx &= \theta \int_{\mathbb{R}^n} f(x) dx + (1-\theta) \int_{\mathbb{R}^n} h(x) dx \\ &= \theta + (1-\theta) = 1. \end{aligned}$$

Thus every convex combination of probability distributions over $\mathbb{R}^n$ is a valid distribution (often called a mixture), and therefore the set of all valid probability distributions over $\mathbb{R}^n$ is convex itself. ■

Example 2.8 (Disjoint Intervals in $\mathbb{R}$). *Let $N = [a, b] \cup [c, d]$ where $a \leq b < c \leq d$. N is **not** convex.*

Proof. Let $b, c \in N$ be as described above. Then for $0 < \theta < 1$ (not we aren't including inequality),

$$\theta b + (1-\theta) c \notin N.$$

Thus not *every* convex combination of elements in N is in N, and N is not convex. ■

Example 2.9 (The Set of All Stochastic Matrices). *The set of all matrices A such that for all elements $0 \leq A_{ij} \leq 1$ and each row sums to 1,*

$$M = \{A : A \in \mathbb{R}^{m \times n} \text{ and } 0 \leq A_{ij} \leq 1 \text{ and } A\mathbf{1} = \mathbf{1}\},$$

is convex.

Note that this set is the set of all matrix representations of every possible Markov Chain.

Proof. Let $A, B \in M$ and $0 \leq \theta \leq 1$. Then

$$c = [\theta A + (1-\theta)B]_{ij} = \theta A_{ij} + (1-\theta)B_{ij}$$

satisfies $0 \leq c \leq 1$ as $A_{ij}, B_{ij} \in [0,1]$ and closed intervals on $\mathbb{R}$ are convex as seen in Example 2.3.

Further, because $A\mathbf{1} = \mathbf{1}$ and $B\mathbf{1} = \mathbf{1}$,

$$[\theta A + (1-\theta)B]\,\mathbf{1} = \theta A\mathbf{1} + (1-\theta)B\mathbf{1} = \theta\mathbf{1} + (1-\theta)\mathbf{1} = \mathbf{1},$$

as desired. Thus every convex combination of elements withn M remains in M, and by definition the set M is convex. ■

Example 2.10 (Norm Balls). *For any valid norm $||\cdot||$ and scalar $r \geq 0$, the set*

$$C = \{\mathbf{x} : ||\mathbf{x}|| \leq r\}$$

is convex.

Proof. Let $\mathbf{x}$ and $\mathbf{y}$ be elements from C and $0 \leq \theta \leq 1$. Then

$$||\theta\mathbf{x} + (1-\theta)\mathbf{y}|| \leq ||\theta\mathbf{x}|| + ||(1-\theta)\mathbf{y}|| = \theta||\mathbf{x}|| + (1-\theta)||\mathbf{y}|| \leq \theta r + (1-\theta)r = r,$$

as desired, where we used the Triangle Inequality for the first step and the second used the homogeneity of norms. Thus every convex combination of elements in C is within C, and therefore C is convex. ■

3 Convex Functions

You likely have already seen convex functions from your time in Calculus I in high school or something similar. Nevertheless, our treatment will be much more rigorous (though not *too* rigorous) and be very applicable to our later studies.

The most important fact to know about convex functions is that they

(a) Every local minimum is a global minimum

(b) Generally have efficient algorithms for finding such a minimizer.

3.1 Convex and Concave Functions

Our idea of a convex set will be useful in considering convex functions.

Definition 3.1 (Convex). *Given a convex set C, a function $f : C \mapsto \mathbb{R}$ is convex if, for $0 \leq \theta \leq 1$, given any $\mathbf{x}, \mathbf{y} \in C$,*

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}).$$

Intuitively, this means that a convex function always lies under a line between any two points where it is evaluated. This is seen in Figure 3.

Definition 3.2 (Concave Functions). *Given a convex set C, function $f : C \mapsto \mathbb{R}$ is concave if $-f$ is convex. That is, for $0 \leq \theta \leq 1$, given any $\mathbf{x}, \mathbf{y} \in C$,*

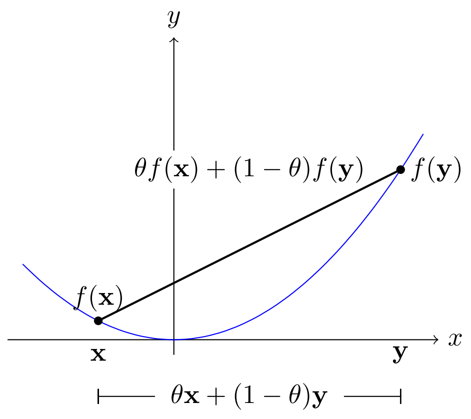
$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \geq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}).$$

Theorem 3.1 (Restriction To a Line). *A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if and only if f is convex when restricted to any line that intersects its domain. Mathematically, f is convex if and only if for all $x \in C$ and any $\mathbf{v} \in \mathbb{R}^n$ and $t \in \mathbb{R}$,*

$$g(t) = f(\mathbf{x} + t\mathbf{v})$$

is convex where $g : \{t : \mathbf{x} + t\mathbf{v} \in C\} \mapsto \mathbb{R}$

Proof. Omitted. See *Convex Optimization* by Boyd and Vandenberghe. ■



y
$\theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y})$
$f(\mathbf{y})$
$f(\mathbf{x})$
x
$\mathbf{x}$
$\mathbf{y}$
$\theta\mathbf{x} + (1-\theta)\mathbf{y}$

Figure 3: **Convex Functions**. This function $f : [a, b] \mapsto \mathbb{R}$ is convex because it's domain is convex and every line between two points on the function lies above the function.

3.2 First Order Conditions: $f(x) \geq f(y) + \nabla f(y)^\top (x - y)$

Generally speaking, determining convexity from the definition is hard. In this section we will develop more tractable conditions that will help us determine if an arbitrary function is convex. The following necessary and sufficient condition for convexity is called the First Order Condition because it relies only on the first derivative.

Theorem 3.2 (First Order Conditions)**.** *Let f be a function mapping from some convex set C to $\mathbb{R}$. Then f is convex if and only if, given $\mathbf{x}$ and $\mathbf{y}$ from C,*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}).$$

Proof. Omitted. See *Convex Optimization* by Boyd and Vandenberghe. ∎

3.2.1 Examples On Determining Convexity

Theorem 3.3 (Linear Functions are Both Concave and Convex)**.** *Given a function $f : C \mapsto \mathbb{R}$ where C is a convex set and for any $a, b \in \mathbb{R}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}$*

$$f(a\mathbf{x} + b\mathbf{y}) = af(\mathbf{x}) + bf(\mathbf{y}),$$

f is both convex and concave. Note that f is the definition of a linear function.

Proof. Let $\mathbf{x}, \mathbf{y}$ be any elements of C and $0 \leq \theta \leq 1$. Then by linearity

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) = \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}).$$

Thus by definition f is both convex and concave. ∎

Example 3.1. *$f(x) = x^2$ is convex.*

Proof. We will use the much more convenient second order condition for this problem later. Consider $x, y \in \mathbb{R}$ and $0 \leq \theta \leq 1$. Then

$$\begin{aligned} f(\theta x + (1-\theta)y) &= (\theta x + (1-\theta)y)^2 \\ &= (\theta x + (1-\theta)y)(\theta x + (1-\theta)y) \\ &= \theta^2 x^2 + 2\theta(1-\theta)xy + (1-\theta)^2 y^2. \end{aligned}$$

f will be convex if and only if

$$\theta f(x) + (1-\theta)f(y) - f(\theta x + (1-\theta)y) \geq 0$$

for all x, y. We have

$$\begin{aligned} g = \theta f(x) + (1-\theta)f(y) - f(\theta x + (1-\theta)y) &= \theta x^2 + (1-\theta)y^2 - \theta^2 x^2 - 2\theta(1-\theta)xy - (1-\theta)^2 y^2 \\ &= \theta(1-\theta)x^2 - 2\theta(1-\theta)xy + \theta(1-\theta)y^2 \\ &= \theta(1-\theta)(x-y)^2 \end{aligned}$$

We know $(x-y)^2 \geq 0$ for all x and y. Similarly, when $\theta \in [0,1]$, both θ and $1-\theta$ are positive, so this expression is positive. Thus f must be convex, as desired. ■

3.3 Second Order Conditions: $\nabla^2 f \succeq 0$

Here we discuss the most useful condition for determining convexity.

Theorem 3.4 (Second Order Conditions)**.** *A twice-differentiable function $f : C \mapsto \mathbb{R}$ is convex if and only if and only if the Hessian*

$$\nabla^2 f \succeq 0$$

Proof. Omitted. ■

3.3.1 More Examples On Determining Convexity

Example 3.2 (Quadratic Functions)**.** *The function $f : \mathbb{R}^n \mapsto \mathbb{R}$ defined by*

$$f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$$

is convex if $A \succeq 0$.

Proof. We know

$$\nabla^2 f = 2A.$$

Similarly we know that $2A \succeq 0$ if and only if $A \succeq 0$. Thus by the second order conditions f is convex if and only if $A \succeq 0$.

We can also show that f is concave if $A \preceq 0$. ■

Example 3.3. *$f(x) = a\, e^x$ is convex if $a \geq 0$.*

Proof. The Hessian

$$\nabla^2 f = \frac{d^2 f}{dx^2} = a\, e^x \geq 0$$

as long as $a \geq 0$. Thus by the second order conditions f is convex if $a \geq 0$. ■

Example 3.4. *The function $f : \mathbb{R}^n \mapsto \mathbb{R}$ defined by*

$$f(\mathbf{x}) = e^{\mathbf{x}^\top \mathbf{x}} = e^{\mathbf{x}_1^2 + \cdots + \mathbf{x}_n^2}$$

is convex.

Proof. Each element of the Hessian

$$\nabla^2 f_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} = 4\mathbf{x}_i \mathbf{x}_j e^{\mathbf{x}^\top \mathbf{x}}.$$

Therefore

$$\nabla^2 f = 4\mathbf{x}\mathbf{x}^\top e^{\mathbf{x}^\top \mathbf{x}}.$$

Is this positive semidefinite? Consider $\mathbf{z} \in \mathbb{R}^n$. Then

$$\mathbf{z}^\top \nabla^2 f(x)\mathbf{z} = \mathbf{z}^\top 4\mathbf{x}\mathbf{x}^\top e^{\mathbf{x}^\top \mathbf{x}} \mathbf{z} = 4(\mathbf{z}^\top \mathbf{x})^2 e^{\mathbf{x}^\top \mathbf{x}} \geq 0$$

because $\exp(\cdot)$ and $(\mathbf{z}^\top \mathbf{x})^2$ are both non-negative. Thus $\nabla^2 f \succeq 0$ and by the second order conditions f is convex. ■

3.4 Operations Preserving Convexity

In this section we examine some handy tools for constructing convex functions from other convex functions. There are *many* more properties than those shown here. See *Convex Optimization* by Boyd and Vandenberghe for many others.

Theorem 3.5 (Non-Negative Weighted Sum)**.** *The function $F : C \mapsto \mathbb{R}$ where C is a convex set, $f_i : C \mapsto \mathbb{R}$ is a convex function, $w_i \geq 0$ and*

$$F(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$$

is convex.

Proof. Let $\mathbf{x}, \mathbf{y} \in C$ and $0 \leq \theta \leq 1$. Then

$$\begin{aligned} F(\theta\mathbf{x} + (1-\theta)\mathbf{y}) &= \sum_i w_i f_i(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \\ &\leq \sum_i w_i \left[\theta f_i(\mathbf{x}) + (1-\theta) f_i(\mathbf{y})\right] \\ &= \theta F(\mathbf{x}) + (1-\theta) F(\mathbf{y}), \end{aligned}$$

as desired. ■

Theorem 3.6 (Pointwise Maximum)**.** *Given convex functions $f_i : C \mapsto \mathbb{R}$, and $F : C \mapsto \mathbb{R}$ defined as*

$$F(\mathbf{x}) = \max_i f_i(\mathbf{x}),$$

F is convex.

Proof. By the definition of convexity, given $\mathbf{x}, \mathbf{y} \in C$ and $0 \leq \theta \leq 1$ we have

$$\begin{aligned} F(\theta\mathbf{x} + (1-\theta)\mathbf{y}) &= \max_i f_i(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \\ &\leq \max_i \{\theta f_i(\mathbf{x}) + (1-\theta) f_i(\mathbf{y})\} && \text{(because } f_i \text{ is convex)} \\ &\leq \theta \max_i f_i(\mathbf{x}) + (1-\theta) \max_i f_i(\mathbf{y}) \\ &= \theta F(\mathbf{x}) + (1-\theta) F(\mathbf{y}), \end{aligned}$$

as desired. ■

4 Optimization Problems

An optimization problem is a problem of the form

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

where $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$ is called the **objective function**, $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ is an **inequality constraint**, and $h_i : \mathbb{R}^n \mapsto \mathbb{R}$ is an **equality constraint**. The goal of an optimization problem, as you might be able to guess, is to find $\mathbf{x}$ in the problem domain domain

$$\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \;\cap\; \bigcap_{i=1}^{p} \mathbf{dom}\, h_i$$

that minimizes $f_0(\mathbf{x})$ subject to the given conditions. The problem is said to be *feasible* if such an $\mathbf{x}$ exists, and *infeasible* otherwise.

The optimal value $p^\star$ of the problem above is defined to be

$$p^\star = \inf\{f_0(\mathbf{x}) : f_i(\mathbf{x}) \leq 0,\ i = 1, \dots, m,\ h_i(\mathbf{x}) = 0,\ i = 1, \dots, p\}.$$

If there are feasible points $\mathbf{x}_k$ with $f_0(\mathbf{x}_k) \to -\infty$ as $k \to \infty$, then we say the problem is *unbounded below.*

Note that we could have a similarly defined *maximization* problem. Everything is the same except for the definition of *unbounded below*, obviously.

4.1 Optimal Points

We say $\mathbf{x}^\star$ is an *optimal point* (or that it solves the above problem) if $\mathbf{x}^\star$ is feasible and $f_0(\mathbf{x}^\star) = p^\star$. If an optimal point exists, we say the problem is *solvable.*

We say a feasible point $\mathbf{x}$ is *locally optimal* if there exists some $R > 0$ such that $\mathbf{x}$ solves

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{z}) \\ \text{subj. to: } & f_i(\mathbf{z}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{z}) = 0, \quad i = 1, \dots, p \\ & ||\mathbf{z} - \mathbf{x}||_2 \leq R \end{aligned}$$

for variable $\mathbf{z}$. This intuitively means that $\mathbf{x}$ minimizes f_0 over nearby points in the feasible set.

4.2 Equivalent Problems

Given an optimization problem of the form

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

we may want to express our problem in a cleaner or easier to solve form. Under certain conditions changing our problem's form will not actually change the solution. Note that, as usual, there are a few more possible transformations that are kosher. Check out Boyd and Vandenberghe for these, but the ones shown here are likely the only ones you will need.

4.2.1 Scaling

The problem

$$\begin{aligned} \text{minimize: } & \alpha_0 f_0(\mathbf{x}) \\ \text{subj. to: } & \alpha_i f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & \beta_i h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

for $\alpha > 0$ and $\beta \neq 0$ is equivalent to the original problem. This should be intuitive since, for example x^2 is minimized at 0, changing the scale of your axes maintains that minimum. Similarly, if equality holds (ie. $4x + 2 = 0$) multiplying by any non-zero number on both sides maintains that equality.

4.2.2 Change of Variables

Given an one-to-one mapping $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$, where the image (range) of ϕ covers the domain of your problem $\mathcal{D}$, the problem

$$\begin{aligned} \text{minimize: } & f_0(\phi(\mathbf{x})) \\ \text{subj. to: } & f_i(\phi(\mathbf{x})) \leq 0, \quad i = 1, \dots, m \\ & h_i(\phi(\mathbf{x})) = 0, \quad i = 1, \dots, p \end{aligned}$$

is equivalent to the original. This should be clear. If x solves the original problem, then $z = \phi^{-1}(x)$ solves the transformed problem. The converse also holds.

4.2.3 Transformation of Objective and Constrain Functions

Suppose $\psi_0 : \mathbb{R} \mapsto \mathbb{R}$ is monotonically increasing, $\phi_1, \dots, \psi_m : \mathbb{R} \mapsto \mathbb{R}$ satisfy $\phi_i(u) \leq 0$ if and only if $u \leq 0$, and $\psi_{m+1}, \dots, \phi_{m+p} : \mathbb{R} \mapsto \mathbb{R}$ satisfy $\phi_i(u) = 0$ if and only if $u = 0$. The problem

$$\begin{aligned} \text{minimize: } & \psi_0(f_0(\mathbf{x})) \\ \text{subj. to: } & \psi_i(f_i(\mathbf{x})) \leq 0, \quad i = 1, \dots, m \\ & \psi_{m+i}(h_i(\mathbf{x})) = 0, \quad i = 1, \dots, p \end{aligned}$$

is equivalent to the original. This should be evident from the conditions we placed on each ψ_j.

4.2.4 Slack Variables

This will come in very handy. A prudent observation is that $f_i(x) \leq 0$ if and only if for some $s_i \geq 0$, $f_i(x) + s_i = 0$. Thus the problem

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) + s_i = 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \\ & s_i \geq 0, \quad i = 1, \dots, m \end{aligned}$$

is equivalent to the original.

4.2.5 Epigraph Form

The *epigraph form* of the original problem is

$$\begin{aligned} \text{minimize: } & t \\ \text{subj. to: } & f_0(\mathbf{x}) - t \leq 0, \\ & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

where $t \in \mathbb{R}$. It should be clear that this is the same as the original problem as the first constraint can be viewed as $f_0(\mathbf{x}) \leq t$. Thus the objective must always lie under t and minimizing t will minimize the highest possible value of f_0.

4.3 Convex Optimization Problems

We will now study the branch of optimization problems we will see *extremely* often in our studies. We say a problem of the form

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & A\mathbf{x} = \mathbf{b} \end{aligned}$$

is a **convex optimization problem** if the objective function and inequalities f_i are convex, and the equality constraints h_i are linear. Convex problems are, as a whole, extremely *nice* in the sense that we have efficient (read 'polynomial time') algorithms for finding global optima.

At heart, a convex optimization problem is just a minimization of a convex function within a convex domain.

4.3.1 Global Optimality of a Convex Optimum

The reason that convex optimization problems are so nice to solve is that *any* optimum is a global optimum. In other words, if you find some solution you find *the* solution. This is by no means the case in non-convex problems such as neural networks, although you might use similar methods to find *sufficiently good solutions.*

Theorem 4.1 (Global Optimality)**.** *Given a convex optimization problem, and local optimum* $\mathbf{x}^\star$ *such that*

$$f_0(\mathbf{x}) = \inf\{f_0(\mathbf{z}) : \mathbf{z} \textit{ feasible and } ||\mathbf{z} - \mathbf{x}||_2 \leq R\}$$

for some $R > 0$*.* $\mathbf{x}^\star$ *is the global optimum.*

Proof. (Boyd) Suppose such a problem and local optimum. Now suppose, to the contrary, that $\mathbf{x}$ is *not* the global optimum, and therefore there exists some feasible $\mathbf{y}$ such that $f_0(\mathbf{y}) < f_0(\mathbf{x})$. Then $||\mathbf{y} - \mathbf{x}||_2 > R$ because otherwise $f_0(\mathbf{x}) \leq f_0(\mathbf{y})$. Consider a point $\mathbf{z}$ given by

$$\mathbf{z} = (1-\theta)\mathbf{x} + \theta\mathbf{y} \quad \text{and} \quad \theta = \frac{R}{2||\mathbf{y} - \mathbf{x}||_2}.$$

Then $||\mathbf{x} - \mathbf{z}||_2 = R/2 < R$. By convexity of the feasible set, $\mathbf{z}$ is feasible. By the convexity of the objective function f_0 we also have

$$f_0(\mathbf{z}) \leq (1-\theta)f_0(\mathbf{x}) + \theta f_0(\mathbf{y}) < f_0(\mathbf{x}),$$

contradicting our assumption of local optimality. Thus $\mathbf{x}$ must be the global optimum ($R \to \infty$). ∎

4.3.2 Linear Programs

While we won't study Linear Programs much in the course, they are necessary to understand and are a central tool in operations research and graph algorithms. A linear program is a convex optimization problem of the form

$$\begin{aligned} \text{minimize: } & \mathbf{c}^\top \mathbf{x} \\ \text{subj. to: } & G\mathbf{x} \leq \mathbf{h}, \\ & A\mathbf{x} = \mathbf{b} \end{aligned}$$

where $G \in \mathbb{R}^{m\times n}$ and $A \in \mathbb{R}^{p\times n}$. Note that because the objective function is linear this could also have been a maximization problem with no change to convexity. For a ton of interesting applications, see Boyd and Vandenberghe, or consider taking the Operations Research Course!

4.3.3 Quadratic Programs

Interestingly enough, of all standard convex optimization problem classes, quadratic programs seem to come up the most in machine learning. Basically, pay attention!

A quadratic program is an optimization program of the form

$$\begin{aligned} \text{minimize: } & \frac{1}{2}\mathbf{x}^\top P\mathbf{x} + \mathbf{q}^\top\mathbf{x} + \mathbf{r} \\ \text{subj. to: } & G\mathbf{x} \preceq \mathbf{h} \\ & A\mathbf{x} = \mathbf{b} \end{aligned}$$

where $P \in \mathbb{S}^n_+$, $G \in \mathbb{R}^{m\times n}$, and $A \in \mathbb{R}^{p\times n}$. In this class of program, we are minimizing a quadratic function inside of a polyhedron.

4.4 Examples of Convex Optimization Problems

Here we present two examples of applied convex optimization problems. See *Convex Optimization* by Boyd & Vandenberghe for many, many more.

4.4.1 Analytic Centering

Consider the problem of finding the 'center' of some polygon $\{\mathbf{x} : A\mathbf{x} \preceq \mathbf{b}\}$. We first need to denote what we consider the optimal center. The *Chebyshev Center* of a set of points $\{\mathbf{x}_i\}$ is the center of the smallest circle that will fit around all of the points. This is easily expressed as the following optimization problem:

$$\begin{aligned} \text{minimize: } & r \\ \text{subj. to: } & ||\mathbf{x}_i - \mathbf{c}||_r \le r \end{aligned}$$

where r and $\mathbf{c}$ are variables. This is obviously convex. We could set $\mathbf{x}_i$ as the vertices of the polygon to then find a center. But that has some issues, for example, when your polygon is a very long, thin rectangle with a triangle slapped onto one side of it. The Chebyshev Center won't include be affected by the triangle at all until it hits the end of the circle. To remedy this issue we will consider a more effective method of finding the center below.

What if we wanted to create a *unconstrained* convex optimization problem that would given us some definition of a 'center'? First note that $A\mathbf{x} \preceq \mathbf{b}$ implies that $a_i^\top \mathbf{x} \le \mathbf{b}_i$ where a_i is the $i-$th row of A. This condition is satisfied, obviously, when $-\log(\mathbf{b}_i - a_i^\top \mathbf{x})$ exists. But more importantly, as $a_i^\top \mathbf{x}$ approaches $\mathbf{b}_i$, we note that the $-\log$ term approaches infinity. This is called a 'log barrier'. Now if we combine a bunch of log barriers, say, perhaps, along each of the edges of our polygon and "walking down hill" until we find the point that minimizes all of the log barriers. This sounds, intuitively like a good idea of a 'center', and is called the *Analytic Center* of a polygon. This is also easily expressed as a convex optimization problem, this time without constraints:

$$\text{minimize: } f_0(\mathbf{x}) = -\sum_i \log(\mathbf{b}_i - a_i^\top \mathbf{x})$$

Note that the gradient

$$\nabla f_0 = \sum_i \frac{\mathbf{a}_i}{\mathbf{b}_i - \mathbf{a}_i^\top \mathbf{x}},$$

so if you are 'far' from a barrier it doesn't effect your gradient. This can be fed into any standard gradient based optimizer, as seen in Figure 4

Now let's show that the analytic centering problem is convex. First, note that log is a concave function (check this for yourself), so $-\log$ is convex. Second, as affine composition $f(A\mathbf{x} + \mathbf{b})$ preserves convexity, $-\log(b_i - a_i^\top \mathbf{x}_i)$ is also convex. Finally, as a non-negative weighted sum of convex functions, the final objective f_0 is therefore convex.

4.4.2 Logistic Regression

Note that our notation will assume $\mathbf{x}_0 = 1$ as a bias term.

Now let's reconsider logistic regression as an optimization problem. Suppose we have a bunch of points $\mathbf{x}_i$, with labels $\mathbf{y}_i \in \{0, 1\}$. We want to classify some new point $\mathbf{x}$. Now suppose we generate a distribution over the labels of a given point $\mathbf{x}$ with

$$\mathbb{P}(\mathbf{y} = 1 | \mathbf{x}, \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^\top \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \mathbf{x}}} \in (0, 1).$$

Intuitively, this takes regular linear regression over the binary labels $\{0, 1\}$ and squashes it into a function $f : \mathbb{R} \mapsto [0, 1]$ to give us the ability to probabilistically interpret the output of our classifier. In a frequentist setting we want to maximize the *likelihood* of our data, that is, the probability of our labelled data given our parameters (just $\boldsymbol{\theta}$ in this case). However, because the actual likelihood will involve products we can

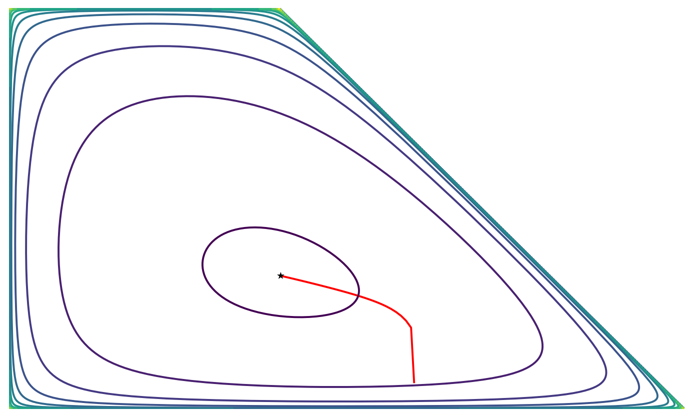


Figure 4: **Analytic Centering**. Here we compute the analytic center of a polygon in $\mathbb{R}^2$. Contour lines represent the objective surface while the red path represents the progression of gradient descent. The star is the optimal center. See `https://math189r.github.io/materials/section/fig_code/analytic_centering.py` for code to generate this plot.

transform into log space (because $\log(x)$ is a monotonically increasing function!!!) and equivalently maximize the log likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \log \mathbb{P}(\mathbf{y} = \mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\theta}) = \sum_{i=1}^{m} \log \begin{cases} \sigma(\boldsymbol{\theta}^\top \mathbf{x}) & \text{if } \mathbf{y}_i = 1 \\ 1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x}) & \text{if } \mathbf{y}_i = 0 \end{cases}$$

Note that due to the binary labels we can represent this compactly as

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \mathbf{y}_i \log(\sigma(\boldsymbol{\theta}^\top \mathbf{x})) + (1 - \mathbf{y}_i) \log(1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x})).$$

Now our (unconstrained) optimization problem becomes simply

$$\text{maximize: } \mathcal{L}(\boldsymbol{\theta})$$

where we could add a $\lambda \boldsymbol{\theta}^\top \boldsymbol{\theta}$ regularization term if we place a Gaussian prior on our parameter vector $\boldsymbol{\theta}$. Usually this would be a good idea.

Now to optimize this, as we will/have seen, first note that the derivative of the sigmoid function $\sigma'(x) = \sigma(x)\,(1 - \sigma(x))$. Then the gradient of our log output $\log \mathbb{P}(\mathbf{y} = 1 | \mathbf{x}, \boldsymbol{\theta}) = \log \sigma(\boldsymbol{\theta}^\top \mathbf{x})$ becomes

$$\nabla_\theta \log \sigma(\boldsymbol{\theta}^\top \mathbf{x}) = \mathbf{x}(1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x})).$$

Similarly,

$$\nabla_\theta \log(1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x})) = -\mathbf{x}\sigma(\boldsymbol{\theta}^\top \mathbf{x}).$$

Prove to yourself why this is correct. Thus the derivative of our objective $\mathcal{L}(\boldsymbol{\theta})$ becomes

$$\nabla_\theta \mathcal{L} = \sum_{i=1}^{m} \Big[\mathbf{y}_i (1 - \sigma(\boldsymbol{\theta}^\top \mathbf{x})) - (1 - \mathbf{y}_i) \sigma(\boldsymbol{\theta}^\top \mathbf{x}) \Big] \mathbf{x}_i = \sum_{i=1}^{m} \Big[\mathbf{y}_i - \sigma(\boldsymbol{\theta}^\top \mathbf{x}) \Big] \mathbf{x}.$$

This can be input into any standard gradient-based optimization algorithm.

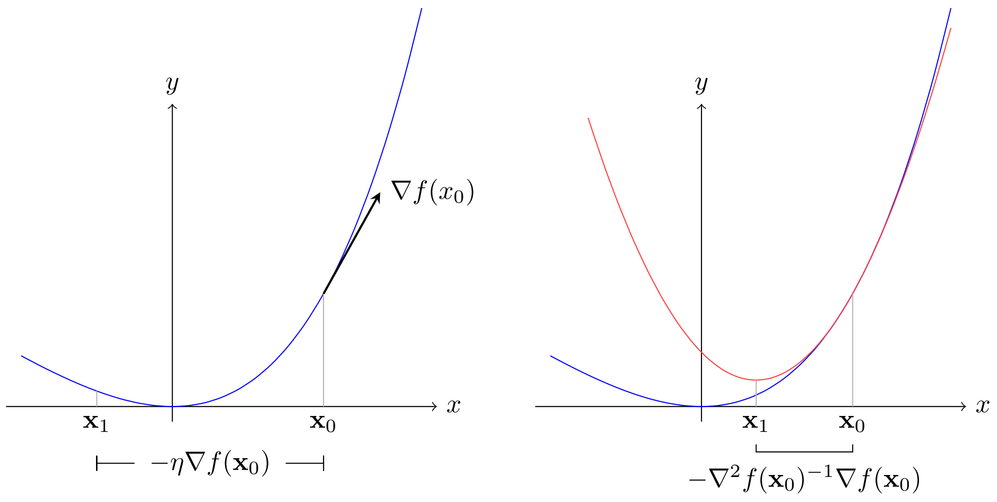


Figure 5: **Optimization Algorithms**. A vizualization of gradient descent and Newton's Method. Newton's Method, although you can't tell much in this example, is much faster. While Newton's Method analtyically solves a quadratic approximation of your (hopefully convex) problem at each iteration, gradient descent acts as an approximation of that, just taking small steps in the right direction to optimize your problem. If the learning rate η is too large, your iterates will start to diverge. If your learning rate is too small, your iterates will take too long to converge.

Now let's show that logistic regression is convex. This is equivalent to showing the objective function f_0 is concave, as we are maximizing. As affine composition $f(A\mathbf{x} + \mathbf{b})$ and non-negative weighted summation preserve concavity, one can see that this is the same as showing that

$$f(x) = \log \sigma(x)$$

is concave on $\mathbb{R}$. As $\sigma'(x) = \sigma(x)(1 - \sigma(x))$, we can see that

$$f'(x) = \frac{\sigma(x)(1 - \sigma(x))}{\sigma(x)} = 1 - \sigma(x)$$

and

$$f''(x) = -\sigma(x)(1 - \sigma(x)),$$

which always negative, because $\sigma(x) \in [0, 1]$. Thus the objective and therefore the whole problem of logistic regression is convex, as desired.

5 Algorithms for Solving Convex Problems

As has been the theme thus far, if you want to see *much* more information about algorimths for solving convex optimization problems, see *Convex Optimization* by Boyd & Vandenberghe. Here we will just revisit good old gradient descent and Newton's Method for solving unconstrained convex optimization problems. Boyd goes into detail on solving constrained optimization problems efficiently, but we likely won't need to do much of that (hopefully!).

5.1 Gradient Descent

Suppose, given $f : \mathbb{R}^n \mapsto \mathbb{R}$, we want to find when $f'(\mathbf{x}) = 0$. Note that while f need not be convex, if f is convex we are guarenteed to find the global optimum when we find a critical point. Now picture our function as a surface mapping $\mathbb{R}^2$ to $\mathbb{R}$. Intuitively, if we want to minimize f, we just need to walk in the direction of ‘down hill’. Assuming f is once-differentiable, we can use the gradient ∇f to define this direction. So, given a starting point, we need to take steps in the direction of ∇f of some size, called the learning rate η. We do this until our gradient ∇f is smaller than some ϵ.

```
Gradient Descent
input  : $f$, $\nabla f$, $\eta(t)$, starting point $\mathbf{x}_0$, tolerance $\epsilon$
output: optimal point $\mathbf{x}^\star$
$t \leftarrow 0$
while $||\nabla f||_2 \geq \epsilon$ do
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(t)\nabla f(\mathbf{x}_t)$
    $t \leftarrow t + 1$
end
return  $\mathbf{x}^\star = \mathbf{x}_{t+1}$
```

Oftentimes you can use a contant $\eta(t) = 0.01$ or some similarly small number. More sophisticated learning rates and augmentation terms, like adding a **momentum term** giving an update

$$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta(t)\nabla f(\mathbf{x}_t) + \mu_t(\mathbf{x}_t - \mathbf{x}_{t-1})$$

where μ_t might be a constant around 0.9. You can get a sense of how heuristic these methods are! Some problems with gradient descent are that

- it is slow in convergence compared to second order methods
- it provides a number of hyperparameters to set, which can end up using a lot of heuristic based approaches to improvement
- if your problem isn’t “round” (think $\mathbf{x}^\top \mathbf{x}$ where the axes are scaled the same) you will end up zig-zagging along a basin. Newton’s method remedies this by being scale invariant

5.1.1 Minibatch and Stochastic Gradient Descent

Often times in machine learning specifically we will have huge datasets and objective functions of the form

$$f_0(\theta) = \frac{1}{m}\sum_{i=1}^{m} f(\mathbf{x}_i|\theta)$$

where $\mathbf{x}_i$ are our training examples and $f(\mathbf{x}|\theta)$ is our loss function. In this case, m is so large (ie. $m > 10^5$ or $m > 10^6$) that attempting to compute

$$\nabla f_0 = \frac{1}{m}\sum_{i=1}^{m} \nabla f(\mathbf{x}_i|\theta)$$

at every iteration of training using gradient descent would be too slow. Intuitively, if you use (batch, like above) gradient descent you have to go through your entire training set each iteration. If you have a million points, however, the last point $i = m$ will hardly give you any information on average about the surface you are on. Thus, a natural approximation is to select a random subset of size k at each iteration and use those k points to compute an approximated gradient

$$\nabla f_0 \approx \frac{1}{m}\sum_{i=1}^{k} \nabla f(\mathbf{x}_i|\theta)$$

and use that to compute the update.
With different values of k this algorithm is called different things. The algorithm is theoretically guarenteed to converge for any k under certain conditions on hyperparameters.

```
**Stochastic Gradient Descent**
input  : $f$, $\nabla f$, $\eta(t)$, starting point $\boldsymbol{\theta}_0$, dataset $\mathcal{D}$, tolerance $\epsilon$, batch size $k$
output: optimal point $\boldsymbol{\theta}^\star$
$t \leftarrow 0$
while $||\nabla f||_2 \geq \epsilon$ do
    select training points $X = \{\mathbf{x}_i\}_{i=1}^k \subseteq \mathcal{D}$ at random
    $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta(t)\nabla f(X|\boldsymbol{\theta}_t)$
    $t \leftarrow t+1$
end
return  $\boldsymbol{\theta}^\star = \boldsymbol{\theta}_{t+1}$
```

- $k = 1$ (Stochastic Gradient Descent): You might think this wouldn't do too well because you lose so much information by only looking at a single point but it is actually used heavily in distributed model learning due to its algorithmic simplicity.
- $1 < k < m$ (Minibatch Gradient Descent): This is the de-facto standard for solving large problems (well, with small variants to the update). Usually in Deep Learning people set $15 < k < 256$ with $k = 128$ being pretty standard.
- $k = m$ This is just regular (batch) gradient descent as seen above.

5.2 Newton's Method

Here we introduce a method that is *super fast* but probably won't scale to problems with more than a couple dozen thousand examples depending on your problem structure. Effectively Newton's Method does the following logic:

1. We want to minimize a twice-differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$. We have an initial guess for the minimizer $\mathbf{x}_0$.
2. We know how to minimize quadratic systems $\mathbf{x}^\top A\mathbf{x} + \mathbf{b}^\top \mathbf{x} + \mathbf{c}$ in closed form. Generally speaking this is the best approximator to our function that we can optimize in closed form.
3. Why don't we take your function and approximate it as a quadratic Taylor polynomial about your guess and set your guess to be the optimum of that approximation?

See Figure 5 for a visualization of a Newton step. Note that, about some $\mathbf{x}_0$, the quadratic Taylor approximation

$$\widetilde{f}(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^\top(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^\top \nabla^2 f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

with gradient vanishing when

$$\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}_0) + \nabla^2 f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) = 0,$$

or

$$\mathbf{x} = \mathbf{x}_0 - \nabla^2 f(\mathbf{x}_0)^{-1}\nabla f(\mathbf{x}_0).$$

Note that this looks strikingly similar to gradient descent, except instead of having a scalar learning rate we have the inverse of a linear map $\nabla^2 f(\mathbf{x}_0)^{-1}$. Also, solving that system

$$\nabla^2 f(\mathbf{x}_0)\mathbf{d} = \nabla f(\mathbf{x}_0)$$

is what calls for trouble when trying to optimize at very large scale. Many approximation algorithms exist (see L-BFGS for an example) to work in first order methods yet approximate the second order terms during optimization. For many problems there exists a helpful structure in the Hessian (eg. tridiagonal or arrow structure) you can exploit to solve that system relatively quickly. In general using a backslash solver (`A\b`) is a terrible idea for using Newton's Method to optimize at scale because of this, but it is much preferred to

```
Newton's Method
input  : $f$, $\nabla f$, $\nabla^2 f$, starting point $\mathbf{x}_0$, tolerance $\epsilon$
output: optimal point $\mathbf{x}^\star$
$t \leftarrow 0$
while $||\nabla f||_2 \geq \epsilon$ do
    solve $\nabla^2 f(\mathbf{x}_t)\mathbf{d}_t = \nabla f(\mathbf{x}_t)$ for $\mathbf{d}_t$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \mathbf{d}_t$
    $t \leftarrow t + 1$
end
return  $\mathbf{x}^\star = \mathbf{x}_{t+1}$
```

using `pinv(A)*b` or `inv(A)*b`. If f is just convex, for example you can use a Cholesky decomposition to cut the backslash time in half.
The reason Newton's Method is fast is because it obeys what is called **quadratic convergence**. Roughly speaking, once your iterates $\mathbf{x}_t$ get close enough to the optimum, every iteration will double the number of digits you are precise to. This means that once you have a pretty accurate answer, running for one or two more iterations will max out double floating point precision on any computer. Even when you aren't in the quadratic region, Newton's Method's number of iterations to become ϵ suboptimal is still bounded above by

$$\frac{f(\mathbf{x}_0) - p^\star}{\gamma} + \log_2 \log_2(1/\epsilon) \qquad (\log_2 \log_2(1/\epsilon) \approx 6.)$$

where γ is a constant depending on the problem. Usually $1/\gamma$ is about 375, depending on the problem, but Newton's Method empirically converges much faster than even that bound. See Boys & Vandenberghe for a proof of this bound.

6 Duality

The idea of "duality", forming a complementary problem that maintains all of the information of your problem, is found all over mathematics. In optimization as a whole, and convex optimization specifically, optimization is central. First we will investigate an intuitive representation of any constrained convex optimization problem which will then lead us directly to Lagrange Duality before investigating the implications of this.

6.1 Constructing The Dual

Given an optimization problem (which we will call the **primal problem**)

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned}$$

note that minimizing f_0 under the constraints is the same as optimizing f_0 over all of $\mathbb{R}^n$ while letting

$$f_0(\mathbf{x}) = \begin{cases} f_0(\mathbf{x}) & \text{if constraints are satisfied} \\ \infty & \text{otherwise.} \end{cases}$$

This makes sense, because if there is any feasible point, there will exist $f_0(\mathbf{x}_{feasible}) < \infty$ if there is any practical solution to your problem. Otherwise, if $f_0(\mathbf{x}_{feasible}) = \infty$, you have still solved your problem! The question now is how to change our constrained optimization problem into an unconstrained one by embedding the constraints in the objective. One simple way might be to use an indicator function

$$I_-(u) = \begin{cases} 0 & \text{if } u \leq 0 \\ \infty & \text{if } u > 0 \end{cases}$$

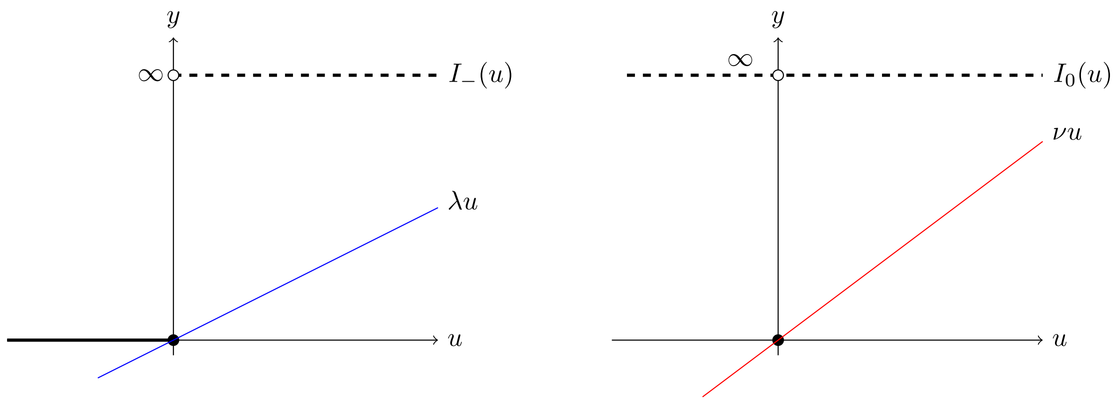


Figure 6: **Constrain Indicator Approximation:** So long as $\lambda > 0$ with $\nu \lesseqgtr 0$, $\lambda u \leq I_-(u)$, and $\nu u \leq I_0(u)$. This lets us construct the dual in a more tractable manner.

to wrap each inequality constraint, so that if at any $\mathbf{x}$ any $f_i(\mathbf{x}) > 0$, the objective will rise to infinity. Convince yourself why this would work. Similarly, for equality constraints, if we use another indicator function

$$I_0(u) = \begin{cases} 0 & \text{if } u = 0 \\ \infty & \text{otherwise} \end{cases}$$

to wrap each equality, we can assure that our objective diverges to infinity if at any $\mathbf{x}$ we have $h_i(\mathbf{x}) \neq 0$. Our problem then becomes

$$\text{minimize: } f_0(\mathbf{x}) + \sum_{i=1}^{m} I_-(f_i(\mathbf{x})) + \sum_{i=1}^{p} I_0(h_i(\mathbf{x})).$$

Convince yourself why this is the same. You can intuitively think of the indicator functions as expressing a "hard displeasure" to the objective whenever constraints are not satisfied. To solve anything tractably a natural step might be to approximate this "hard displeasure function" with a "soft", linear displeasure function

$$I_-(u) \approx \lambda u \text{ for } \lambda \geq 0 \qquad I_0(u) \approx \nu u \text{ for unconstrained } \nu.$$

You can immediately see that these linear approximations are not the same, but are always underestimates of the hard indicators. See Figure 6. Representing our problem in this new way we have an objective

$$\begin{aligned} &\text{minimize: } L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \\ &\text{subj. to: } \boldsymbol{\lambda} \succeq 0. \end{aligned}$$

We call $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ the **Lagrangian** of our optimization problem. As our linear approximation of the indicator functions are always below these functions, we might see that our primal problem is *exactly*

$$p^\star = \inf_{\mathbf{x}} \sup_{\nu;\lambda \geq 0} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$$

The **dual** of our problem is just the flip of order with which we are optimizing, namely the dual

$$d^\star = \sup_{\nu;\lambda \geq 0} \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}).$$

6.2 Duality Gap: $p^\star - d^\star$

As we discussed, the linear approximation of our constraints is always an under approximation, so if we don't maximize that first our problem is always an under approximation of our primal problem. Thus we have the following theorem:

Theorem 6.1 (Weak Duality)**.** *For any optimization problem (even a non-convex one),*

$$d^\star \leq p^\star.$$

Proof. See discussion above. ■

From this we can create a handy definition.

Definition 6.1 (Duality Gap)**.** *We define the* ***duality gap*** *to be*

$$p^\star - d^\star \geq 0.$$

As the duality gap tends towards 0 it means our dual better and better approximates our primal problem. In some (any many we will see) cases, $p^\star = d^\star$. In this case we say **strong duality** holds, and solving the dual is equivalent to solving the primal problem. In general this is not the case, but if the problem is convex, ie. of the form

$$\begin{aligned} \text{minimize: } & f_0(\mathbf{x}) \\ \text{subj. to: } & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & A\mathbf{x} = \mathbf{b} \end{aligned}$$

with f_i convex, we usually (but not always) will have strong duality. One sufficient but not necessary condition for strong duality is **Slater's Condition**.

Theorem 6.2 (Slater's Condition)**.** *If the primal problem is convex as above, and there exists some* $\mathbf{x} \in \operatorname{relint} \mathcal{D}$ *such that*

$$f_i(\mathbf{x}) < 0 \ \ \forall i \qquad \textit{and} \qquad A\mathbf{x} = \mathbf{b},$$

then

$$d^\star = p^\star.$$

When some of the f_i $i = 1, \dots, k$ *are affine, we can refine Slater's Condition to allow*

$$f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, k \qquad f_i(\mathbf{x}) < 0, \quad i = k+1, \dots, m \qquad A\mathbf{x} = \mathbf{b}$$

to ensure the same strong duality.

6.3 The Lagrange Dual Function: $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$

6.4 Finding the Dual

6.4.1 The Dual of The Linear Program

6.5 Solving Problems Using Duality

6.5.1 Minimize a Quadratic Under Quadratic Constraints

6.6 Complementary Slackness of Duality

6.6.1 Solving The Dual Linear Program via a Primal Solution

6.7 The KKT (Karush-Kuhn-Tucker) Conditions