

Subgradients and Projected Subgradient Descent

Michael J. Neely
University of Southern California

An updated version of these notes is here:

<https://ee.usc.edu/stochastic-nets/docs/subgradients.pdf>

I. INTRODUCTION

These notes define the concept of a subgradient $f'(x)$ of a convex function $f : X \rightarrow \mathbb{R}$ at a point $x \in X$. Subgradients are useful for treating general (possibly nondifferentiable) convex functions. Convex functions that are not differentiable are also called “nonsmooth” functions. These notes show that if f is a convex function, then f has at least one subgradient $f'(x)$ at every point x in the interior of $\text{dom}(f)$. This is proven using the hyperplane separation theorem. *Therefore, subgradients of a convex function can exist even at points in the domain where gradients do not exist.* However, there is a direct connection between subgradients and gradients: If a convex function f is differentiable at a point x in the interior of its domain (so the gradient $\nabla f(x)$ exists), then $\nabla f(x)$ is the one and only subgradient of f at x .

This is useful for minimizing a convex and possibly nonsmooth function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ over a closed and convex set $C \subseteq \mathbb{R}^n$. A simple algorithm, called *projected subgradient descent*, initializes any vector $x_0 \in C$ and computes iterates $\{x_k\}_{k=0}^\infty$ via

$$x_{k+1} = \pi_C(x_k - \delta_k f'(x_k)) \quad \forall k \in \{0, 1, 2, \dots\}$$

where $\delta_k > 0$ is some *stepsize* for step k , and $\pi_C(w)$ denotes the unique projection of the point $w \in \mathbb{R}^n$ onto the closed and convex set C . This method utilizes a *subgradient oracle* that provides a particular subgradient $f'(x) \in \mathbb{R}^n$ when queried about a particular $x \in C$. Convergence of this method can be analyzed in three regimes: (i) General (possibly nonsmooth) convex functions f ; (ii) Smooth (differentiable) convex functions f ; (iii) Strongly convex (and smooth) functions f . For general (possibly nonsmooth) convex functions, these notes analyze the case of fixed stepsize $\delta_k = \delta$, vanishing stepsize $\delta_k = 1/\sqrt{k+1}$, and a modified algorithm called *mirror descent*. For smooth and strongly convex cases, these notes consider fixed stepsize and a simple form of adaptive stepsize. See [1] for extended results in all three cases, and [2] for backtracking algorithms in the strongly convex case.

The analysis of the general (nonsmooth) case allows a near-identical analysis for a remarkable situation of *online convex optimization* where functions are time-varying [3]. These notes show the algorithm provides desirable performance without knowing the future. This is useful, for example, for dynamic portfolio optimization with time-varying stock prices.

II. PRELIMINARIES

This section reminds the reader of the definition of a convex set and convex function. It also states the hyperplane separation theorem for the case of separating a point from a convex set. When $\mathbb{R}^n$ is used, it is implicitly assumed that n is a given positive integer. Given a nonempty set $X \subseteq \mathbb{R}^n$, a function $f : X \rightarrow \mathbb{R}$ is a mapping that assigns each $x \in X$ to a value $f(x) \in \mathbb{R}$. The set X is called the *domain* of f and is also written $\text{dom}(f)$. That is, if $f : X \rightarrow \mathbb{R}$ then $X = \text{dom}(f)$.

A. Open and closed balls

Given $x \in \mathbb{R}^n$ and $\epsilon > 0$, we define the open and closed balls of radius ϵ about x by

$$\begin{aligned} B_\epsilon(x) &= \{y \in \mathbb{R}^n : \|y - x\| < \epsilon\} \\ \overline{B}_\epsilon(x) &= \{y \in \mathbb{R}^n : \|y - x\| \leq \epsilon\} \end{aligned}$$

where $\|\cdot\|$ denotes the standard Euclidean norm (also called the L_2 norm):

$$\|v\| = \sqrt{\sum_{i=1}^n v_i^2} \quad \forall v \in \mathbb{R}^n$$

Given a set $X \subseteq \mathbb{R}^n$, we define the *complement* of X as

$$X^c = \{x \in \mathbb{R}^n : x \notin X\}$$

We say that x is an *interior point* of a set $X \subseteq \mathbb{R}^n$ if there is an $\epsilon > 0$ such that $B_\epsilon(x) \subseteq X$. Equivalently, x is an interior point of a set $X \subseteq \mathbb{R}^n$ if there is an $\epsilon > 0$ such that $\overline{B}_\epsilon(x) \subseteq X$. We say that x is a *boundary point* of X if for all $\epsilon > 0$, the set $B_\epsilon(x)$ contains points in both X and X^c . Every boundary point x of X satisfies either: (i) $x \in X$ and x is arbitrarily close to points in X^c ; (ii) $x \in X^c$ and x is arbitrarily close to points in X .

It can be shown that if $X \subseteq \mathbb{R}^n$, then every point in X is either an interior point of X or a boundary point of X (it cannot be both). A set $X \subseteq \mathbb{R}^n$ is *open* if it has no boundary points (so all points are interior points); A set $X \subseteq \mathbb{R}^n$ is *closed* if it contains all of its boundary points.¹ The set $\mathbb{R}^n$ is both open and closed.

B. Convexity definitions

Definition 1: A set $X \subseteq \mathbb{R}^n$ is a *convex set* if for all $x, y \in X$ and all $p \in [0, 1]$ we have

$$px + (1 - p)y \in X$$

Definition 2: Let $X \subseteq \mathbb{R}^n$ be a nonempty set. A function $f : X \rightarrow \mathbb{R}^n$ is a *convex function* if

- 1) X is a convex set
- 2) For all $x, y \in X$ and $p \in [0, 1]$ we have

$$f(px + (1 - p)y) \leq pf(x) + (1 - p)f(y)$$

C. Hyperplane separation

The next theorem states that if X is a nonempty convex subset of $\mathbb{R}^n$ and w is a point that is either not in X , or is a boundary point of X , then there is a hyperplane that passes through w and contains set X on one side. The theorem is foundational for convexity theory. Among other things, it leads to important results about: (i) subgradients of convex functions; (ii) Lagrange multipliers for constrained convex programs.

Theorem 1: (Hyperplane separation – point and set) Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. Let $w \in \mathbb{R}^n$ be a point such that either $w \notin X$, or w is a boundary point of X . Then

- a) There is a vector $a \in \mathbb{R}^n$, where $a \neq 0$, such that

$$a^\top x \leq a^\top w \quad \forall x \in X$$

- b) There is a vector $a \in \mathbb{R}^n$, where $a \neq 0$, such that

$$a^\top x \geq a^\top w \quad \forall x \in X$$

Proof: Part (b) of the theorem follows by multiplying part (a) of the theorem by -1 and defining $\tilde{a} = -a$. The proof of part (a) is omitted for brevity (see [2][4][5]). $\square$

D. Continuity

Theorem 2: Let $X \subseteq \mathbb{R}^n$ be a convex set and let $f : X \rightarrow \mathbb{R}$ be a convex function.

- a) f is continuous at each point x that is in the interior of X .
- b) f is continuous at each point x that is in the relative interior of X .

Proof: Omitted for brevity. $\square$

If $f : X \rightarrow \mathbb{R}$ is a convex function and X is an open set, the above theorem ensures f is continuous at all points of its domain. In particular, since $\mathbb{R}^n$ is an open set, every convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous at all $x \in \mathbb{R}^n$. A simple example of a discontinuous convex function is $f : [0, 1] \rightarrow \mathbb{R}$ defined by

$$f(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \in (0, 1] \end{cases} \quad (1)$$

It can be seen that this function is continuous at each point x in the interior of its domain, that is, each point $x \in (0, 1)$. It is also continuous at the right boundary point 1. However, it is not continuous at the left boundary point 0.

III. SUBGRADIENTS

Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set.

Definition 3: Let $f : X \rightarrow \mathbb{R}$ be a convex function. A vector $s \in \mathbb{R}^n$ is a *subgradient* of f at a given point $x \in X$ if

$$f(y) \geq f(x) + s^\top(y - x) \quad \forall y \in X \quad (2)$$

The set of all subgradients for f at a point $x \in X$, denoted $\partial f(x)$, is called the *subdifferential* of f at x . In particular, for each $x \in X$, the set $\partial f(x)$ exists (possibly being the empty set) and satisfies $\partial f(x) \subseteq \mathbb{R}^n$.

If $w, z \in X$ then $\partial f(w)$ and $\partial f(z)$ are two different (possibly empty) sets. If f has a subgradient s at a particular point $x \in X$, then $\partial f(x)$ is nonempty because $s \in \partial f(x)$. Given that $s \in \partial f(x)$, we can define the linear function $h(y) = f(x) + s^\top(y - x)$ for all $y \in X$, which is the function on the right-hand-side of (2). The function h can be viewed as a linear expansion of f about the point x . However, rather than *approximating* f , the function h acts as a *lower bound* on f . Indeed, the inequality (2) states that $f(y) \geq h(y)$ for all $y \in X$. Notice that the lower bound is exact when $y = x$, that is, $f(x) = h(x)$.

¹Equivalently, a set $X \subseteq \mathbb{R}^n$ is closed if and only if it contains all of its limit points.

A. Subgradient inequality

Suppose $X \subseteq \mathbb{R}^n$ is a convex set and $f : X \rightarrow \mathbb{R}$ is a convex function. A *particular* subgradient of f at a point $x \in X$ is typically denoted $f'(x)$. Thus, if $x \in X$ and $f'(x) \in \partial f(x)$ then (by definition of *subgradient*):

$$f(y) \geq f(x) + f'(x)^\top (y - x) \quad \forall y \in X \quad (3)$$

Inequality (3) is called the *subgradient inequality*. For a given $x \in X$, the set $\partial f(x)$ can have *many subgradients*, so a given subgradient $f'(x)$ is not necessarily unique within the set $\partial f(x)$. That is, $f'(x)$ represents just one particular element of $\partial f(x)$.

B. Minimizing a function f

Recall that a vector $x^* \in \mathbb{R}^n$ is a minimizer of a function $f : X \rightarrow \mathbb{R}$ if $x^* \in X$ and

$$f(x^*) \leq f(x) \quad \forall x \in X$$

The next lemma shows that a point $x^* \in X$ is a minimizer of f if and only if the all-zero vector is a subgradient of f at the point x^* .

Lemma 1: Suppose $X \subseteq \mathbb{R}^n$ is a nonempty convex set and $f : X \rightarrow \mathbb{R}$ is a convex function. Fix $x^* \in X$. Vector x^* is a minimizer of f if and only if $0 \in \partial f(x^*)$.

Proof: First suppose $0 \in \partial f(x^*)$. By definition (2),

$$f(y) \geq f(x^*) + 0^\top (y - x^*) \quad \forall y \in X$$

which means that $f(y) \geq f(x^*)$ for all $y \in X$, that is, x^* is a minimizer of f .

Conversely, suppose $x^* \in X$ is a minimizer of f . Then for all $y \in X$ we have

$$f(y) \geq f(x^*) = f(x^*) + 0^\top (y - x^*)$$

and so $0 \in \partial f(x^*)$. □

C. The structure of $\partial f(x)$

The following is a fundamental property of convex functions: *If f is a convex function, then f has at least one subgradient at each interior point of its domain.* This important result is proven by the hyperplane separation theorem. The following theorem restates this result and provides additional structural results for the set $\partial f(x)$. The proof of Theorem 3 is given in separate Lemmas in Section IV.

Theorem 3: Suppose $X \subseteq \mathbb{R}^n$ is a nonempty convex set and $f : X \rightarrow \mathbb{R}$ is a convex function. Then

- a) For each $x \in X$, the set $\partial f(x)$ is convex and closed (possibly empty).
- b) If x is an interior point of X then $\partial f(x)$ is nonempty and compact.
- c) If x is an interior point of X and $\nabla f(x)$ exists, then $\partial f(x) = \{\nabla f(x)\}$. That is, if the gradient of f exists at a particular interior point $x \in X$, then there is one and only one subgradient of f at x (and that subgradient is in fact the gradient).

Proof: See Section IV. □

If X is an open set then all points of X are interior points, so part (b) in the above theorem implies that every convex function $f : X \rightarrow \mathbb{R}$ has a subgradient at every point in the open set X . In particular, since $\mathbb{R}^n$ is an open set, every convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has at least one subgradient at every point $x \in \mathbb{R}^n$. This can be used as an alternative proof of Jensen's inequality for the special case of convex functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (see Exercise 2). The following provides a detail that is useful if the affine hull of X is not equal to $\mathbb{R}^n$ (so that f has no interior points).

Theorem 4: Suppose $X \subseteq \mathbb{R}^n$ is a nonempty convex set and $f : X \rightarrow \mathbb{R}$ is a convex function. If x is in the relative interior of X then $\partial f(x)$ is nonempty.

Theorem 4 can be used to prove some residual aspects of the general version of Jensen's inequality (see Exercise 3).

D. Examples

- Consider the nonsmooth convex function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined by the absolute value function:

$$f(x) = |x| \quad \forall x \in \mathbb{R}$$

This function is differentiable at every nonzero point in $\mathbb{R}$. It can be shown that

$$\partial f(x) = \begin{cases} \{1\} & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ \{-1\} & \text{if } x < 0 \end{cases}$$

For each $x \in \mathbb{R}$, set $\partial f(x)$ is indeed convex, nonempty, and compact. This *must* be true because each $x \in \mathbb{R}$ is an interior point of $\mathbb{R}$. The points where the derivative of f exists have $\partial f(x)$ equal to the one-element set given by that derivative.

- Consider the convex function $f : [0, 1] \rightarrow \mathbb{R}$ given by

$$f(x) = 5x \quad \forall x \in [0, 1]$$

This function is differentiable at each interior point $x \in (0, 1)$ and so it can be shown

$$\partial f(x) = \begin{cases} \{5\} & \text{if } x \in (0, 1) \\ (-\infty, 5] & \text{if } x = 0 \\ [5, \infty) & \text{if } x = 1 \end{cases}$$

In all cases, $\partial f(x)$ is convex and closed. The set $\partial f(x)$ has one element whenever $x \in (0, 1)$. The sets $\partial f(0)$ and $\partial f(1)$ are not bounded because 0 and 1 are boundary points of the domain (see related Exercise 30).

- The discontinuous convex function $f : [0, 1] \rightarrow \mathbb{R}$ given in (1) has subgradients at all points $x \in (0, 1]$, but does not have a subgradient at the boundary point $x = 0$. That is, $\partial f(0) = \emptyset$ (where $\emptyset$ denotes the empty set).
- An example continuous convex function that does not have a subgradient at a boundary point of its domain is $f : [0, \infty) \rightarrow \mathbb{R}$ given by

$$f(x) = -\sqrt{x} \quad \forall x \geq 0$$

Then

$$\partial f(x) = \begin{cases} \{-\frac{1}{2\sqrt{x}}\} & \text{if } x > 0 \\ \emptyset & \text{if } x = 0 \end{cases}$$

- Fix $a \in \mathbb{R}^n$ and consider the nonsmooth convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ given by

$$f(x) = \|x - a\| \quad \forall x \in \mathbb{R}^n$$

where $\|\cdot\|$ denotes the Euclidean norm. We want to compute a subgradient $f'(x)$ for each $x \in \mathbb{R}^n$. We can use

$$f'(x) = \begin{cases} \frac{x-a}{\|x-a\|} & \text{if } x \neq a \\ 0 & \text{if } x = a \end{cases}$$

This is valid because if $x \neq a$ then f is differentiable and $\nabla f(x) = (x-a)/\|x-a\|$, which is the one and only subgradient at point x . If $x = a$ then, since a is a minimizer of f , we know $0 \in \partial f(a)$. It turns out that 0 is not the *only* subgradient in $\partial f(a)$, but it is the easiest one to use.

- Let $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex functions for $i \in \{1, \dots, k\}$. Fix nonnegative weights $w_1, \dots, w_k$ and define $h : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$h(x) = \sum_{i=1}^k w_i f_i(x)$$

Given $x \in \mathbb{R}^n$ and $i \in \{1, \dots, k\}$, let $f'_i(x)$ be any subgradient in $\partial f_i(x)$. Then a valid subgradient $h'(x) \in \partial h(x)$ is (see Exercise 4):

$$h'(x) = \sum_{i=1}^n w_i f'_i(x)$$

- Consider the convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ given by

$$f(x) = \max\{g_1(x), g_2(x), \dots, g_k(x)\} \quad \forall x \in \mathbb{R}^n$$

where k is a positive integer and $g_1, \dots, g_k$ are assumed to be convex functions over all $x \in \mathbb{R}^n$. For each $x \in \mathbb{R}^n$ and each $i \in \{1, \dots, k\}$, let $g'_i(x)$ denote a particular subgradient of g_i at the point x . It can be shown that for each $x \in \mathbb{R}^n$, the following is a valid subgradient in $\partial f(x)$ (see Exercise 6):

$$f'(x) = g'_{j(x)}(x)$$

where $j(x)$ represents any particular index in the set $\arg \max_{i \in \{1, \dots, k\}} \{g_i(x)\}$. That is, to compute a subgradient of f at a point x , we first find any index $j \in \{1, \dots, k\}$ such that $g_j(x) \geq g_i(x)$ for all $i \in \{1, \dots, k\}$. Then we define $f'(x) = g'_j(x)$.

- Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. For each point $x \in \mathbb{R}^n$ let $f'(x)$ denote a subgradient of f at x . Define the function $h : \mathbb{R}^m \rightarrow \mathbb{R}$ by

$$h(x) = f(Ax + b) \quad \forall x \in \mathbb{R}^m$$

where A is any $n \times m$ matrix and b is any column vector of size n . For each $x \in \mathbb{R}^m$, a valid subgradient of h at x is

$$h'(x) = A^\top f'(Ax + b)$$

E. Subgradient calculus

Lemma 2: Let X and Y be nonempty convex subsets of $\mathbb{R}^n$ that satisfy $X \subseteq Y$. Let $f : Y \rightarrow \mathbb{R}$ be a convex function. Define the convex function $g : X \rightarrow \mathbb{R}$ as the restriction of f to the smaller domain X , so that

$$g(x) = f(x) \quad \forall x \in X$$

Then

$$\partial f(x) \subseteq \partial g(x) \quad \forall x \in X$$

Proof: The proof is straightforward and is left as an exercise (see Exercise 7). □

The above result is useful when $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function and $g : X \rightarrow \mathbb{R}$ is the restriction of f to a smaller convex domain $X \subseteq \mathbb{R}^n$. Since f has at least one subgradient $f'(x)$ for all $x \in \mathbb{R}^n$, we know g has at least one subgradient for all $x \in X$, namely, $g'(x) = f'(x)$. It turns out that $\partial g(x) = \partial f(x)$ whenever x is an interior point of X , but $\partial g(x)$ can be strictly larger than $\partial f(x)$ when x is a boundary point of X . In particular, restricting the domain of a convex function can create new boundary points, which can create new subgradients at these boundary points (see related Exercise 30).

Theorem 5: Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. Let $f_i : X \rightarrow \mathbb{R}$ be convex functions for $i \in \{1, \dots, k\}$. Let $w_1, \dots, w_k$ be nonnegative real numbers. Define $h : X \rightarrow \mathbb{R}$ by

$$h(x) = \sum_{i=1}^k w_i f_i(x) \quad \forall x \in X$$

For each $v \in X$ we have

$$\partial h(v) = \sum_{i=1}^k w_i \partial f_i(v)$$

where the right-hand-side is defined by Minkowski arithmetic for weighted sums of sets. In particular, $\partial h(v)$ is nonempty if and only if $\partial f_i(v)$ is nonempty for each $i \in \{1, \dots, k\}$.

Proof: The proof that $\sum_{i=1}^k w_i \partial f_i(v) \subseteq \partial h(v)$ is straightforward (see Exercise 4). The reverse inclusion is

$$\partial h(v) \subseteq \sum_{i=1}^k w_i \partial f_i(v)$$

This statement means that every subgradient $h'(v) \in \partial h(v)$ can be decomposed as

$$h'(v) = \sum_{i=1}^k w_i f'_i(v)$$

for some subgradients $f'_i(v) \in \partial f_i(v)$ for each $i \in \{1, \dots, k\}$. This reverse inclusion result is nontrivial and its proof is omitted for brevity (see [4]). □

The direction that is the easiest to prove in Theorem 5 is also the most useful: If h is a weighted sum of convex functions $f_1, \dots, f_k$ (with nonnegative weights), a subgradient of h at a point $x \in \text{dom}(h)$ can be found by using any individual subgradients $f'_1(x), \dots, f'_k(x)$ and computing a weighted sum of those. This opens the door to exercises that ask to find subgradients of a sum of convex functions (see Exercises 8, 9, 10).

IV. PROOFS FOR THEOREM 3

Suppose $X \subseteq \mathbb{R}^n$ is a nonempty convex set and $f : X \rightarrow \mathbb{R}$ is a convex function.

Lemma 3: ($\partial f(x)$ is convex) For each $x \in X$, the set $\partial f(x)$ is a convex set (possibly empty).

Proof: Fix $x \in X$. To show convexity of $\partial f(x)$, fix $s_1, s_2 \in \partial f(x)$ and $p \in [0, 1]$. We want to show $ps_1 + (1-p)s_2 \in \partial f(x)$. That is, we want to show

$$f(y) \geq f(x) + (ps_1 + (1-p)s_2)^\top (y - x) \quad \forall y \in X$$

Fix $y \in X$. We have

$$\begin{aligned} f(y) &= pf(y) + (1-p)f(y) \\ &\stackrel{(a)}{\geq} p(f(x) + s_1^\top (y - x)) + (1-p)(f(x) + s_2^\top (y - x)) \\ &= f(x) + (ps_1 + (1-p)s_2)^\top (y - x) \end{aligned}$$

where (a) holds by the fact that $s_1 \in \partial f(x)$ and $s_2 \in \partial f(x)$ (and the fact that p and $1-p$ are nonnegative). This proves the desired inequality, so $\partial f(x)$ is a convex set. $\square$

Lemma 4: ($\partial f(x)$ is closed) For each $x \in X$, the set $\partial f(x)$ is a closed set.

Proof: Fix $x \in X$. To show that $\partial f(x)$ is a closed set, let w be a limit point of $\partial f(x)$. We want to show $w \in \partial f(x)$. Since w is a limit point of $\partial f(x)$ there is a sequence of points $\{s_i\}_{i=1}^\infty$ such that $s_i \in \partial f(x)$ for all $i \in \{1, 2, 3, \dots\}$ and $\lim_{i \rightarrow \infty} s_i = w$. Now fix $y \in X$. Since $s_i \in \partial f(x)$ we have

$$f(y) \geq f(x) + s_i^\top (y - x) \quad \forall i \in \{1, 2, 3, \dots\}$$

Taking a limit as $i \rightarrow \infty$ gives

$$f(y) \geq f(x) + w^\top (y - x)$$

This holds for all $y \in X$, so $w \in \partial f(x)$. $\square$

Lemma 5: (Existence of subgradient at interior points) If v is an interior point of X then $\partial f(v)$ is nonempty.

Proof: Let v be an interior point of X . Define the set $D = \text{epi}(f) = \{(x, t) \in \mathbb{R}^{n+1} : x \in X, t \geq f(x)\}$. It can be shown that D is a convex set (the epigraph of a convex function f is always a convex set). Further, the point $(v, f(v))$ is in D and is a boundary point of D . To see this, note that $(v, f(v)) \in D$ because $v \in X$ and $f(v) \geq f(v)$. However, for each $k \in \{1, 2, 3, \dots\}$ we observe that $(v, f(v) - 1/k) \notin D$. By choosing a sufficiently large positive integer k , the point $(v, f(v) - 1/k)$ can be made arbitrarily close to $(v, f(v))$. So $(v, f(v))$ is a point in D that is arbitrarily close to points in D^c , meaning $(v, f(v))$ is a boundary point of D .

Since $D \subseteq \mathbb{R}^{n+1}$ is a convex set and $(v, f(v))$ is a boundary point of D , the hyperplane separation theorem ensures existence of a nonzero vector $\alpha = (\alpha_1, \dots, \alpha_n, \alpha_0) \in \mathbb{R}^{n+1}$ such that

$$\alpha^\top (x, t) \geq \alpha^\top (v, f(v)) \quad \forall (x, t) \in D$$

In particular,

$$(\alpha_1, \dots, \alpha_n)^\top x + \alpha_0 t \geq (\alpha_1, \dots, \alpha_n)^\top v + \alpha_0 f(v) \quad \forall (x, t) \in D \quad (4)$$

Since $(v, f(v) + 1) \in D$, we have from the above inequality

$$(\alpha_1, \dots, \alpha_n)^\top v + \alpha_0 (f(v) + 1) \geq (\alpha_1, \dots, \alpha_n)^\top v + \alpha_0 f(v)$$

and so

$$\alpha_0 \geq 0$$

This leads to two cases: The case $\alpha_0 > 0$ (the case of a *nonvertical hyperplane*) and the case $\alpha_0 = 0$ (the case of a *vertical hyperplane*). For both cases, it shall be useful to simplify (4): Since $(y, f(y)) \in D$ for all $y \in X$ we have from (4):

$$(\alpha_1, \dots, \alpha_n)^\top y + \alpha_0 f(y) \geq (\alpha_1, \dots, \alpha_n)^\top v + \alpha_0 f(v) \quad \forall y \in X \quad (5)$$

Case 1: Suppose $\alpha_0 > 0$. Define $a \in \mathbb{R}^n$ by

$$a = \frac{1}{\alpha_0} (\alpha_1, \dots, \alpha_n)$$

Dividing (5) by the positive number α_0 gives

$$a^\top y + f(y) \geq a^\top v + f(v) \quad \forall y \in X$$

Rearranging this inequality gives

$$f(y) \geq f(v) - a^\top(y - v) \quad \forall y \in X$$

and so $-a \in \partial f(v)$, that is, $\partial f(v)$ is nonempty.

Case 2: Suppose $\alpha_0 = 0$ (we reach a contradiction). Since the vector $(\alpha_0, \alpha_1, \dots, \alpha_n) \in \mathbb{R}^{n+1}$ is not the all-zero vector, but $\alpha_0 = 0$, we must have $(\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n$ is not the all zero vector. Define $b = (\alpha_1, \dots, \alpha_n)$ and note that $b \neq 0$. Using $\alpha_0 = 0$, the inequality (5) reduces to

$$b^\top y \geq b^\top v \quad \forall y \in X$$

That is

$$b^\top(y - v) \geq 0 \quad \forall y \in X \quad (6)$$

Recall that v is an interior point of X . Thus, there is an $\epsilon > 0$ such that $v - \epsilon b \in X$. Substituting $y = v - \epsilon b$ into (6) gives

$$-\epsilon \|b\|^2 \geq 0$$

which can only occur if $b = 0$ (recall that $\epsilon > 0$). But we know $b \neq 0$, a contradiction. Thus, case 2 cannot occur. $\square$

Lemma 6: (Compactness of subdifferential at interior points) If v is an interior point of X then $\partial f(v)$ is compact.

Proof: Let v be an interior point of X . We already know $\partial f(v)$ is closed and nonempty. It remains to show $\partial f(v)$ is bounded. Suppose not (we reach a contradiction). Since v is an interior point of X , there is an $\epsilon > 0$ such that the closed ball $\overline{B}_\epsilon(v)$ consists of interior points of X . It is known that every convex function is continuous at each interior point of its domain. Thus f is continuous at all points $x \in \overline{B}_\epsilon(v)$. By the extreme value theorem, it follows that the continuous function f has a maximum value over the compact set $\overline{B}_\epsilon(v)$. In particular, there is a finite value M such that

$$f(y) \leq M \quad \forall y \in \overline{B}_\epsilon(v) \quad (7)$$

On the other hand, since we have assumed $\partial f(v)$ is unbounded, there is a sequence of nonzero subgradients $s_i \in \partial f(v)$ such that $\|s_i\| \rightarrow \infty$ (we can assume they are all nonzero because their norm diverges to ∞). Fix $i \in \{1, 2, 3, \dots\}$. Since $s_i \in \partial f(v)$ we have

$$f(y) \geq f(v) + s_i^\top(y - v) \quad \forall y \in X$$

Since $B_\epsilon(v) \subseteq X$ we have

$$f(y) \geq f(v) + s_i^\top(y - v) \quad \forall y \in \overline{B}_\epsilon(v)$$

Combining the above inequality with (7) gives

$$M \geq f(v) + s_i^\top(y - v) \quad \forall y \in \overline{B}_\epsilon(v)$$

Define $y = v + \epsilon \frac{s_i}{\|s_i\|}$ and observe that $y \in \overline{B}_\epsilon(v)$. Substituting this particular y into the above inequality yields

$$M \geq f(v) + \epsilon \|s_i\|$$

This holds for all $i \in \{1, 2, 3, \dots\}$. Since $\epsilon > 0$ and $\|s_i\| \rightarrow \infty$, taking a limit as $i \rightarrow \infty$ gives $M \geq \infty$, a contradiction. $\square$

Lemma 7: (The gradient $\nabla f(x)$ is the only subgradient at x) If x is an interior point of X and $\nabla f(x)$ exists, then $\partial f(x) = \{\nabla f(x)\}$. That is, the gradient $\nabla f(x)$ is the one and only subgradient of f at the point x .

Proof: Suppose x is an interior point of X and $\nabla f(x)$ exists. Since x is an interior point, we also know by Lemma 5 that $\partial f(x)$ is nonempty. Let $f'(x)$ be any subgradient in $\partial f(x)$. We show that $\nabla f(x) = f'(x)$. Suppose $\nabla f(x) \neq f'(x)$ (we reach a contradiction). The subgradient inequality implies

$$f(y) \geq f(x) + f'(x)^\top(y - x) \quad \forall y \in X \quad (8)$$

Since x is an interior point of X and $\nabla f(x)$ exists, there is an $\epsilon > 0$ such that $\overline{B}_\epsilon(x) \subseteq X$ and

$$f(y) = f(x) + \nabla f(x)^\top(y - x) + g(y - x) \quad \text{whenever } \|y - x\| \leq \epsilon \quad (9)$$

for some real-valued function $g(h)$ that satisfies $g(0) = 0$ and

$$\lim_{h \rightarrow 0} \frac{g(h)}{\|h\|} = 0 \quad (10)$$

Combining (8) and (9) gives

$$f(x) + \nabla f(x)^\top(y - x) + g(y - x) \geq f(x) + f'(x)^\top(y - x) \quad \text{whenever } \|y - x\| \leq \epsilon$$

In particular

$$(\nabla f(x) - f'(x))^\top(y - x) + g(y - x) \geq 0 \quad \text{whenever } \|y - x\| \leq \epsilon \quad (11)$$

For each $\delta > 0$ define

$$h_\delta = \delta(f'(x) - \nabla f(x))$$

We have assumed $f'(x) \neq \nabla f(x)$, so we know $h_\delta \neq 0$ whenever $\delta > 0$. For all sufficiently small $\delta > 0$ we have $\|h_\delta\| \leq \epsilon$ and substituting $y = x + h_\delta$ into (11) gives (for all sufficiently small $\delta > 0$):

$$-\delta\|\nabla f(x) - f'(x)\|^2 + g(h_\delta) \geq 0$$

Dividing both sides by the positive value $\|h_\delta\| = \delta\|f'(x) - \nabla f(x)\|$ gives (for all sufficiently small $\delta > 0$):

$$-\|\nabla f(x) - f'(x)\| + \frac{g(h_\delta)}{\|h_\delta\|} \geq 0$$

Taking $\delta \rightarrow 0^+$ means $\|h_\delta\| \rightarrow 0$ and so (10) implies

$$-\|\nabla f(x) - f'(x)\| \geq 0$$

which can only hold if $\nabla f(x) = f'(x)$. □

V. PROJECTED SUBGRADIENT DESCENT

Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, and convex set. Let $X \subseteq \mathbb{R}^n$ be a convex set that contains C . Let $f : X \rightarrow \mathbb{R}$ be a convex function. The goal of this section is to develop an algorithm that finds an *approximate solution* to the problem:

$$\text{Minimize: } f(x) \tag{12}$$

$$\text{Subject to: } x \in C \tag{13}$$

It is assumed throughout that:

- Problem (12)-(13) has at least one optimal solution $x^* \in C$. The extreme value theorem ensures that this holds whenever f is continuous over all $x \in C$ and when C is both closed and bounded.
- $\partial f(x)$ is nonempty for each $x \in C$. By Lemma 2, this holds whenever X is an open set, such as whenever $X = \mathbb{R}^n$. It also holds when f can be extended to a convex function $\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}$, so that $\tilde{f}(x) = f(x)$ for all $x \in X$.

A. Projections

When $C \subseteq \mathbb{R}^n$ is a nonempty convex set, for each $w \in \mathbb{R}^n$ there is a unique point in C that is closest to w . Let $\pi_C(w)$ denote this unique closest point, called the *projection of w onto the convex set C* . It is important that C is a nonempty closed set (else a closest point in C may not exist). It is also important that C is a convex set (else there might be more than one closest point in C). Given $w \in \mathbb{R}^n$, the projection $\pi_C(w)$ is computed by solving the convex program

$$\text{Minimize: } \|w - x\|^2 \tag{14}$$

$$\text{Subject to: } x \in C \tag{15}$$

where the norm $\|\cdot\|$ represents the standard Euclidean norm $\|v\| = \sqrt{\sum_{i=1}^n v_i^2}$ for $v \in \mathbb{R}^n$. The problem (14)-(15) replaces the objective function in (12)-(13) with a quadratic function.² An important assumption in our numerical method for approximately solving (12)-(13) is that the problem (14)-(15) can be easily solved. The problem (14)-(15) has easy solutions when set C is simple, for example

- Suppose $C = \mathbb{R}^n$. Then

$$\pi_C(w) = w \quad \forall w \in \mathbb{R}^n$$

- Suppose C is a box. That is,

$$C = \{x \in \mathbb{R}^n : a_i \leq x_i \leq b_i \quad \forall i \in \{1, \dots, n\}\}$$

for some given real numbers a_i, b_i where $a_i < b_i$ for all $i \in \{1, \dots, n\}$. Then for each $w \in \mathbb{R}^n$, we can obtain $\pi_C(w)$ by separately projecting each component of w onto its corresponding interval. That is,

$$\pi_C(w) = ([w_1]_{a_1}^{b_1}, [w_2]_{a_2}^{b_2}, \dots, [w_n]_{a_n}^{b_n})$$

where $[z]_a^b$ denotes the projection of $z \in \mathbb{R}$ onto the interval $[a, b]$:

$$[z]_a^b = \begin{cases} b & \text{if } z > b \\ z & \text{if } z \in [a, b] \\ a & \text{if } z < a \end{cases}$$

²In fact, the objective function in (14) is *strongly convex*, which is defined in Section VII.

- Suppose C is a closed ball. Specifically, C is the closed ball of radius $r > 0$ about some point $a \in \mathbb{R}^n$, so $C = \overline{B}_r(a)$. Then

$$\pi_C(w) = \begin{cases} w & \text{if } \|w - a\| \leq r \\ a + r \frac{(w-a)}{\|w-a\|} & \text{if } \|w - a\| > r \end{cases}$$

where $\|\cdot\|$ denotes the Euclidean norm.

When the closed and convex set C has a more complicated structure, then the solution to (14)-(15) may not have a closed form and may be difficult to obtain. In that case, other numerical algorithms, such as dual and primal-dual algorithms, must be used to provide an *approximate projection* $\pi_C(w)$. Fortunately, these other algorithms can be fine tuned to operate on quadratic objectives of the type $\|w - x\|^2$. These quadratic objective functions are *strongly convex* (as defined in Section VII) and hence can typically be optimized more quickly in comparison to general convex objective functions.

Theorem 6: (Normality condition) Let $C \subseteq \mathbb{R}^n$ be a nonempty, closed, convex set. Fix $w \in \mathbb{R}^n$. Then $v = \pi_C(w)$ if and only if $v \in C$ and

$$(w - v)^\top (x - v) \leq 0 \quad \forall x \in C$$

Proof: We apply the normal cone condition for minimizing a differentiable convex function over a convex set C : A point $v \in C$ minimizes f over all $x \in C$ if and only if $-\nabla f(v)^\top (x - v) \leq 0$ for all $x \in C$. Here we use the differentiable function $f(x) = \|x - w\|^2 = (x - w)^\top (x - w)$, which has gradient $\nabla f(x) = 2(x - w)$. The gradient at point v is then $2(v - w)$. So v minimizes f over all $x \in C$ if and only if $-2(v - w)^\top (x - v) \leq 0$ for all $x \in C$. $\square$

Lemma 8: (Projections shrink distances) Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set. Then

$$\|\pi_C(x) - \pi_C(y)\| \leq \|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

where the Euclidean norm is used.

Proof: This can be proven as a consequence of the normality condition (see Exercise 28). $\square$

The normality condition can sometimes be used to find $\pi_C(w)$, and/or to prove that reasonable guesses for $\pi_C(w)$ are correct (see Exercise 16 for projection onto a probability simplex).

B. Projected subgradient descent

To treat the problem (12)-(13), we consider an algorithm that utilizes an oracle that, when queried about a particular point $x \in C$, returns a subgradient $f'(x) \in \partial f(x)$. It is sometimes useful to also assume the oracle provides the function value $f(x)$. These notes focus attention on the following *subgradient descent algorithm*:

- Fix $x_0 \in C$ arbitrarily.
- For each step $k \in \{0, 1, 2, \dots\}$ do:
 - 1) Query the oracle about x_k to receive a particular $f'(x_k) \in \partial f(x_k)$.
 - 2) Perform the iteration:

$$x_{k+1} = \pi_C(x_k - \delta_k f'(x_k))$$

where $\delta_k > 0$ is a particular *stepsize* used for step k .

After m iterations, the resulting vectors $\{x_0, x_1, x_2, \dots, x_{m-1}\}$ are used to construct an approximate solution $\tilde{x}_m \in C$. The accuracy gets better when m is large. Specifically, we shall show that for a particular choice of stepsizes, we have

$$f(\tilde{x}_m) \leq f(x^*) + \epsilon_m$$

where $\epsilon_m > 0$ is a value that is small when m is large. In particular, we get better accuracy when more iterations are used. In some special cases, such as cases when a *strong convexity* assumption holds, the optimal vector x^* is *unique* and we can obtain bounds on $\|\tilde{x}_m - x^*\|$.

C. Convex programs

The problem (12)-(13) seems restrictive because it does not include functional inequality constraints. In particular, a *convex program* is a problem of the type

$$\text{Minimize: } f(x) \tag{16}$$

$$\text{Subject to: } g_i(x) \leq 0 \quad \forall i \in \{1, \dots, a\} \tag{17}$$

$$h_j(x) = 0 \quad \forall j \in \{1, \dots, b\} \tag{18}$$

$$x \in X \tag{19}$$

where $X \subseteq \mathbb{R}^n$ is a convex set; $f : X \rightarrow \mathbb{R}$ is a convex function; $g_i : X \rightarrow \mathbb{R}$ are convex functions for all $i \in \{1, \dots, a\}$; $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are affine functions for all $j \in \{1, \dots, b\}$. However, this problem can be expressed in the form (12)-(13) by defining the convex set:

$$C = X \cap \{x \in X : g_i(x) \leq 0 \quad \forall i \in \{1, \dots, a\}\} \cap \{x \in \mathbb{R}^n : h_j(x) = 0 \quad \forall j \in \{1, \dots, b\}\}$$

The set C is closed whenever X is closed and all g_i functions are continuous (the affine functions h_j are always continuous). Of course, C is typically more complicated than a box or a ball, so an exact projection $\pi_C(w)$ may not be available for the update step $x_{k+1} = \pi_C(x_k - \delta_k f'(x_k))$. In this case, the projection must be approximated by some (possibly dual or primal-dual) numerical algorithm that, on each step k , defines $w_k = x_k - \delta_k f'(x_k)$ and seeks an $x_{k+1} \in X$ that (approximately) solves

$$\begin{aligned} \text{Minimize:} \quad & \|w_k - x\|^2 \\ \text{Subject to:} \quad & g_i(x) \leq 0 \quad \forall i \in \{1, \dots, a\} \\ & h_j(x) = 0 \quad \forall j \in \{1, \dots, b\} \\ & x \in X \end{aligned}$$

In particular, the desired convex program (16)-(19) is treated using an iterative scheme that, on each step $k \in \{0, 1, 2, \dots\}$, finds an approximate solution to a problem that replaces the convex objective $f(x)$ with a quadratic objective $\|w_k - x\|^2$. Such quadratic objectives are strongly convex (as defined in Section VII) and hence can typically be minimized quickly by other algorithms, such as dual or primal-dual algorithms.

D. B -Lipschitz continuity and B -bounded subgradients

Definition 4: Fix $X \subseteq \mathbb{R}^n$ as a nonempty set. Fix $B \geq 0$. A function $f : X \rightarrow \mathbb{R}$ is B -Lipschitz continuous if

$$|f(x) - f(y)| \leq B\|x - y\| \quad \forall x, y \in X$$

where Euclidean norm is used.

Definition 5: Fix $X \subseteq \mathbb{R}^n$ as a nonempty convex set. Let $f : X \rightarrow \mathbb{R}$ be a convex function. Fix $B \geq 0$. Let $C \subseteq X$. The function f has B -bounded subgradients over all $x \in C$ if for all $x \in C$ there is at least one subgradient $f'(x) \in \partial f(x)$ such that $\|f'(x)\| \leq B$, where Euclidean norm is used. The function f has B -bounded subgradients if it has B -bounded subgradients over all x in its domain X .

Theorem 7: Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. Let $f : X \rightarrow \mathbb{R}$ be a convex function. Fix $B \geq 0$. The function f is B -Lipschitz continuous if and only if it has B -bounded subgradients.

Proof: This follows by the *McShane-Whitney extension theorem* (see [6] and Theorem 2.1 in [7]). The result is developed in Exercises 37, 38, 39. $\square$

For example, the convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ given by $f(x) = \|x\|^2$ has subgradient $f'(x) = \nabla f(x) = 2x$ for $x \in \mathbb{R}^n$. Then f does not have bounded subgradients over all $\mathbb{R}^n$, but does have bounded subgradients over any bounded subset $C \subseteq \mathbb{R}^n$.

E. Analysis of projected subgradient descent (fixed stepsize)

Theorem 8: (Possibly nonsmooth convex functions – fixed stepsize) Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, and convex set. Let $f : C \rightarrow \mathbb{R}$ be a convex function that has B -bounded subgradients for some $B \geq 0$ (and our oracle provides such subgradients). Suppose $x^* \in C$ minimizes f over all $x \in C$. Fix $\delta > 0$ as a constant stepsize. Fix $x_0 \in C$ and do

$$x_{k+1} = \pi_C(x_k - \delta f'(x_k)) \quad \forall k \in \{0, 1, 2, \dots\}$$

Then for all positive integers m we have

$$f\left(\frac{1}{m} \sum_{k=0}^{m-1} x_k\right) \leq f(x^*) + \frac{\delta B^2}{2} + \frac{\|x_0 - x^*\|^2}{2m\delta} \quad (20)$$

and also

$$\min\{f(x_0), f(x_1), \dots, f(x_{m-1})\} \leq f(x^*) + \frac{\delta B^2}{2} + \frac{\|x_0 - x^*\|^2}{2m\delta} \quad (21)$$

Therefore, given any $\delta > 0$, we can define $\tilde{x}_m$ either by $\tilde{x}_m = \frac{1}{m} \sum_{k=0}^{m-1} x_k$, or by $\tilde{x}_m \in \arg \min_{x_k \in \{x_0, \dots, x_{m-1}\}} \{f(x_k)\}$, and

$$f(\tilde{x}_m) \leq f(x^*) + O(\delta) \quad \forall m \geq 1/\delta^2$$

So we obtain an $O(\delta)$ -approximation after a number of iterations that is at least $1/\delta^2$.

Proof: Fix $k \in \{0, 1, 2, \dots\}$. We have $x_k \in C$ and

$$x_{k+1} = \pi_C(x_k - \delta f'(x_k))$$

Throughout we use $\|\cdot\|$ as the Euclidean norm. Since $x^* \in C$ we have $x^* = \pi_C(x^*)$. Then

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|\pi_C(x_k - \delta f'(x_k)) - x^*\|^2 \\ &\stackrel{(a)}{=} \|\pi_C(x_k - \delta f'(x_k)) - \pi_C(x^*)\|^2 \\ &\stackrel{(b)}{\leq} \|x_k - \delta f'(x_k) - x^*\|^2 \\ &= \|x_k - x^*\|^2 + \delta^2 \|f'(x_k)\|^2 + 2\delta f'(x_k)^\top (x^* - x_k) \\ &\stackrel{(c)}{\leq} \|x_k - x^*\|^2 + \delta^2 \|f'(x_k)\|^2 + 2\delta(f(x^*) - f(x_k)) \\ &\leq \|x_k - x^*\|^2 + \delta^2 B^2 + 2\delta(f(x^*) - f(x_k)) \end{aligned}$$

where (a) follows because $x^* = \pi_C(x^*)$; (b) uses the fact that projections do not increase distances (Lemma 8); (c) uses the subgradient inequality (3), namely

$$f(x^*) \geq f(x_k) + f'(x_k)^\top (x^* - x_k)$$

Rearranging gives

$$f(x_k) \leq f(x^*) + \frac{\delta B^2}{2} + \frac{\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2}{2\delta} \quad \forall k \in \{0, 1, 2, \dots\} \quad (22)$$

Fix m as a positive integer. Summing the above over $k \in \{0, 1, 2, \dots, m-1\}$ gives

$$\sum_{k=0}^{m-1} f(x_k) \leq m f(x^*) + \frac{m \delta B^2}{2} + \frac{\|x_0 - x^*\|^2 - \|x_m - x^*\|^2}{2\delta}$$

where we have used *telescoping sums*:

$$(\|x_0 - x^*\|^2 - \|x_1 - x^*\|^2) + (\|x_1 - x^*\|^2 - \|x_2 - x^*\|^2) + \dots + (\|x_{m-1} - x^*\|^2 - \|x_m - x^*\|^2) = \|x_0 - x^*\|^2 - \|x_m - x^*\|^2$$

Dividing by m gives

$$\begin{aligned} \frac{1}{m} \sum_{k=0}^{m-1} f(x_k) &\leq f(x^*) + \frac{\delta B^2}{2} + \frac{\|x_0 - x^*\|^2 - \|x_m - x^*\|^2}{2m\delta} \\ &\leq f(x^*) + \frac{\delta B^2}{2} + \frac{\|x_0 - x^*\|^2}{2m\delta} \end{aligned}$$

from which we obtain the result (20) by Jensen's inequality on the convex function f :

$$f\left(\frac{1}{m} \sum_{k=0}^{m-1} x_k\right) \leq \frac{1}{m} \sum_{k=0}^{m-1} f(x_k)$$

and the result (21) by the fact

$$\min\{f(x_0), f(x_1), \dots, f(x_{m-1})\} \leq \frac{1}{m} \sum_{k=0}^{m-1} f(x_k)$$

□

A note on $\tilde{x}_m$:

- Using $\tilde{x}_m = \frac{1}{m} \sum_{k=0}^{m-1} x_k$ is useful when the oracle only returns $f'(x_k)$ at each step k , rather than both $f'(x_k)$ and $f(x_k)$.
- Using $\tilde{x}_m = x_j$ for the index j with the smallest value of $f(x_j)$ over all points $\{x_1, \dots, x_{m-1}\}$ is useful when $f(x_k)$ values are known.

This result shows that the stepsize δ relates directly to accuracy of the approximation, but requires a number of iterations that is $1/\delta^2$. This is a fundamental tradeoff between accuracy and algorithm complexity. Well known converse-type bounds in [1] show that this $1/\delta^2$ tradeoff is the *best we can do* for minimizing a nonsmooth convex function by using a large class of subgradient-based algorithms, including any algorithm that computes each iterate as a linear combination of previous subgradients. This is done by constructing a nonsmooth convex function that is the “worst function in the world” and showing that no algorithm within the specified class can produce an accuracy within δ using a number of iterations that is smaller than $\Theta(1/\delta^2)$. A detail is that the converse result of [1] has additional restrictions, for example, it only applies to the first n iterations. Nevertheless, there are no known subgradient algorithms for minimizing nonsmooth convex functions that converge faster than $\Theta(1/\delta^2)$. Such algorithms are often called *first order* algorithms because they do not use second derivative information.

F. Decreasing stepsize

This subsection considers projected subgradient descent using a decreasing stepsize

$$\delta_k = \frac{1}{\sqrt{1+k}} \quad \forall k \in \{0, 1, 2, \dots\} \quad (23)$$

The advantage is that we obtain

$$f(\tilde{x}_m) \leq f(x^*) + O(1/\sqrt{m}) \quad \forall m \in \{1, 2, 3, \dots\}$$

and so $f(\tilde{x}_m) \rightarrow f(x^*)$ as $m \rightarrow \infty$. In particular, with a fixed stepsize $\delta > 0$, there is little benefit to running the algorithm beyond $\Theta(1/\delta^2)$ iterations. In contrast, this vanishing stepsize method continues to improve as m gets larger and larger. In addition to the bounded subgradient assumption, it is convenient to assume the set C is compact and has a *bounded diameter*:

Definition 6: A set $C \subseteq \mathbb{R}^n$ has *D-bounded diameter* (for some $D \geq 0$) if

$$\|x - y\| \leq D \quad \forall x, y \in C \quad (24)$$

where Euclidean norm is used.

Theorem 9: (Possibly nonsmooth convex functions – vanishing stepsize) Fix $C \subseteq \mathbb{R}^n$ as a nonempty, compact, convex set with D -bounded diameter for some $D \geq 0$. Let $f : C \rightarrow \mathbb{R}$ be a convex function with B -bounded subgradients for some $B \geq 0$ (and our oracle provides such subgradients). Suppose $x^* \in C$ minimizes f over all $x \in C$. Fix $x_0 \in C$ and do

$$x_{k+1} = \pi_C(x_k - \delta_k f'(x_k)) \quad \forall k \in \{0, 1, 2, \dots\}$$

where δ_k satisfies (23). Then for all positive integers m we have

$$f\left(\frac{1}{m} \sum_{k=0}^{m-1} x_k\right) \leq f(x^*) + \frac{B^2}{2} \left(\frac{2}{\sqrt{m}} - \frac{1}{m}\right) + \frac{D^2 \sqrt{m+1}}{2m}$$

and also

$$\min\{f(x_0), f(x_1), \dots, f(x_{m-1})\} \leq f(x^*) + \frac{B^2}{2} \left(\frac{2}{\sqrt{m}} - \frac{1}{m}\right) + \frac{D^2 \sqrt{m+1}}{2m}$$

so the error decays like $O(1/\sqrt{m})$. (Of course, the time to drive error below a given value $\delta > 0$ remains $\Theta(1/\delta^2)$).

Proof: Using the same proof method to reach (22) we obtain for all $k \in \{0, 1, 2, \dots\}$

$$\begin{aligned} f(x_k) &\leq f(x^*) + \frac{\delta_k B^2}{2} + \frac{\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2}{2\delta_k} \\ &\leq f(x^*) + \frac{\delta_k B^2}{2} + \frac{\|x_k - x^*\|^2}{2\delta_k} - \frac{\|x_{k+1} - x^*\|^2}{2\delta_{k+1}} \\ &\quad + \left(\frac{1}{\delta_{k+1}} - \frac{1}{\delta_k}\right) \frac{\|x_{k+1} - x^*\|^2}{2} \\ &\leq f(x^*) + \frac{\delta_k B^2}{2} + \frac{\|x_k - x^*\|^2}{2\delta_k} - \frac{\|x_{k+1} - x^*\|^2}{2\delta_{k+1}} \\ &\quad + \left(\frac{1}{\delta_{k+1}} - \frac{1}{\delta_k}\right) \frac{D^2}{2} \end{aligned}$$

where the final inequality uses the bounded diameter assumption (24) and the fact $1/\delta_{k+1} \geq 1/\delta_k$. Fix m as a positive integer. Summing the above inequality over all $k \in \{0, 1, \dots, m-1\}$ and dividing by m gives

$$\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \frac{B^2}{2m} \sum_{k=0}^{m-1} \delta_k + \frac{\|x_0 - x^*\|^2}{2m\delta_0} - \frac{\|x_m - x^*\|^2}{2m\delta_m} + \left(\frac{1}{\delta_m} - \frac{1}{\delta_0}\right) \frac{D^2}{2m}$$

and so

$$\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \frac{B^2}{2m} \sum_{k=0}^{m-1} \delta_k + \frac{D^2 \sqrt{m+1}}{2m}$$

which proves the result upon application of the inequality

$$\sum_{k=0}^{m-1} \delta_k = \sum_{i=1}^m \frac{1}{\sqrt{i}} \leq 1 + \int_1^m \frac{1}{\sqrt{t}} dt = 2m^{1/2} - 1$$

□

G. Oracle complexity and stochastic subgradients

The “oracle” is really software that is used to compute $f(x_k)$ and $f'(x_k)$ for a given $x_k \in C$ on step k of the projected subgradient descent algorithm. Suppose we have a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that is a sum of 1000 terms:

$$f(x) = \sum_{j=1}^{1000} f_j(x) \quad \forall x \in \mathbb{R}^n \quad (25)$$

Therefore a particular $f'(x) \in \partial f(x)$ is

$$f'(x) = \sum_{j=1}^{1000} f'_j(x) \quad \forall x \in \mathbb{R}^n$$

where $f'_j(x) \in \partial f_j(x)$ for all $j \in \{1, \dots, 1000\}$ and $x \in \mathbb{R}^n$. Given $x \in C$, computation of $f(x)$ involves computing 1000 component functions $f_j(x)$ and then summing the result. Similarly, computation of $f'(x)$ involves computing 1000 component subgradients $f'_j(x)$. Thus, computing f is costly. Computing $f'(x)$ is also costly. Using an algorithm that only requires $f'(x)$ can roughly save a factor of 2 in *oracle complexity*. This can be implemented by using

$$\tilde{x}_m = \frac{1}{m} \sum_{k=0}^{m-1} x_k$$

rather than

$$\tilde{x}_m = \arg \min_{x_k \in \{x_0, \dots, x_{m-1}\}} \{f(x_k)\}$$

The *total complexity* of subgradient descent is the number of iterations m multiplied by the complexity of each iteration. Assuming the projection step is easy, the complexity of each iteration is essentially the oracle complexity. Using an oracle that only provides $f'(x)$ reduces oracle complexity and suffers no impact in the error bounds. Indeed, the error bounds in Theorem 8 are the same for both choices of $\tilde{x}_m$, as are the bounds in Theorem 9.

Can we reduce the oracle complexity of computing $f'(x)$? A common method that decreases oracle complexity by a factor of 1000 for this example is to use an oracle that returns a *stochastic subgradient* S_k on each step k . Specifically, when the oracle is given a query $x_k \in C$ on step k it performs the following procedure:

- 1) Independently generate a random variable $J_k \in \{1, \dots, 1000\}$, uniformly over all 1000 choices.
- 2) Compute $f'_{J_k}(x_k)$.
- 3) Return the *stochastic subgradient* $S_k = 1000 f'_{J_k}(x_k)$.

This only requires subgradient computation of *one randomly chosen component function*, rather than all 1000 component functions. Neglecting the complexity of generating the random variable J_k , this new procedure is 1000 times faster. It does not return the *exact* subgradient. However, the random vector S_k that it returns is called a *stochastic subgradient* because its expectation is equal to the exact subgradient:

$$\mathbb{E}[S_k] = \sum_{j=1}^{1000} \underbrace{\mathbb{E}[S_k | J_k = j]}_{1000 f'_j(x_k)} \underbrace{P[J_k = j]}_{1/1000} = f'(x_k)$$

The stochastic subgradient S_k can be viewed as a “noisy” version of the true subgradient $f'(x_k)$.

In this example, it is 1000 times easier to compute the noisy version in comparison to the true $f'(x_k)$ value. Can S_k be a good replacement for the actual subgradient? This would use the iteration

$$x_{k+1} = \pi_C(x_k - \delta_k S_k) \quad \forall k \in \{0, 1, 2, \dots\} \quad (26)$$

for some choice of stepsizes δ_k . Strictly speaking, this makes each iterate x_k a *random vector*. For both fixed stepsize and vanishing stepsize, the analysis in Theorems 8 and 9 can be repeated by taking expectations of the corresponding inequalities.³ Of course, the number of iterations needed can increase. This is because the corresponding value of B can increase:

$$\mathbb{E}[\|S_k\|^2 | x_k] = \sum_{j=1}^{1000} \mathbb{E}[\|S_k\|^2 | J_k = j, x_k] \frac{1}{1000} = \sum_{j=1}^{1000} 1000^2 \|f'_j(x_k)\|^2 \frac{1}{1000} = 1000 \sum_{j=1}^{1000} \|f'_j(x_k)\|^2$$

and

$$\|f'(x_k)\|^2 = \|\sum_{j=1}^{1000} f'_j(x_k)\|^2 = 1000^2 \|\frac{1}{1000} \sum_{j=1}^{1000} f'_j(x_k)\|^2 \stackrel{(a)}{\leq} 1000^2 \cdot \frac{1}{1000} \sum_{j=1}^{1000} \|f'_j(x_k)\|^2 = 1000 \sum_{j=1}^{1000} \|f'_j(x_k)\|^2$$

³It is important to assume J_k is generated independently of the past. In particular, we need J_k and x_k to be independent.

where inequality (a) uses Jensen's inequality for the convex $\|\cdot\|^2$ function. The amount by which $\|f'(x_k)\|^2$ is smaller than $\mathbb{E}[\|S_k\|^2|x_k]$ depends on the *Jensen inequality gap*. For example, if the $f'(x_k)$ subgradients happen to be pairwise orthogonal, then $\|f'(x_k)\|^2 = \sum_{j=1}^{1000} \|f'_j(x_k)\|^2$, which is a factor of 1000 smaller. However, simulations on example f functions of the type (25) often reveal a net savings in the *total algorithm complexity* (number of iterations multiplied by oracle complexity) when using a stochastic subgradient. See Exercise 36 for another type of stochastic gradient. See Exercise 42 for analysis of the stochastic gradient method for these general (possibly nonsmooth) problems using a fixed stepsize.

H. Projection as a minimization

Consider the following two methods for choosing $x_{k+1} \in C$:

- Projection method (fix $\delta_k > 0$):

$$x_{k+1} = \pi_C(x_k - \delta_k f'(x_k))$$

- Minimization method (fix $\alpha_k > 0$):

$$x_{k+1} = \arg \min_{x \in C} \left\{ f'(x_k)^\top (x - x_k) + \frac{\alpha_k}{2} \|x - x_k\|^2 \right\}$$

It can be shown that both methods produce a unique value for x_{k+1} . The next result shows the two methods are identical when $\alpha_k = 1/\delta_k$. This is useful. In particular, thinking of the update rule as a minimization can help to get extended convergence results in smooth and strongly convex regimes, and can also help to develop an improved mirror descent algorithm that is important when C is a probability simplex.

Lemma 9: The projection method is identical to the minimization method when $\alpha_k = 1/\delta_k$.

Proof: Fix $\delta_k > 0$. Assume $\alpha_k = 1/\delta_k$. The projection method chooses x_{k+1} as the solution to the problem

$$\begin{aligned} \text{Minimize: } & \|x_k - \delta_k f'(x_k) - x\|^2 \\ \text{Subject to: } & x \in C \end{aligned}$$

where we use the Euclidean norm. The objective function to minimize is

$$\begin{aligned} \|x_k - \delta_k f'(x_k) - x\|^2 &= \|x_k - x\|^2 + \delta_k^2 \|f'(x_k)\|^2 + 2\delta_k f'(x_k)^\top (x - x_k) \\ &= 2\delta_k \left(f'(x_k)^\top (x - x_k) + \frac{1}{2\delta_k} \|x - x_k\|^2 \right) + \delta_k^2 \|f'(x_k)\|^2 \\ &= 2\delta_k \left(f'(x_k)^\top (x - x_k) + \frac{\alpha_k}{2} \|x - x_k\|^2 \right) + \delta_k^2 \|f'(x_k)\|^2 \end{aligned}$$

Since the term $\delta_k^2 \|f'(x_k)\|^2$ does not depend on x , and since $\delta_k > 0$, minimizing this objective over all $x \in C$ is equivalent to minimizing the expression

$$f'(x_k)^\top (x - x_k) + \frac{\alpha_k}{2} \|x - x_k\|^2$$

over all $x \in C$, which is the same as the minimization method. $\square$

I. Mirror descent for optimization over a probability simplex

This subsection presents a modified algorithm that improves convergence time in certain cases. For simplicity, we focus on the case when C is a probability simplex. While the modified algorithm does not change the $\Theta(1/\delta^2)$ behavior, it significantly improves on the constant coefficient that multiplies $1/\delta^2$, often reducing it from $O(n)$ to $O(\log(n))$. This is crucial when n is large, such as when we perform online portfolio optimization over a large number of stocks (Section VI-A).

Suppose $C = P$ where $P \subseteq \mathbb{R}^n$ is the probability simplex

$$P = \{(x_1, \dots, x_n) \in \mathbb{R}^n : x_i \geq 0 \quad \forall i \in \{1, \dots, n\}, \sum_{i=1}^n x_i = 1\} \quad (27)$$

Define $P_0 \subseteq P$ by

$$P_0 = \{(x_1, \dots, x_n) \in \mathbb{R}^n : x_i > 0 \quad \forall i \in \{1, \dots, n\}, \sum_{i=1}^n x_i = 1\}$$

Fix $\alpha > 0$. Initialize $x_0 \in P_0$. Replace the quadratic term in the minimization method with a function $D(x; y)$:

$$x_{k+1} = \arg \min_{x \in P} \{ f'(x_k)^\top (x - x_k) + \alpha D(x; x_k) \} \quad \forall k \in \{0, 1, 2, \dots\} \quad (28)$$

where $D : P \times P_0 \rightarrow \mathbb{R}$ is defined

$$D(x; y) = \sum_{i=1}^n x_i \log(x_i/y_i) \quad \forall x \in P, y \in P_0$$

where $\log(\cdot)$ denotes a natural logarithm. The function $D(\cdot)$ is the *Kullback-Leibler (K-L) divergence function*. For historical reasons, this modified method is called *mirror descent*. The method can be generalized to treat more general *Bregman divergence* functions, but that shall not be pursued in these notes.

It is well known that $D(\cdot)$ is a nonnegative function that satisfies

$$D(x; (1/n, \dots, 1/n)) \leq \log(n) \quad \forall x \in P \quad (29)$$

$$D(x; y) \geq \frac{1}{2} \|x - y\|_1^2 \quad \forall x \in P, y \in P_0 \quad (30)$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$ is the L_1 norm. Inequality (30) is the *Pinsker inequality*. The following *pushback lemma* is standard [8][9][10]:

Theorem 10: (K-L Pushback) Fix $y \in P_0$, $\alpha \geq 0$. Let $A \subseteq P$ be a convex set and $f : A \rightarrow \mathbb{R}$ a convex function. Consider:

$$\text{Minimize: } f(x) + \alpha D(x; y) \quad (31)$$

$$\text{Subject to: } x \in A \quad (32)$$

If x^{opt} solves (31)-(32), and if $x^{opt} \in P_0$, then for all $x \in A$:

$$f(x^{opt}) + \alpha D(x^{opt}; y) \leq f(x) + \alpha D(x; y) - \underbrace{\alpha D(x, x^{opt})}_{\text{pushback}}$$

The lemma can be understood as follows: The definition of x^{opt} being a minimizer of $f(x) + \alpha D(x; y)$ over all $x \in A$ is that

$$f(x^{opt}) + \alpha D(x^{opt}; y) \leq f(x) + \alpha D(x; y) \quad \forall x \in A$$

Lemma 10 shows that the structure of the D function enables a strengthened version of this inequality that includes a *pushback term* (shown in the underbrace in the lemma).

The condition that $x^{opt} \in P_0$ is typically satisfied when this K-L pushback result is applied. For example, suppose f is an affine function:

$$f(x) = c^\top x + b \quad \forall x \in \mathbb{R}^n$$

for some $c \in \mathbb{R}^n$ and some $b \in \mathbb{R}$. It can be shown that for any $y \in P_0$ and any $\alpha > 0$, the minimizer of $f(x) + \alpha D(x; y)$ over all $x \in P$ is

$$x_i = \frac{y_i e^{-c_i/\alpha}}{\sum_{j=1}^n y_j e^{-c_j/\alpha}} \quad \forall i \in \{1, \dots, n\} \quad (33)$$

It can be directly verified that these x_i values are strictly positive and sum to 1, so they are in P_0 .

For the following theorem it is useful to distinguish between the L_2 norm, L_1 norm, and L_∞ norm: Given $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ we have

$$\begin{aligned} \|x\|_2 &= \sqrt{\sum_{i=1}^n x_i^2} \\ \|x\|_1 &= \sum_{i=1}^n |x_i| \\ \|x\|_\infty &= \max\{|x_1|, \dots, |x_n|\} \end{aligned}$$

Theorem 11: (Performance under mirror descent) Consider the problem of minimizing a continuous and potentially nonsmooth convex function $f : P \rightarrow \mathbb{R}$ over all $x \in P$, where P is the probability simplex in (27). Assume f has subgradients $f'(x)$ at all $x \in P$ (given to us by an oracle) that satisfy

$$\|f'(x)\|_\infty \leq H \quad \forall x \in P$$

where H is some nonnegative constant. Let $x^* \in P$ be an optimal solution (the extreme value theorem ensures x^* exists). Fix $x_0 = (1/n, 1/n, \dots, 1/n)$. Fix $\alpha > 0$. Define

$$x_k = (x_{k,1}, \dots, x_{k,n})$$

The mirror descent iteration (28) reduces to the following for all $k \in \{0, 1, 2, 3, \dots\}$:

$$x_{k+1,i} = \frac{x_{k,i} e^{-c_{k,i}/\alpha}}{\sum_{j=1}^n x_{k,j} e^{-c_{k,j}/\alpha}} \quad \forall i \in \{1, \dots, n\} \quad (34)$$

where $c_{k,i}$ is the i th component of $f'(x_k)$. Under this algorithm we have for all positive integers m :

$$\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \frac{H^2}{2\alpha} + \frac{\alpha \log(n)}{m}$$

In particular, fixing $\delta > 0$ and defining $\alpha = 1/\delta$ yields an $O(\delta)$ approximation $\tilde{x}_m$ after a number of iterations $1/\delta^2$. As before, the $\tilde{x}_m$ vector can be defined either by $\frac{1}{m} \sum_{k=0}^{m-1} x_k$, or by the best x_k vector seen over $k \in \{0, \dots, m-1\}$.

Proof: The fact that (34) holds follows directly from (33). In particular, it means that $x_k \in P_0$ for all $k \in \{0, 1, 2, \dots\}$.

Fix $k \in \{0, 1, 2, \dots\}$. By definition of x_{k+1} being the minimizer of $f'(x_k)^\top (x - x_k) + \alpha D(x, x_k)$ over all $x \in P$, we have by the K-L pushback theorem (using the fact that $x^* \in P$):

$$f'(x_k)^\top (x_{k+1} - x_k) + \alpha D(x_{k+1}; x_k) \leq f'(x_k)^\top (x^* - x_k) + \alpha D(x^*; x_k) - \alpha D(x^*; x_{k+1})$$

By (30) we obtain

$$D(x_{k+1}; x_k) \geq \frac{\|x_{k+1} - x_k\|_1^2}{2}$$

We also have

$$f'(x_k)^\top (x_{k+1} - x_k) \geq -\|f'(x_k)\|_\infty \cdot \|x_{k+1} - x_k\|_1$$

and so

$$\begin{aligned} -H\|x_{k+1} - x_k\|_1 + \frac{\alpha\|x_{k+1} - x_k\|_1^2}{2} &\leq f'(x_k)^\top (x^* - x_k) + \alpha D(x^*; x_k) - \alpha D(x^*; x_{k+1}) \\ &\leq f(x^*) - f(x_k) + \alpha D(x^*; x_k) - \alpha D(x^*; x_{k+1}) \end{aligned}$$

where the final inequality uses the subgradient inequality for the convex function f . By completing the square on the left-hand-side we obtain

$$\left(\frac{H}{2\sqrt{\alpha/2}} - \sqrt{\alpha/2}\|x_{k+1} - x_k\|_1 \right)^2 - \frac{H^2}{2\alpha} \leq f(x^*) - f(x_k) + \alpha D(x^*; x_k) - \alpha D(x^*; x_{k+1})$$

Rearranging terms and neglecting the nonnegative squared term gives

$$f(x_k) \leq f(x^*) + \frac{H^2}{2\alpha} + \alpha D(x^*; x_k) - \alpha D(x^*; x_{k+1})$$

This holds for all $k \in \{0, 1, 2, \dots\}$. Fix m as a positive integer. Summing over $k \in \{0, \dots, m-1\}$ gives

$$\begin{aligned} \sum_{k=0}^{m-1} f(x_k) &\leq mf(x^*) + \frac{mH^2}{2\alpha} + \alpha D(x^*; x_0) - \alpha D(x^*; x_m) \\ &\leq mf(x^*) + \frac{mH^2}{2\alpha} + \alpha \log(n) \end{aligned}$$

where the final inequality uses the fact $x_0 = (1/n, 1/n, \dots, 1/n)$ with (29) together with the fact that $D(x^*; x_m) \geq 0$. Dividing by m gives the result. $\square$

To compare performance with standard projected subgradient descent over the probability simplex P , we have

- Standard subgradient descent with fixed stepsize $\delta > 0$:

$$\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \frac{\delta B^2}{2} + \frac{2}{m\delta}$$

which uses the fact that for $x_0, x^* \in P$ we have $\|x_0 - x^*\| \leq \|x_0\| + \|x^*\| \leq 2$. With $m = 1/\delta^2$ (assuming this is an integer for simplicity) we have an error bound

$$\boxed{\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \left(\frac{B^2}{2} + 2 \right) \delta}$$

- Mirror descent with $\alpha = 1/\delta$:

$$\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \frac{\delta H^2}{2} + \frac{\log(n)}{m\delta}$$

With $m = 1/\delta^2$ we have an error bound

$$\boxed{\frac{1}{m} \sum_{k=0}^{m-1} f(x_k) \leq f(x^*) + \left(\frac{H^2}{2} + \log(n) \right) \delta}$$

We want to compare the coefficient $B^2/2 + 2$ with the coefficient $H^2/2 + \log(n)$. While $\log(n) \geq 2$ whenever $n \geq 8$, in many cases $B^2/2$ is much larger than $H^2/2$. Thus, the savings from mirror descent is in the *first term* that compares $B^2/2$ with $H^2/2$. Note that B is a bound on $\|f'(x)\|_2$ over all $x \in P$, while H is a bound on $\|f'(x)\|_\infty$ over all $x \in P$, where the infinity norm of $f'(x)$ is the largest magnitude of any component of $f'(x)$. For example, if $f(x) = \sum_{i=1}^n x_i$ then $\|f'(x)\|_2 = \sqrt{n}$ while $\|f'(x)\|_\infty = 1$. In general, if we know that H is a bound on the magnitude of any component of $f'(x)$, we can define $B = H\sqrt{n}$. This means the term $B^2/2$ can be a factor of n worse than the corresponding term $H^2/2$. Roughly speaking, we often view H^2 as being a value that is independent of the dimension n , while B^2 grows linearly with n . So the coefficient of the error bound for standard subgradient descent is proportional to n , while the coefficient for mirror descent is proportional to $\log(n)$.

VI. ONLINE CONVEX OPTIMIZATION

This section extends the subgradient descent technique to explore a class of time-varying problems called *online convex optimization* problems. These problems, and the performance bounds of the resulting algorithms, are developed by Zinkevich [3]. Let $C \subseteq \mathbb{R}^n$ be a nonempty closed and convex set. Every time step $k \in \{0, 1, 2, \dots\}$ we choose a vector $x_k \in C$ and this incurs a cost $f_k(x_k)$, where $f_k : C \rightarrow \mathbb{R}$ is a convex (possibly nonsmooth) function that satisfies $\|f'(x_k)\| \leq B$ for some $B \geq 0$. For some positive integer m , our goal is to minimize the time average cost

$$\frac{1}{m} \sum_{k=0}^{m-1} f_k(x_k)$$

If each function f_k were known before the decision $x_k \in C$ was made, the problem would be trivial: At each step k , just observe the new function f_k and choose $x_k \in C$ as the minimizer of f_k over all $x \in C$. The problem is challenging because we assume the function f_k is *unknown* at the start of step k . The function f_k is only revealed at the *end* of step k , which is *after* the decision $x_k \in C$ is made. Further, we assume no probability model. That is, the f_k functions can be any arbitrary functions that vary over k , including functions that change dramatically from one time step to the next. The only assumption is that these functions are convex and their subgradients are always bounded in magnitude by B .

Since the decision $x_k \in C$ is made without knowing f_k , it is impossible to develop an online algorithm that yields time average cost as good as

$$\text{cost}_1(m) = \frac{1}{m} \sum_{k=0}^{m-1} \inf_{x \in C} \{f_k(x)\}$$

which is a cost that could only be achieved by choosing x_k as the minimizer of f_k at each step $k \in \{0, \dots, m-1\}$. However, Zinkevich shows in [3] that we can develop an algorithm that comes within a small *regret* of the best *fixed decision in hindsight*. Specifically, as a target cost, we can consider a time-traveler who knows the future and is able to choose a vector $x \in C$, and then use this as a *fixed decision* $x_k = x$ for all $k \in \{0, \dots, m-1\}$ to minimize $\frac{1}{m} \sum_{k=0}^{m-1} f_k(x)$. The time-traveler is able to use knowledge of the future functions f_k to choose the best $x \in C$ before time 0. This defines a new target cost:

$$\text{cost}_2(m) = \inf_{x \in C} \left\{ \frac{1}{m} \sum_{k=0}^{m-1} f_k(x) \right\}$$

Clearly $\text{cost}_1(m) \leq \text{cost}_2(m)$. However, $\text{cost}_2(m)$ is still a meaningful and nontrivial target. Remarkably, an algorithm that *does not know the future*, but allows choosing different x_k vectors every step, can perform almost as well as the best fixed-decision algorithm (which chooses $x_k = x$ for all k), where the fixed decision x is optimized with full knowledge of the future.

The algorithm is simple: Fix $\delta > 0$ as a constant stepsize. Then

- For $k = 0$, choose $x_0 \in C$ arbitrarily.
- For $k \in \{1, 2, 3, \dots\}$ do
 - 1) Observe $f'_{k-1}(x_{k-1})$ (the subgradient of the *previous* function at the point x_{k-1} chosen in the *previous* step).
 - 2) Choose x_k on the current step k by

$$x_k = \pi_C(x_{k-1} - \delta f'_{k-1}(x_{k-1}))$$

Theorem 12: Let $C \subseteq \mathbb{R}^n$ be a nonempty, convex, and compact set. Assume each convex function $f_k : C \rightarrow \mathbb{R}$ has B -bounded subgradients for some $B \geq 0$ (and our oracle provides such subgradients). Assume there is a value D such that $\|x_0 - x\| \leq D$ for all $x \in C$. Under the above online algorithm, we have for all $z \in C$ and all positive integers m

$$\frac{1}{m} \sum_{k=0}^{m-1} f_k(x_k) \leq \inf_{x \in C} \left\{ \frac{1}{m} \sum_{k=0}^{m-1} f_k(x) \right\} + \frac{\delta B^2}{2} + \frac{D^2}{2m\delta}$$

and so the error term $\left(\frac{\delta B^2}{2} + \frac{D^2}{2m\delta} \right)$ is $O(\delta)$ whenever $m \geq 1/\delta^2$.

Proof: Fix $x \in C$ and note that $x = \pi_C(x)$. We have for each $k \in \{1, 2, 3, \dots\}$

$$\begin{aligned}
\|x_k - x\|^2 &= \|\pi_C(x_{k-1} - \delta f'_{k-1}(x_{k-1})) - x\|^2 \\
&\stackrel{(a)}{=} \|\pi_C(x_{k-1} - \delta f'_{k-1}(x_{k-1})) - \pi_C(x)\|^2 \\
&\stackrel{(b)}{\leq} \|x_{k-1} - \delta f'_{k-1}(x_{k-1}) - x\|^2 \\
&= \|x_{k-1} - x\|^2 + \delta^2 \|f'_{k-1}(x_{k-1})\|^2 + 2\delta f'_{k-1}(x_{k-1})^\top (x - x_{k-1}) \\
&\stackrel{(c)}{\leq} \|x_{k-1} - x\|^2 + \delta^2 \|f'_{k-1}(x_{k-1})\|^2 + 2\delta(f_{k-1}(x) - f_{k-1}(x_{k-1})) \\
&\leq \|x_{k-1} - x\|^2 + \delta^2 B^2 + 2\delta(f_{k-1}(x) - f_{k-1}(x_{k-1}))
\end{aligned}$$

where (a) holds by $x = \pi_C(x)$; (b) holds because projections do not increase distance; (c) holds by the subgradient inequality:

$$f_{k-1}(x) \geq f_{k-1}(x_{k-1}) + f'_{k-1}(x_{k-1})^\top (x - x_{k-1})$$

Rearranging terms gives

$$f_{k-1}(x_{k-1}) \leq f_{k-1}(x) + \frac{\delta B^2}{2} + \frac{\|x_{k-1} - x\|^2 - \|x_k - x\|^2}{2\delta}$$

Summing over $k \in \{1, \dots, m\}$ and dividing by m gives

$$\begin{aligned}
\frac{1}{m} \sum_{k=0}^{m-1} f_k(x_k) &\leq \frac{1}{m} \sum_{k=0}^{m-1} f_k(x) + \frac{\delta B^2}{2} + \frac{\|x_0 - x\|^2 - \|x_m - x\|^2}{2m\delta} \\
&\leq \frac{1}{m} \sum_{k=0}^{m-1} f_k(x) + \frac{\delta B^2}{2} + \frac{D^2}{2m\delta}
\end{aligned}$$

This holds for all $x \in C$, so it also holds if we take the inf over all $x \in C$. $\square$

The fixed-stepsizes method for online convex optimization can be extended to vanishing stepsize $\delta_k = 1/\sqrt{k+1}$ using the same technique as before. For online convex optimization with no additional structure on the convex f_k functions, it can be shown that the $O(1/\delta^2)$ convergence time is generally the best possible (this holds even if the f_k functions are linear) [11]. However, a better asymptotic growth with δ is possible if the f_k functions are strongly convex [11][12]. For problems that optimize over a probability simplex, an improved *coefficient* that multiplies the $1/\delta^2$ is possible using a mirror descent version of online convex optimization.

A. Online portfolio optimization

The following problem of online portfolio optimization is formulated in [13] (where a different approach is developed). Let $\theta[k]$ for $k \in \{0, 1, 2, \dots\}$ be a vector of stock prices for n different stocks:⁴

$$\theta[k] = (\theta_1[k], \dots, \theta_n[k])$$

The prices can vary arbitrarily over time, but are assumed to always be strictly positive. Also, there are worst-case growth ratios $q_{min} > 0, q_{max} > 0$ such that

$$q_{min} \leq \frac{\theta_i[k+1]}{\theta_i[k]} \leq q_{max} \quad \forall i \in \{1, \dots, n\}, k \in \{0, 1, 2, \dots\} \quad (35)$$

Without loss of generality we assume $\theta_n[k] = 1$ for all $k \in \{0, 1, 2, \dots\}$. Specifically, stock investment n is a savings account: Putting money into the savings account keeps it there for the next time step without increasing or decreasing its value. The goal is to maximize profit by buying and selling over time. For simplicity (and to maintain a problem structure that fits the online convex optimization framework) we ignore transaction fees.

We initially have money $v[0] > 0$. All of it is in our savings account (account n). We invest a fraction $x_i[0]$ of this amount $v[0]$ into each stock $i \in \{1, \dots, n\}$. Thus, we invest $v[0]x_i[0]$ dollars in each stock $i \in \{1, \dots, n\}$. The vector $x[0] = (x_1[0], \dots, x_n[0])$ is the *decision vector* and is chosen as any vector in the probability simplex P . Let $s_i[1]$ be the amount of shares of stock i we buy during time 0 (which will be the number of shares we have at the start of time step 1):

$$s_i[1] = \frac{v[0]x_i[0]}{\theta_i[0]} \quad \forall i \in \{1, \dots, n\}$$

⁴It is convenient to use notation $\theta[k]$, rather than θ_k , so individual components $\theta_i[k]$ can be more easily specified.

where we allow $s_i[1]$ to be noninteger for simplicity. At the next time step, the valuation of our portfolio depends on the new vector of stock prices $\theta[1]$:

$$v[1] = \sum_{i=1}^n s_i[1]\theta_i[1]$$

and we hope that $v[1] \geq v[0]$.

In this way, define for each time step $k \in \{0, 1, 2, \dots\}$

$$\begin{aligned} s[k] &= (s_1[k], \dots, s_n[k]) = \text{stock shares owned at start of step } k \\ v[k] &= \sum_{i=1}^n s_i[k]\theta_i[k] = \text{value of portfolio at start of step } k \\ x[k] &= (x_1[k], \dots, x_n[k]) = \text{decision vector for step } k \end{aligned}$$

where $x[k]$ is chosen to satisfy $x[k] \in P$. The vector $x[k]$ is chosen without knowledge of the future stock price $\theta[k+1]$. We define $s[0] = (0, 0, \dots, 0, v[0])$, so initially all money is in the savings account (since $\theta_n[k] = 1$ for all k , one share of the savings account corresponds to one dollar in that account). At the start of every step $k \in \{0, 1, 2, \dots\}$, we liquidate the entire value $v[k]$ of our portfolio and then reinvest according to the allocation vector $x[k]$, so we invest $v[k]x_i[k]$ dollars into stock i , buying shares $s_i[k+1]$ so that

$$s_i[k+1] = \frac{v[k]x_i[k]}{\theta_i[k]} \quad \forall i \in \{1, \dots, n\} \quad (36)$$

The value $v[k+1]$ of our portfolio can be written in terms of $v[k]$ as

$$\begin{aligned} v[k+1] &= \sum_{i=1}^n s_i[k+1]\theta_i[k+1] \\ &= v[k] \sum_{i=1}^n x_i[k] \frac{\theta_i[k+1]}{\theta_i[k]} \end{aligned}$$

where the final equality substitutes the value of $s_i[k+1]$ given in (36).

Define $q_i[k]$ as the ratio of stock prices between time $k+1$ and k for stock i

$$q_i[k] = \frac{\theta_i[k+1]}{\theta_i[k]} \quad \forall i \in \{1, \dots, n\}$$

The values $q_i[k]$ are positive real numbers. From the assumption (35) we know

$$q_{min} \leq q_i[k] \leq q_{max} \quad \forall i, k$$

Of course, the values $q_i[k]$ are unknown at time step k because the future stock prices $\theta[k+1]$ are unknown. With this definition we have

$$v[k+1] = v[k] \sum_{i=1}^n x_i[k] q_i[k] \quad \forall k \in \{0, 1, 2, \dots\}$$

Taking the log of both sides gives

$$\log(v[k+1]) = \log(v[k]) + \log\left(\sum_{i=1}^n x_i[k] q_i[k]\right) \quad \forall k \in \{0, 1, 2, \dots\}$$

Summing this over $k \in \{0, \dots, m-1\}$ for some positive integer m gives

$$\log\left(\frac{v[m]}{v[0]}\right) = \sum_{k=0}^{m-1} \log\left(\sum_{i=1}^n x_i[k] q_i[k]\right) \quad \forall m \in \{1, 2, 3, \dots\}$$

The left-hand-side is the logarithm of the growth ratio of our portfolio over m steps. We want to choose $x[k]$ over time to make this value large. By dividing by m we obtain what can be viewed as an averaged growth exponent per time step:

$$\frac{1}{m} \log\left(\frac{v[m]}{v[0]}\right) = \frac{1}{m} \sum_{k=0}^{m-1} \log\left(\sum_{i=1}^n x_i[k] q_i[k]\right) \quad \forall m \in \{1, 2, 3, \dots\}$$

If the averaged growth exponent is zero then we have neither increased nor decreased the value of our portfolio. If the averaged growth exponent is positive then we have increased our wealth. If the averaged growth exponent is negative then we have decreased our wealth. In particular:

$$\frac{1}{m} \log\left(\frac{v[m]}{v[0]}\right) = \beta \implies e^{m\beta} = \frac{v[m]}{v[0]}$$

For each $k \in \{0, 1, 2, \dots\}$ define the concave function $g_k : P \rightarrow \mathbb{R}$ by

$$g_k(x) = \log \left(\sum_{i=1}^n x_i q_i[k] \right) \quad \forall x \in P$$

Thus

$$\frac{1}{m} \log \left(\frac{v[m]}{v[0]} \right) = \frac{1}{m} \sum_{k=0}^{m-1} g_k(x[k]) \quad \forall m \in \{1, 2, 3, \dots\}$$

Making the above expression large is the same as making -1 times that expression small. In particular, we can define $f_k(x) = -g_k(x)$ and we want to minimize the sum of convex functions f_k . We see that the problem of *maximizing our portfolio growth ratio is exactly an online convex optimization problem*: Every step k we choose $x[k] \in P$ with the goal of maximizing the unknown concave function $g_k(x)$. The function $g_k(x)$ is revealed to us at the end of the step (after we make the $x[k]$ decision) when the new stock prices $\theta[k+1]$ are revealed.

The concave functions g_k are differentiable and so

$$g'_k(x) = \nabla g_k(x) = \frac{q[k]}{\sum_{i=1}^n x_i q_i[k]} \quad (37)$$

where

$$q[k] = (q_1[k], q_2[k], \dots, q_n[k])$$

There are B -bounded gradients (for some $B > 0$) because of the assumption (35) (see also Exercise 13). The online portfolio algorithm is as follows: Fix $\delta > 0$. Then

- Choose $x[0] \in P$ arbitrarily.
- For each $k \in \{1, 2, 3, \dots\}$ do
 - 1) Observe $q_i[k-1] = \theta_i[k]/\theta_i[k-1]$ for each $i \in \{1, \dots, n\}$.
 - 2) Choose $x[k] \in P$ by

$$x[k] = \pi_P(x[k-1] - \delta g'_{k-1}(x[k-1]))$$

where the $g'_{k-1}(\cdot)$ functions are defined in (37), specifically

$$g'_{k-1}(x[k-1]) = \frac{q[k-1]}{\sum_{i=1}^n x_i[k-1] q_i[k-1]}$$

Theorem 13: Assume there are positive real numbers q_{min}, q_{max} such that (35) holds for all $i \in \{1, \dots, n\}$ and $k \in \{0, 1, 2, \dots\}$. Assume $v[0] > 0$. The above dynamic portfolio optimization algorithm ensures for each positive integer m :

$$\frac{1}{m} \sum_{k=0}^{m-1} g_k(x[k]) \geq \sup_{x \in C} \left\{ \frac{1}{m} \sum_{k=0}^{m-1} g_k(x) \right\} - \frac{\delta B^2}{2} - \frac{r^2}{2m\delta}$$

where $\sum_{k=0}^{m-1} g_k(x[k]) = \log \left(\frac{v[m]}{v[0]} \right)$, $r = 2$, and

$$B = \sup_{x \in P, k \in \{0, 1, 2, \dots\}} \left\| \frac{q[k]}{\sum_{i=1}^n x_i q_i[k]} \right\|$$

where B can be upper bounded independent of k using the q_{min}, q_{max} values (see Exercise 13). It follows that, without knowing the future, we can achieve a growth exponent that comes within $O(\delta)$ of the best *fixed-decision exponent in hindsight* after a number of steps $m \geq 1/\delta^2$. Further, since $\theta_n[k] = 1$ for all $k \in \{0, 1, 2, \dots\}$ we have

$$\sup_{x \in C} \left\{ \frac{1}{m} \sum_{k=0}^{m-1} g_k(x) \right\} \geq 0$$

so the best fixed-decision exponent is always greater than or equal to zero.

Proof: This is a direct application of Theorem 12 and the fact that for all $x[0], x \in P$ we have

$$\|x[0] - x\| \leq \|x[0]\| + \|x\| \leq 2$$

and

$$\sup_{x \in P, k \in \{0, 1, 2, \dots\}} \left\{ \frac{q[k]}{\sum_{i=1}^n x_i q_i[k]} \right\} \geq \sum_{k=0}^{m-1} g_k(0, 0, \dots, 0, 1) = \sum_{k=0}^{m-1} \log(1) = 0$$

□

B. Fast online portfolio optimization via mirror descent

Since this online portfolio problem optimizes over a probability simplex P , the mirror descent method is important for improving convergence time (see Exercise 15). See also [11] for alternative algorithms.

VII. STRONG CONVEXITY

Throughout this section we use $\|\cdot\|$ to denote the Euclidean norm on $\mathbb{R}^n$:

$$\|x\| = \sqrt{\sum_{i=1}^n x_i^2} \quad \forall x \in \mathbb{R}^n$$

Let $X \subseteq \mathbb{R}^n$ be a nonempty set.

A. Basic definitions

Definition 7: A function $f : X \rightarrow \mathbb{R}$ is *strictly convex* if

- 1) X is a convex set.
- 2) For all $x, y \in X$ with $x \neq y$, and all $p \in (0, 1)$, we have

$$f(px + (1-p)y) < pf(x) + (1-p)f(y)$$

Definition 8: Fix $c \geq 0$. A function $f : X \rightarrow \mathbb{R}$ is *c-strongly convex* if

- 1) X is a convex set.
- 2) For all $x, y \in X$ and all $p \in [0, 1]$ we have

$$f(px + (1-p)y) \leq pf(x) + (1-p)f(y) - \frac{c}{2}p(1-p)\|x - y\|^2$$

Definition 9: Fix $c \geq 0$. A function $f : X \rightarrow \mathbb{R}$ is *c-strongly concave* if $-f$ is *c-strongly convex*.

Definition 10: A function $f : X \rightarrow \mathbb{R}$ is *strongly convex* if it is *c-strongly convex* for some $c > 0$. The function is *strongly concave* if it is *c-strongly concave* for some $c > 0$.

It can be seen that every strongly convex function is also strictly convex, and every strictly convex function is convex. A function is convex if and only if it is 0-strongly convex.⁵ It can be shown that if a function $f : X \rightarrow \mathbb{R}$ is strictly convex and has a minimizer $x^* \in X$, then the minimizer is unique (see Exercise 19).

B. Motivation from quadratic functions

Let $\|x\| = \sqrt{x^\top x}$ denote the Euclidean norm. It can be shown that for all $x, y \in \mathbb{R}^n$ and all $p \in \mathbb{R}$ we have

$$\|px + (1-p)y\|^2 = p\|x\|^2 + (1-p)\|y\|^2 - p(1-p)\|x - y\|^2 \quad (38)$$

The equality (38) is related to the *parallelogram law* and holds more generally for any inner product norm.⁶ It means that $\|x\|^2$ is a 2-strongly convex function of $x \in \mathbb{R}^n$.

More generally, fix $c \geq 0$ and $a \in \mathbb{R}^n$. Consider the quadratic function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by

$$f(x) = \frac{c}{2}\|x - a\|^2 \quad \forall x \in \mathbb{R}^n$$

The following equality holds for all $x, y \in \mathbb{R}^n$ and all $p \in \mathbb{R}$:

$$f(px + (1-p)y) = pf(x) + (1-p)f(y) - \frac{c}{2}p(1-p)\|x - y\|^2$$

It can be seen that f is *c-strongly convex*.

⁵Since strongly convex functions must have $c > 0$, it may be confusing to treat the case $c = 0$ and to use terminology “0-strongly convex.” Nevertheless, this is a standard. The value c is the *convexity strength parameter*, and so one can view the case $c = 0$ as having “no strength,” meaning a 0-strongly convex function is not strongly convex at all.

⁶A norm $\|\cdot\|$ on $\mathbb{R}^n$ is an *inner product norm* if $\|x\| = \sqrt{x^\top Ax}$ for all $x \in \mathbb{R}^n$, where A is some real $n \times n$ symmetric, positive definite matrix. It can be shown that a norm on $\mathbb{R}^n$ is an inner product norm if and only if it satisfies the *parallelogram law*: $\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2$.

C. Conditions that imply strong convexity

Lemma 10: Fix $c \geq 0$. A function $f : X \rightarrow \mathbb{R}$ is c -strongly convex if and only if the function $h : X \rightarrow \mathbb{R}$ defined by

$$h(x) = f(x) - \frac{c}{2} \|x\|^2 \quad \forall x \in X$$

is a convex function.

Proof: See Exercises 24 and 25. □

Lemma 11: (Second derivative test) Suppose $f : \mathbb{R} \rightarrow \mathbb{R}$ is twice differentiable. Fix $c \geq 0$. Then f is c -strongly convex if and only if

$$f''(x) \geq c \quad \forall x \in \mathbb{R}$$

Similarly, if f is twice differentiable over an interval $[a, b]$, then it is c -strongly convex (treated as a function with domain $[a, b]$) if and only if $f''(x) \geq c$ for all $x \in [a, b]$.

Proof: See Exercise 23. □

Lemma 12: (Hessian condition) Let $X \subseteq \mathbb{R}^n$ be a convex set. Let $f : X \rightarrow \mathbb{R}$ be a twice differentiable function. Fix $c \geq 0$. The function f is c -strongly convex if and only if for all $x \in X$ the Hessian matrix $\nabla^2 f$ is symmetric positive semidefinite and every eigenvalue λ of $\nabla^2 f$ is greater than or equal to c .

Proof: Omitted for brevity. □

Lemma 13: (Weighted sums) Let $X \subseteq \mathbb{R}^n$ be a convex set. Let $f : X \rightarrow \mathbb{R}$ and $g : X \rightarrow \mathbb{R}$ be functions. Fix $c \geq 0, d \geq 0$. If f is c -strongly convex and g is d -strongly convex, then for any weights $w_1, w_2 \geq 0$, the function $h : X \rightarrow \mathbb{R}$ defined by

$$h(x) = w_1 f(x) + w_2 g(x) \quad \forall x \in X$$

is $(w_1 c + w_2 d)$ -strongly convex.

Proof: See Exercise 17. □

Lemma 14: (Building strongly convex functions) Let $\|\cdot\|$ denote the Euclidean norm. Let $f : X \rightarrow \mathbb{R}$ be any convex function. Then for any $c \geq 0$ and any $w \in \mathbb{R}^n$, the function $h : X \rightarrow \mathbb{R}$ defined by

$$h(x) = f(x) + \frac{c}{2} \|x - w\|^2 \quad \forall x \in \mathbb{R}^n$$

is c -strongly convex.

Proof: See Exercise 18. □

D. Pathological examples

Given any $c \geq 0$, Lemma 14 makes it easy to build a c -strongly convex function $g : \mathbb{R} \rightarrow \mathbb{R}$ that is *not differentiable*:

$$g(x) = |x| + \frac{c}{2} x^2 \quad \forall x \in \mathbb{R}$$

Lemma 14 also makes it easy to build a c -strongly convex function $h : [0, 1] \rightarrow \mathbb{R}$ that is *not continuous*:

$$h(x) = f(x) + \frac{c}{2} x^2 \quad \forall x \in [0, 1]$$

where f is the discontinuous convex function in (1).

E. Pushback

Theorem 14: (Quadratic pushback) Let $X \subseteq \mathbb{R}^n$ be a convex set. Fix $c \geq 0$ and suppose $f : X \rightarrow \mathbb{R}$ is c -strongly convex. Let $x^* \in X$ be a minimizer of f over all $x \in X$. Then

$$f(x^*) \leq f(x) - \frac{c}{2} \|x - x^*\|^2 \quad \forall x \in X$$

Proof: Fix $x \in X$. Fix $p \in (0, 1)$. Since X is a convex set and $x \in X$ and $x^* \in X$ we have

$$px + (1 - p)x^* \in X$$

Since x^* minimizes f over all other vectors in the domain X , we have

$$\begin{aligned} f(x^*) &\leq f(px + (1 - p)x^*) \\ &\leq pf(x) + (1 - p)f(x^*) - \frac{cp(1 - p)}{2} \|x - x^*\|^2 \end{aligned}$$

where the final inequality uses the definition of f being c -strongly convex. Subtracting $(1-p)f(x^*)$ from both sides of the above inequality gives

$$pf(x^*) \leq pf(x) - \frac{c}{2}p(1-p)\|x - x^*\|^2$$

Dividing both sides by p (recall $p > 0$) gives

$$f(x^*) \leq f(x) - \frac{c}{2}(1-p)\|x - x^*\|^2$$

This holds for all $p \in (0, 1)$. Taking a limit of both sides of the above inequality as $p \rightarrow 0$ gives

$$f(x^*) \leq f(x) - \frac{c}{2}\|x - x^*\|^2$$

□

To understand the strength of Theorem 14, recall that for any (possibly nonconvex) function $f : X \rightarrow \mathbb{R}$, if $x^* \in X$ minimizes f over all vectors in the domain then (by definition of a minimizer):

$$f(x^*) \leq f(x) \quad \forall x \in X$$

Theorem 14 shows that this key inequality can be strengthened when f is c -strongly convex by subtracting a nonnegative *pushback term* $\frac{c}{2}\|x - x^*\|^2$ from the right-hand-side. It means that the function value $f(x)$ grows at least quadratically in the distance between x and x^* . Any mathematical analysis that uses the key inequality $f(x^*) \leq f(x)$ for a minimizer x^* can be strengthened when the f function is c -strongly convex. The next theorem shows that the standard subgradient inequality for convex functions can be strengthened for *strongly* convex functions.

Theorem 15: (Strengthened subgradient inequality) Fix $c \geq 0$. Let $f : X \rightarrow \mathbb{R}$ be a c -strongly convex function. Fix $x \in X$ and suppose $\partial f(x)$ is nonempty. Then for each $f'(x) \in \partial f(x)$ we have

$$f(y) \geq f(x) + f'(x)^\top(y - x) + \frac{c}{2}\|y - x\|^2 \quad \forall y \in X$$

Proof: Fix $x \in X$ and fix $f'(x) \in \partial f(x)$. Define the function $h : X \rightarrow \mathbb{R}$ by

$$h(y) = f(y) - f(x) - f'(x)^\top(y - x) \quad \forall y \in X$$

This function of y sums the c -strongly convex function $f(y)$ and the affine (hence convex) function $-f(x) - f'(x)^\top(y - x)$, so it is c -strongly convex. The standard subgradient inequality for f states

$$f(y) \geq f(x) + f'(x)^\top(y - x) \quad \forall y \in X$$

which means that $h(y) \geq 0$ for all $y \in X$. Further, it is clear that $h(x) = 0$, so x is a minimizer of the c -strongly convex function h . Theorem 14 implies

$$0 = h(x) \leq h(y) - \frac{c}{2}\|y - x\|^2 \quad \forall y \in X$$

Substituting the definition of $h(y)$ gives

$$0 \leq f(y) - f(x) - f'(x)^\top(y - x) - \frac{c}{2}\|y - x\|^2 \quad \forall y \in X$$

which proves the result. □

F. A note on general norms

The definition of c -strongly convex can be extended to treat general norms. If $\|\cdot\|$ is a general norm on $\mathbb{R}^n$ (possibly different from the Euclidean norm), we define f to be *c-strongly convex with respect to norm $\|\cdot\|$* if it satisfies Definition 8 where the norm in that definition uses the general norm of interest (rather than the Euclidean norm). Theorem 14 (the pushback theorem) holds verbatim with respect to any general norm, with no change in the proof. The only change is that the resulting theorem can no longer be called the “Quadratic pushback” theorem because the pushback term $\frac{c}{2}\|y - x\|^2$ now uses a general norm that does not necessarily have a quadratic structure.

It is known that for any two norms $\|x\|_{Alice}$ and $\|x\|_{Bob}$ on $\mathbb{R}^n$, there are positive numbers a, b such that

$$a\|x\|_{Alice} \leq \|x\|_{Bob} \leq b\|x\|_{Alice} \quad \forall x \in \mathbb{R}^n$$

This well known result is called *equivalence of norms*. From this, it holds that if there is a $c > 0$ such that f is c -strongly convex with respect to $\|\cdot\|_{Alice}$, then there is a $d > 0$ such that f is d -strongly convex with respect to $\|\cdot\|_{Bob}$.

When using a general norm, one should be careful about building strongly convex functions: While Lemma 14 does extend more generally to any *inner product norm*, it does not necessarily hold if the norm is not an inner product norm. For example, for $n \geq 2$, the L_∞ norm $\|x\|_\infty$ is not an inner product norm, and the function $\|x\|_\infty^2$ defined for $x \in \mathbb{R}^n$ is not even strictly convex, so it is not strongly convex with respect to any norm (see Exercise 1).

G. A note on lower bounds and unique minimizers over closed sets

It can be shown that every nonempty convex set $X \subseteq \mathbb{R}^n$ has a nonempty relative interior. It can also be shown that every convex function $f : X \rightarrow \mathbb{R}$ has at least one subgradient at every point in its relative interior. Thus, if $f : X \rightarrow \mathbb{R}$ is a convex function, there must be some point $a \in X$ for which $\partial f(x)$ is nonempty. This point a acts as an anchor to establish a linear lower bound on f :

$$f(x) \geq f(a) + f'(a)^\top (x - a) \quad \forall x \in X$$

When f is c -strongly convex for some $c > 0$, Theorem 15 ensures the subgradient inequality can be strengthened to establish a quadratic lower bound on f :

$$f(x) \geq f(a) + f'(a)^\top (x - a) + \frac{c}{2} \|x - a\|^2 \quad \forall x \in X \quad (39)$$

Lemma 15: Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. If $f : X \rightarrow \mathbb{R}$ is strongly convex then it is lower bounded, meaning there is a finite $L \in \mathbb{R}$ such that

$$f(x) \geq L \quad \forall x \in X$$

Proof: If $f : X \rightarrow \mathbb{R}$ is strongly convex then it is c -strongly convex for some $c > 0$, and there is an $a \in X$ such that (39) holds. The quadratic function $f(a) + f'(a)^\top (x - a) + \frac{c}{2} \|x - a\|^2$ is lower bounded over all $x \in \mathbb{R}^n$, so it is certainly lower bounded over all $x \in X$. $\square$

Lemma 16: Let $X \subseteq \mathbb{R}^n$ be a nonempty, closed, and convex set. If $f : X \rightarrow \mathbb{R}$ is continuous and strongly convex, then f has a unique minimizer $x^* \in X$.

Proof: Suppose X is a closed convex set and $f : X \rightarrow \mathbb{R}$ is continuous and c -strongly convex for some $c > 0$. Then there is an $a \in X$ such that (39) holds. Define

$$f^* = \inf_{x \in X} \{f(x)\}$$

The value f^* is finite because f is lower bounded. Define

$$C = \{x \in X : f(x) \leq f^* + 1\}$$

$$D = \{x \in \mathbb{R}^n : f(a) + f'(a)^\top (x - a) + \frac{c}{2} \|x - a\|^2 \leq f^* + 1\}$$

Set C excludes those points $x \in X$ with $f(x)$ values far away from f^* , so

$$f^* = \inf_{x \in C} \{f(x)\}$$

Since $C \subseteq X \cap D$, we shall restrict our search for minimizers of f over all $x \in X \cap D$. The set X is closed and the set D is compact, so $X \cap D$ is compact. The extreme value theorem ensures the continuous function f has a minimum value over all x in the compact set $X \cap D$. Let $x^* \in X \cap D$ be a point at which this minimum value occurs. Since $C \subseteq X \cap D \subseteq X$ we have

$$f^* = \inf_{x \in C} \{f(x)\} \geq \underbrace{\inf_{x \in X \cap D} \{f(x)\}}_{f(x^*)} \geq \inf_{x \in X} \{f(x)\} = f^*$$

so $f(x^*) = f^*$, meaning x^* is a minimizer of f . Since f is strongly convex, the minimizer is unique. $\square$

VIII. FASTER CONVERGENCE FOR SMOOTH AND STRONGLY CONVEX CASES

The general (nonsmooth) analysis of projected subgradient descent requires the stepsize δ to be made smaller and smaller if more and more accuracy is desired. However, if the algorithm is run when the function f is smooth, an interesting behavior emerges: If a fixed stepsize is *sufficiently small*, the algorithm runs and iterations continue to get closer and closer to optimality, converging to optimality in the limit, *even without changing the stepsize*. It suffices to choose a stepsize δ that is small enough relative to a *smoothness parameter* β . For smooth convex functions, this section shows the optimality gap vanishes like $O(1/m)$ as the number of iterations m increases. Further, for smooth and strongly convex functions, the optimality gap vanishes even faster, like $O(e^{-cm})$, for some constant $c > 0$. Another advantage of this smooth regime is that the bounded subgradient assumption is not needed. Indeed, for smooth convex functions we use gradients $\nabla f(x)$ that are allowed to have unbounded magnitudes over $x \in \text{dom}(f)$. The algorithm considered in this section can be called *projected gradient descent* because the subgradient $f'(x)$ that is used is in fact a gradient $\nabla f(x)$.

A. β -smooth functions

Definition 11: A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is β -smooth if

- 1) The gradient $\nabla f(x)$ exists for all $x \in \mathbb{R}^n$.
- 2) The gradient function $\nabla f(x)$ is β -Lipschitz continuous, meaning that

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\| \quad \forall x, y \in \mathbb{R}^n \quad (40)$$

where the standard Euclidean norm (L_2 norm) is used.

Definition 12: Let $X \subseteq \mathbb{R}^n$ be a nonempty set. A function $f : X \rightarrow \mathbb{R}$ is β -smooth if it can be extended to a β -smooth function over all of $\mathbb{R}^n$. That is, there exists a β -smooth function $\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ that satisfies $\tilde{f}(x) = f(x)$ for all $x \in X$. For such a function f , we define $\nabla f(x) = \nabla \tilde{f}(x)$ for all $x \in X$. In particular, while, strictly speaking, the function f is only defined to have a gradient at an interior point of its domain, we can define its gradient at boundary points in terms of the gradient of its extension.⁷

Lemma 17: (Descent lemma [14]) Let $X \subseteq \mathbb{R}^n$ be a convex nonempty set. Let $f : X \rightarrow \mathbb{R}$ be a β -smooth function (possibly a nonconvex function). Then for all $x, y \in X$ we have

$$|f(y) - f(x) - \nabla f(x)^\top (y - x)| \leq \frac{\beta}{2} \|y - x\|^2$$

where we use the Euclidean norm. In particular, the function deviates from its linear expansion about a point x by at most a quadratic function of the distance between y and x . Equivalently

$$-\frac{\beta}{2} \|y - x\|^2 \leq f(y) - f(x) - \nabla f(x)^\top (y - x) \leq \frac{\beta}{2} \|y - x\|^2 \quad (41)$$

Proof: For simplicity let $\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ denote the extended function that is β -smooth over all $\mathbb{R}^n$, so $\nabla \tilde{f}(x)$ exists and is β -Lipschitz continuous (this notation is easier to use than $\tilde{f}$). Following the proof given in [14], define $h : \mathbb{R} \rightarrow \mathbb{R}$ by

$$h(t) = f(x + t(y - x)) \quad \forall t \in \mathbb{R}$$

Then h has the continuous derivative

$$h'(t) = \nabla f(x + t(y - x))^\top (y - x) \quad \forall t \in \mathbb{R}$$

The fundamental theorem of calculus implies

$$\begin{aligned} h(1) - h(0) &= \int_0^1 h'(t) dt \\ &= \int_0^1 (h'(0) + (h'(t) - h'(0))) dt \\ &= h'(0) + \int_0^1 (h'(t) - h'(0)) dt \end{aligned}$$

Which means

$$\begin{aligned} |h(1) - h(0) - h'(0)| &= \left| \int_0^1 (h'(t) - h'(0)) dt \right| \\ &\leq \int_0^1 |h'(t) - h'(0)| dt \end{aligned}$$

where the final inequality uses Jensen's inequality for the convex function $|v|$ defined for $v \in \mathbb{R}$. We have

$$\begin{aligned} h(1) &= f(y) \\ h(0) &= f(x) \\ h'(0) &= \nabla f(x)^\top (y - x) \\ h'(t) &= \nabla f(x + t(y - x))^\top (y - x) \end{aligned}$$

⁷Caveat: There may be multiple extensions of $f : X \rightarrow \mathbb{R}$ to the domain $\mathbb{R}^n$, and not all of them are guaranteed to have the same gradient $\nabla \tilde{f}(x)$ at a boundary point $x \in X$. We assume a single extension function $\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ is consistently used for which (40) holds.

and so

$$\begin{aligned}
|f(y) - f(x) - \nabla f(x)^\top (y - x)| &\leq \int_0^1 |(\nabla f(x + t(y - x)) - \nabla f(x))^\top (y - x)| dt \\
&\stackrel{(a)}{\leq} \int_0^1 \|\nabla f(x + t(y - x)) - \nabla f(x)\| \cdot \|y - x\| dt \\
&\stackrel{(b)}{\leq} \int_0^1 \beta t \|y - x\|^2 dt \\
&= \beta \|y - x\|^2 \int_0^1 t dt \\
&= \frac{\beta \|y - x\|^2}{2}
\end{aligned}$$

where (a) holds by the Cauchy-Schwarz inequality; (b) uses the fact that ∇f is β -Lipschitz:

$$\|\nabla f(x + t(y - x)) - \nabla f(x)\| \leq \beta \|t(y - x)\| \quad \forall t \in [0, 1]$$

□

Note that all β -smooth functions are differentiable and hence continuous. For example, fix $a \in \mathbb{R}^n$ and consider $f : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by $f(x) = \|x - a\|^2$ for all $x \in \mathbb{R}^n$, where the Euclidean norm is used. Then $\nabla f(x) = 2(x - a)$ for all $x \in \mathbb{R}^n$ and for all $x, y \in \mathbb{R}^n$ we have

$$\|\nabla f(x) - \nabla f(y)\| = 2\|x - y\|$$

and so f is 2-smooth, that is $\beta = 2$. Further, f is 2-strongly convex because $f(x) - \|x\|^2$ is an affine (hence convex) function.

B. Smooth convex functions

Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set. We seek an approximate solution to the problem of minimizing $f(x)$ over all $x \in C$.

Theorem 16: (Performance for smooth functions) Let $f : C \rightarrow \mathbb{R}$ be convex and β -smooth for some $\beta > 0$. Suppose $x^* \in C$ is a minimizer of f over all $x \in C$. We use an oracle that returns gradients $\nabla f(x)$ for $x \in C$ that satisfy (41). Fix stepsize δ such that $0 < \delta \leq 1/\beta$. Fix $x_0 \in C$ and do

$$x_{k+1} = \pi_C(x_k - \delta \nabla f(x_k)) \quad \forall k \in \{0, 1, 2, \dots\}$$

Then

$$f\left(\frac{1}{m} \sum_{k=1}^m x_m\right) \leq f(x^*) + \frac{\|x^* - x_0\|^2}{2m\delta} \quad \forall m \in \{1, 2, 3, \dots\}$$

and also

$$\min\{f(x_0), f(x_1), \dots, f(x_m)\} \leq f(x^*) + \frac{\|x^* - x_0\|^2}{2m\delta} \quad \forall m \in \{1, 2, 3, \dots\}$$

and so with this fixed stepsize δ , the error decays like $\Theta(1/m)$ as m increases. That is, given $\epsilon > 0$, the convergence time to an $O(\epsilon)$ -approximation is $1/\epsilon$.

Proof: Fix $k \in \{0, 1, 2, \dots\}$. By Lemma 9 the update rule for x_{k+1} is equivalent to choosing $x_{k+1} \in C$ to minimize

$$\nabla f(x_k)^\top (x - x_k) + \frac{1}{2\delta} \|x - x_k\|^2$$

where the norm is a Euclidean norm. This is a $(1/\delta)$ -strongly convex function of x , and since x_{k+1} minimizes this function over all $x \in C$ we have by the quadratic pushback theorem (and the fact that $x^* \in C$):

$$\begin{aligned}
&\nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{1}{2\delta} \|x_{k+1} - x_k\|^2 \\
&\leq \nabla f(x_k)^\top (x^* - x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2 \\
&\leq f(x^*) - f(x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2
\end{aligned}$$

where the final inequality uses the standard subgradient inequality. Since $1/\delta \geq \beta$ we obtain

$$\nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|^2 \leq f(x^*) - f(x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2 \quad (42)$$

However, the β -smooth property implies by the descent lemma (Lemma 17):

$$f(x_{k+1}) - f(x_k) \leq \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|^2 \quad (43)$$

Substituting this inequality into the left-hand-side of (42) gives

$$f(x_{k+1}) - f(x_k) \leq f(x^*) - f(x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2$$

Cancelling $-f(x_k)$ from both sides gives

$$f(x_{k+1}) \leq f(x^*) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2$$

Fix m as a positive integer. Summing over $k \in \{0, \dots, m-1\}$ and dividing by m gives

$$\frac{1}{m} \sum_{k=1}^m f(x_k) \leq f(x^*) + \frac{\|x^* - x_0\|^2 - \|x^* - x_m\|^2}{2m\delta}$$

The final result follows by Jensen's inequality. $\square$

The β -smoothness property is used only for inequality (43). Therefore, the same result holds if the stepsize is less than or equal to $1/\beta_{\text{local}}$, where β_{local} is a *local* smoothness parameter that holds over the line segment between x_{k+1} and x_k for each k over which the algorithm is simulated. This fact is useful for *adaptive* stepsize methods where the algorithm often performs better than expected because the local smoothness parameter is much smaller than the global smoothness parameter over the full domain of the function (see adaptive algorithms in Section VIII-D and simulations in Section VIII-E).

The above result treats smooth convex functions and produces an $O(\delta)$ approximation with convergence time $1/\delta$. This is *not* the best known first-order result for smooth convex functions: An accelerated gradient method developed by Nesterov can improve convergence time to $O(1/\sqrt{\delta})$ [1]. We do not pursue this analysis. Instead, the next subsection treats convex functions that are both smooth and strongly convex, where the *same* projected gradient algorithm yields *exponentially fast* convergence.⁸

C. Strongly convex functions

Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set. We seek an approximate solution to the problem of minimizing a smooth and *strongly convex* $f(x)$ over all $x \in C$. Recall that there exists a unique minimizer $x^* \in C$ (Lemma 16). Note that if $f : C \rightarrow \mathbb{R}$ is both c -strongly convex and β -smooth, we must have $c \leq \beta$ (see Exercise 22).

Theorem 17: (Performance for strongly convex and smooth functions) Fix $c > 0$, $\beta > 0$. Let $f : C \rightarrow \mathbb{R}$ be c -strongly convex and β -smooth. Let $x^* \in C$ be the unique minimizer of f . We use an oracle that returns gradients $\nabla f(x)$ for $x \in C$ such that (41) is upheld. Fix stepsize δ such that $0 < \delta \leq 1/\beta$. Fix $x_0 \in C$ and do

$$x_{k+1} = \pi_C(x_k - \delta \nabla f(x_k)) \quad \forall k \in \{0, 1, 2, \dots\}$$

Then

$$\|x^* - x_k\| \leq \rho^k \|x^* - x_0\| \quad \forall k \in \{1, 2, 3, \dots\}$$

where the Euclidean norm is used, and

$$\rho = \sqrt{\frac{(1/\delta) - c}{(1/\delta) + c}}$$

so $0 \leq \rho < 1$ (recall that $1/\delta \geq c$). Thus, for this sufficiently small stepsize δ , we have $x_k \rightarrow x^*$, and convergence is exponentially fast.

Proof: Fix $k \in \{0, 1, 2, \dots\}$. By Lemma 9, the update rule for x_{k+1} is equivalent to choosing $x_{k+1} \in C$ to minimize

$$\nabla f(x_k)^\top (x - x_k) + \frac{1}{2\delta} \|x - x_k\|^2$$

where the norm is a Euclidean norm. This is a $(1/\delta)$ -strongly convex function of x (recall Lemma 14). Since x_{k+1} minimizes this function over all $x \in C$ we have by the quadratic pushback theorem (and the fact $x^* \in C$):

$$\begin{aligned} & \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{1}{2\delta} \|x_{k+1} - x_k\|^2 \\ & \leq \nabla f(x_k)^\top (x^* - x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2 \\ & \leq f(x^*) - f(x_k) - \frac{c}{2} \|x^* - x_k\|^2 + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2 \end{aligned}$$

⁸For historical reasons, “exponentially fast convergence” is sometimes referred to as “linear convergence.” We do not encourage this terminology. However, it may help to know the terminology exists.

where the final inequality uses the strengthened subgradient inequality for c -strongly convex functions (Theorem 15). Since $1/\delta \geq \beta$ we obtain

$$\nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|^2 \leq f(x^*) - f(x_k) + \frac{1/\delta - c}{2} \|x^* - x_k\|^2 - \frac{1/\delta}{2} \|x^* - x_{k+1}\|^2 \quad (44)$$

However, the β -smooth property implies by the descent lemma (Lemma 17):

$$f(x_{k+1}) - f(x_k) \leq \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|^2$$

Substituting this inequality into the left-hand-side of (44) gives

$$f(x_{k+1}) - f(x_k) \leq f(x^*) - f(x_k) + \frac{(1/\delta) - c}{2} \|x^* - x_k\|^2 - \frac{1/\delta}{2} \|x^* - x_{k+1}\|^2$$

Cancelling $-f(x_k)$ from both sides gives

$$f(x_{k+1}) \leq f(x^*) + \frac{(1/\delta) - c}{2} \|x^* - x_k\|^2 - \frac{1/\delta}{2} \|x^* - x_{k+1}\|^2 \quad (45)$$

However, the Quadratic pushback theorem (Theorem 14) ensures

$$f(x^*) \leq f(x_{k+1}) - \frac{c}{2} \|x^* - x_{k+1}\|^2$$

Substituting this inequality into the right-hand-side of (45) gives

$$f(x_{k+1}) \leq f(x_{k+1}) + \frac{(1/\delta) - c}{2} \|x^* - x_k\|^2 - \frac{(1/\delta) + c}{2} \|x^* - x_{k+1}\|^2$$

Rearranging terms gives

$$\|x^* - x_{k+1}\|^2 \leq \underbrace{\left(\frac{(1/\delta) - c}{(1/\delta) + c} \right)}_{\rho^2} \|x^* - x_k\|^2$$

□

The above theorem shows convergence of the iterates x_k to the optimal solution x^* . Bounds on the objective function value can be obtained in various ways, see Exercise 21 for one way. An interesting fact is that if the algorithm of Theorem 17 is applied to a quadratic function $f : X \rightarrow \mathbb{R}$ defined by

$$f(x) = \frac{\beta}{2} \|x - a\|^2 \quad \forall x \in X$$

which is a function that is both β -smooth and β -strongly convex, then stepsize $\delta = 1/\beta$ yields *exact* optimality in one step (see Exercise 26). Specifically, this holds for any nonempty convex set $X \subseteq \mathbb{R}^n$, any $a \in \mathbb{R}^n$, any $\beta > 0$.

D. Adaptive step size

For implementation of projected gradient descent for β -smooth convex functions, Theorems 16 and 17 require the positive stepsize δ to be chosen small enough to satisfy $\delta \leq 1/\beta$. This requires knowledge of the parameter β . Variations on the method consider line searches and backtracking (see, for example, [2]). This section provides a related but simpler halving method.

Setup: Assume $C \subseteq \mathbb{R}^n$ is a nonempty, closed, convex set. Suppose $f : C \rightarrow \mathbb{R}$ is β -smooth for some *unknown* $\beta > 0$. We have an oracle that provides both $f(x)$ and $\nabla f(x)$ when queried about a given $x \in C$. We want to solve

$$\begin{aligned} &\text{Minimize: } f(x) \\ &\text{Subject to: } x \in C \end{aligned}$$

Let $x^* \in C$ be a minimizer of f . Fix initial stepsize $\delta_0 > 0$ (without knowing β). For example, we can use $\delta_0 = 1$.

Idea: We update the stepsize so that

$$\delta_0 \geq \delta_1 \geq \dots \geq \delta_k > 0$$

Throughout we use notation $\alpha_k = 1/\delta_k$ for all $k \in \{0, 1, 2, \dots\}$, so that

$$0 < \alpha_0 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_k$$

At the start of each step $k \in \{0, 1, 2, \dots\}$ we know α_k (and hence $\delta_k = 1/\alpha_k$) and perform the projected gradient descent iteration:

$$\tilde{x}_{k+1} = \pi_C(x_k - \delta_k \nabla f(x_k)) \quad \forall k \in \{0, 1, 2, \dots\} \quad (46)$$

which by Lemma 9 is equivalent to

$$\tilde{x}_{k+1} = \arg \min_{x \in C} \left\{ \nabla f(x_k)^\top (x - x_k) + \frac{\alpha_k}{2} \|x - x_k\|^2 \right\}$$

The resulting $\tilde{x}_{k+1}$ value satisfies (by the quadratic pushback theorem, Theorem 14):

$$\underbrace{\nabla f(x_k)^\top (\tilde{x}_{k+1} - x_k) + \frac{\alpha_k}{2} \|\tilde{x}_{k+1} - x_k\|^2}_{LHS_k} \leq \nabla f(x_k)^\top (z - x_k) + \frac{\alpha_k}{2} \|z - x_k\|^2 - \frac{\alpha_k}{2} \|z - \tilde{x}_{k+1}\|^2 \quad \forall z \in C \quad (47)$$

Let LHS_k denote the numerical value of the left-hand-side of the above inequality. We know that

- $LHS_k \leq 0$ (consider $z = x_k$ in (47)).
- If $\alpha_k \geq \beta$ the descent lemma ensures

$$f(\tilde{x}_{k+1}) - f(x_k) \leq LHS_k \quad (48)$$

The above two bullet points imply that if (48) fails, then we know $\alpha_k < \beta$ and we should increase the α_k parameter (decreasing the stepsize). We double α_k whenever such an event occurs (halving the stepsize). Further, in the case $f(\tilde{x}_{k+1}) > f(x_k)$ (which can only occur if (48) fails) we discard $\tilde{x}_{k+1}$ and start again from x_k but using the smaller stepsize. Specifically:

Adaptive doubling method: Fix $x_0 \in C$. Fix $\alpha_0 > 0$ (such as $\alpha_0 = 1$). Fix $tolerance > 0$ as a small value for tolerating numerical imprecision (such as $tolerance = 0.0000001$). Use oracle to obtain $f(x_0), \nabla f(x_0)$. Use $f(x_0)$ to initialize a running best function value. At the start of each step $k \in \{0, 1, 2, \dots\}$ do

1) Define

$$\tilde{x}_{k+1} := \pi_C(x_k - (1/\alpha_k) \nabla f(x_k))$$

2) Use oracle to find $f(\tilde{x}_{k+1})$ and $\nabla f(\tilde{x}_{k+1})$.

3) Consider one of three possibilities:

- If $f(\tilde{x}_{k+1}) > f(x_k) + tolerance$ then:
 - Set $x_{k+1} := x_k, \alpha_{k+1} = 2\alpha_k$.
 - Set $f(x_{k+1}) := f(x_k), \nabla f(x_{k+1}) := \nabla f(x_k)$.
- Else, if $f(\tilde{x}_{k+1}) - f(x_k) > LHS_k + tolerance$, then:
 - Set $x_{k+1} := \tilde{x}_{k+1}, \alpha_{k+1} := 2\alpha_k$.
 - Set $f(x_{k+1}) := f(\tilde{x}_{k+1}), \nabla f(x_{k+1}) := \nabla f(\tilde{x}_{k+1})$.
- Else:
 - Set $x_{k+1} := \tilde{x}_{k+1}$ and $\alpha_{k+1} := \alpha_k$.
 - Set $f(x_{k+1}) := f(\tilde{x}_{k+1}), \nabla f(x_{k+1}) := \nabla f(\tilde{x}_{k+1})$.

4) Use $f(x_{k+1})$ to update running best function value.

5) Go to bullet 1 for step $k + 1$.

If $\alpha_0 \geq \beta$ then the stepsize never changes, so $\alpha_k = \alpha_0 \geq \beta$ for all $k \in \{0, 1, 2, \dots\}$. Then $\delta_k = \delta = 1/\alpha_0 \leq 1/\beta$ for all k , so the algorithm yields the same performance as the fixed stepsize method developed in Theorem 16 (for β -smooth convex functions) and Theorem 17 (for β -smooth and c -strongly convex functions).

On the other hand, if $\alpha_0 < \beta$ then the algorithm only performs a doubling step if $\alpha_k < \beta$. It follows that the α_k parameter either stays the same or doubles, and never doubles after it meets or exceeds the threshold β . Thus

$$0 < \alpha_0 \leq \alpha_k \leq \alpha_{k+1} \leq 2\beta \quad \forall k$$

Further, the number of doubling steps is at most

$$z = \left\lceil \frac{\log(\beta/\alpha_0)}{\log(2)} \right\rceil \quad (49)$$

and so in m iterations there is one sequence of at least $m/(z+1)$ consecutive time slots k for which (48) holds and $x_{k+1} = \tilde{x}_k$, so that

$$f(x_{k+1}) - f(x_k) \leq LHS_k + tolerance$$

so that $\alpha_k = \alpha$ (constant) over this sequence of steps and (47) reduces to (using $z = x^* \in C$):

$$f(x_{k+1}) - f(x_k) \leq \nabla f(x_k)^\top (x^* - x_k) + \frac{\alpha}{2} \|x^* - x_k\|^2 - \frac{\alpha}{2} \|x^* - x_{k+1}\|^2 + tolerance$$

so the result of Theorems 16 and 17 can be applied over this sequence of $m/(z+1)$ steps. Exercises 44-45 consider adaptation concepts.

E. Simulation of adaptive stepsize methods

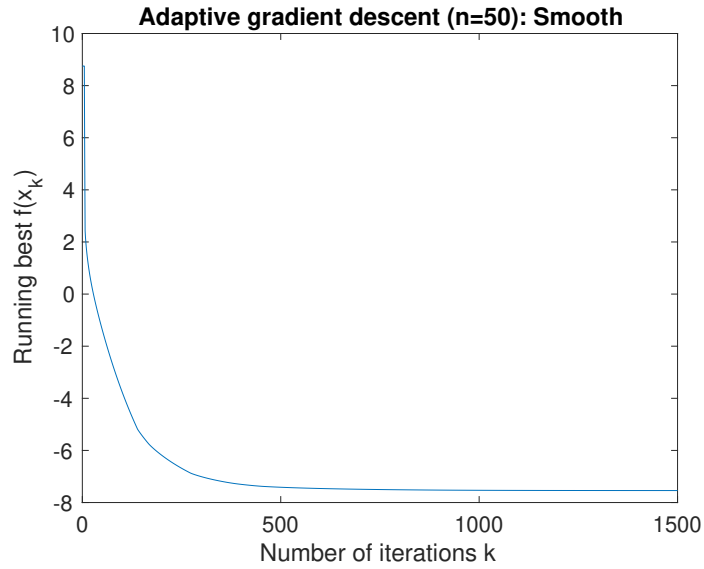


Fig. 1. Smooth case with f in (50) and $n = 50, k = 20, C = [-1, 1]^n$. There were a total of 7 doubling instances, 5 of them due to $f(\tilde{x}_{k+1}) > f(x_k)$.

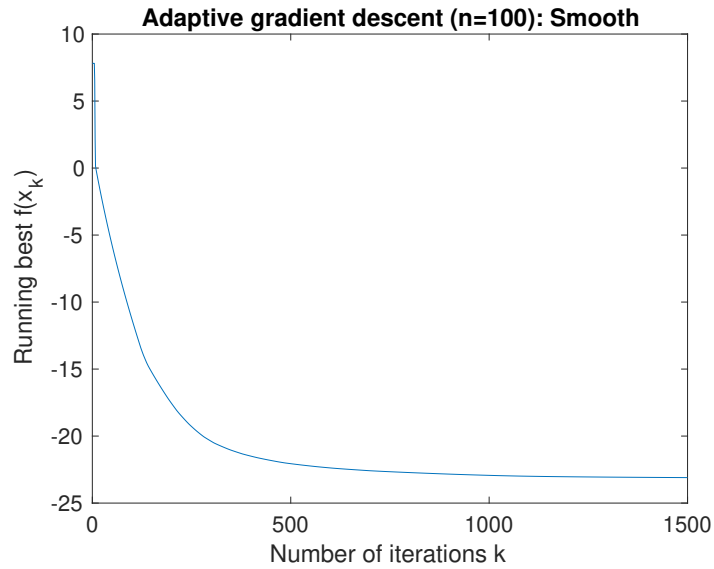


Fig. 2. Smooth case with f in (50) and $n = 100, k = 20, C = [-1, 1]^n$. There were a total of 7 doubling instances, 5 of them due to $f(\tilde{x}_{k+1}) > f(x_k)$.

- Smooth situations: Fix positive integers n, k, m and define $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$f(x) = c^\top x + \|Ax + b\|^2 + \frac{1}{m} \sum_{j=1}^m e^{d_j^\top x + r_j} \quad \forall x \in \mathbb{R}^n \quad (50)$$

$$\nabla f(x) = c + 2A^\top(Ax + b) + \frac{1}{m} \sum_{j=1}^m d_j e^{d_j^\top x + r_j} \quad \forall x \in \mathbb{R}^n$$

where A is a randomly generated $k \times n$ matrix, $c, d_1, \dots, d_m$ are randomly generated $n \times 1$ vectors, b is a randomly generated $k \times 1$ vector, $r_1, \dots, r_j$ are randomly generated real numbers. (See also computation of gradients for similar functions in Exercise 29.) We consider minimizing $f(x)$ subject to $x \in C$, where $C = [-1, 1]^n$. In all simulations we use $x_0 = 0$ and $\delta_0 = 1$ (so $\alpha_0 = 1/\delta_0 = 1$). When n is large, the global smoothness parameter of this function is ginormous due to the behavior of the exponential functions around undesirable corner points of the domain (see Exercise

29). However, the initial condition $x_0 = 0$ avoids the corners, and our simulations show the adaptive algorithm converges to optimality while staying within a portion of the domain with a small *local* smoothness parameter.

Fig. 1 shows results for the case $n = 50, k = 20, m = 100$, and when components of $A, b, c, d_1, \dots, d_m, r_1, \dots, r_m$ are chosen i.i.d. $\text{Unif}[-1, 1]$. The figure shows results for one such randomly generated f . Similar performance was observed for many different randomly generated functions of this type. There were 7 doubling events, 5 of which were due to $f(\tilde{x}_{k+1}) > f(x_k)$. The algorithm produced

$$[\alpha_0, \dots, \alpha_8] = [1, 2, 4, 8, 16, 32, 64, 128, 128]$$

and $\alpha_k = 128$ for all $k \geq 7$. The local smoothness parameter is less than or equal to 128, even though the global smoothness parameter for this function is orders of magnitude larger.

Fig. 2 shows results for the case $n = 100, k = 20, m = 100$. The figure shows results for one such randomly generated f , although similar performance was observed for many different randomly generated functions of this type. As before, there were 7 doubling events, 5 of which were due to $f(\tilde{x}_{k+1}) > f(x_k)$. The algorithm produced

$$[\alpha_0, \dots, \alpha_8] = [1, 2, 4, 8, 16, 32, 64, 128, 128]$$

and $\alpha_k = 128$ for all $k \geq 7$.

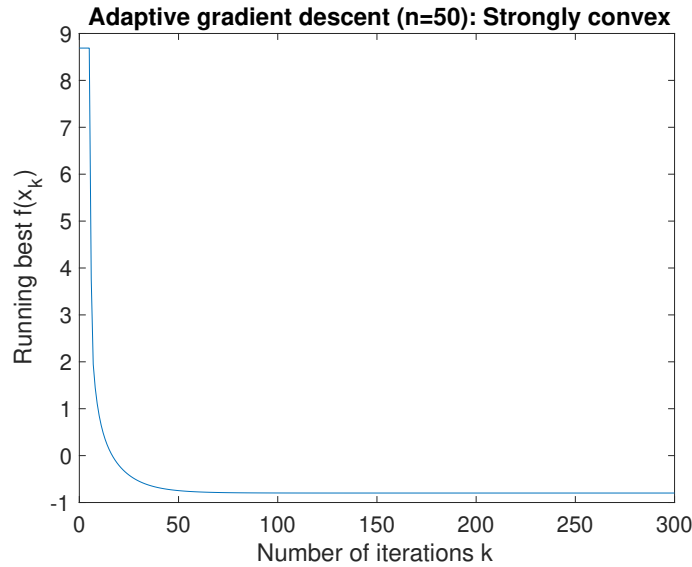


Fig. 3. Strongly convex case with f in (51) and $n = 50, k = 20, C = [-1, 1]^n$. There were a total of 6 doubling instances, 5 of them due to $f(\tilde{x}_{k+1}) > f(x_k)$.

- Strongly convex (and smooth) situations: Fix positive integers n, k and define $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$f(x) = \|x\|^2 + c^\top x + \|Ax + b\|^2 + \frac{1}{m} \sum_{j=1}^m e^{d_j^\top x + r_j} \quad \forall x \in \mathbb{R}^n \quad (51)$$

$$\nabla f(x) = 2x + c + 2A^\top(Ax + b) + \frac{1}{m} \sum_{j=1}^m d_j e^{d_j^\top x + r_j} \quad \forall x \in \mathbb{R}^n$$

This adds a strongly convex term $\|x\|^2$ to the function in (50). Again we use $C = [-1, 1]^n, x_0 = 0, \alpha_0 = 1$.

Fig. 3 shows results for the case $n = 50, k = 20, m = 100$, and when components of $A, b, c, d_1, \dots, d_m, r_1, \dots, r_m$ are chosen i.i.d. $\text{Unif}[-1, 1]$. The figure shows results for one such randomly generated f . Similar performance was observed for many different randomly generated functions of this type. The number of iterations was significantly less than the corresponding situation in Fig. 1. There were 6 doubling events, 5 of which were due to $f(\tilde{x}_{k+1}) > f(x_k)$. The algorithm produced

$$[\alpha_0, \dots, \alpha_7] = [1, 2, 4, 8, 16, 32, 64, 64]$$

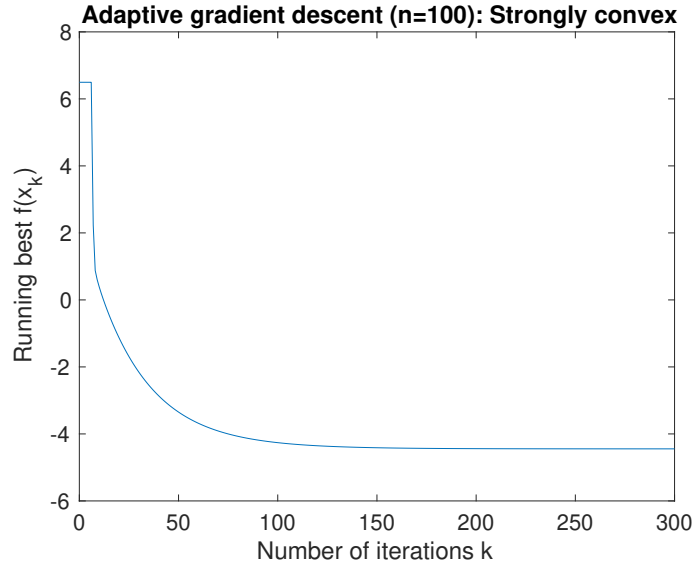


Fig. 4. Strongly convex case with f in (51) and $n = 100, k = 20, C = [-1, 1]^n$. There were a total of 7 doubling instances, 6 of them due to $f(\tilde{x}_{k+1}) > f(x_k)$.

and $\alpha_k = 64$ for all $k \geq 6$.

Fig. 4 shows results for the case $n = 100, k = 20, m = 100$. The figure shows results for one such randomly generated f , although similar performance was observed for many different randomly generated functions of this type. The number of iterations was significantly less than the corresponding situation in Fig. 2. There were 7 doubling events, 6 of which were due to $f(\tilde{x}_{k+1}) > f(x_k)$. The algorithm produced

$$[\alpha_0, \dots, \alpha_8] = [1, 2, 4, 8, 16, 32, 64, 128, 128]$$

and $\alpha_k = 128$ for all $k \geq 7$.

F. Simulation of stochastic gradient descent

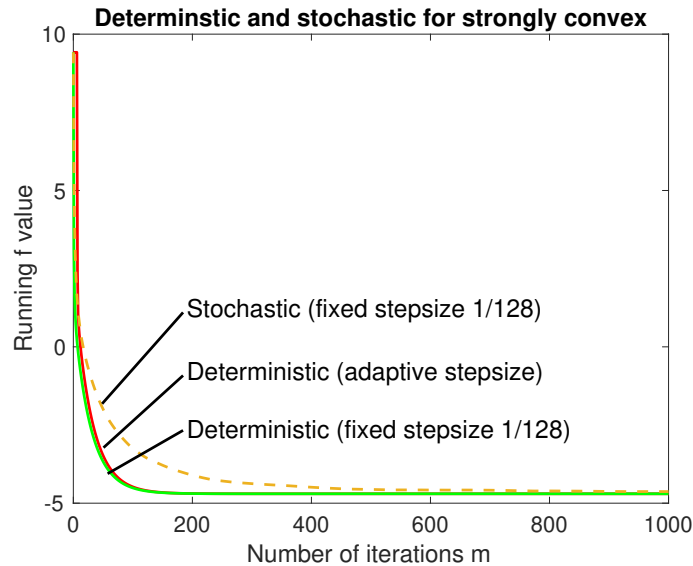


Fig. 5. Comparison of deterministic and stochastic gradient methods for smooth and strongly convex f function in (52) ($n = 100, k = 20, m = 100$). The adaptive method converged to a stepsize of $1/128$ after seven steps (using that for all iterations after). Thus, the fixed stepsize for the deterministic and stochastic methods was chosen to be $\delta = 1/128$. In this strongly convex case, the stochastic oracle is roughly 100 times less complex, but the number of iterations needed increases by less than a factor of 100, making a net win for the stochastic method.

This section compares stochastic gradient descent to deterministic gradient descent. The short summary is that there is a net win for stochastic gradient descent in the case of a strongly convex f considered in this subsection, but there are problems with stochastic gradient descent for the general smooth function f considered here. It should be noted that stochastic gradients are easier to analyze in the general nonsmooth case where slower convergence times apply (see Exercise 42). It is difficult to obtain improved analytical bounds in the smooth case (see Exercise 43). Nevertheless, the stochastic gradient approach can be compared in simulation and appears to work well in a particular strongly convex example (Fig. 5).

Fix positive integers n, k, m . Again consider the same smooth and strongly convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ from (51), repeated here for convenience:

$$\begin{aligned} f(x) &= \|x\|^2 + c^\top x + \|Ax + b\|^2 + \frac{1}{m} \sum_{j=1}^m e^{d_j^\top x + r_j} \quad \forall x \in \mathbb{R}^n \\ \nabla f(x) &= 2x + c + 2A^\top(Ax + b) + \frac{1}{m} \sum_{j=1}^m d_j e^{d_j^\top x + r_j} \quad \forall x \in \mathbb{R}^n \end{aligned} \quad (52)$$

Given $x \in \mathbb{R}^n$, independently generate a random variable $J \sim \text{Unif}\{1, 2, \dots, m\}$ and define the random vector

$$S(x) = 2x + c + 2A^\top(Ax + b) + d_J e^{d_J^\top x + r_J} \quad (53)$$

It can be shown that $S(x)$ is a stochastic gradient because $\mathbb{E}[S(x)] = \nabla f(x)$ (see Exercise 40). For a given nonempty, closed, and convex set $C \subseteq \mathbb{R}^n$ we fix $x_0 \in C$, fix stepsize $\delta > 0$, and compare the following:

- Projected gradient descent: For all $k \in \{0, 1, 2, \dots\}$ do

$$x_{k+1} = \pi_C(x_k - \delta \nabla f(x_k))$$

Here our oracle provides both $f(x_k)$ and $\nabla f(x_k)$ values. We keep track of the running best $f(x_k)$ value. That is, our graph in Fig. 5 plots $\min\{f(x_0), f(x_1), \dots, f(x_{m-1})\}$ versus m .

- Projected stochastic gradient descent: For all $k \in \{0, 1, 2, \dots\}$ do

$$x_{k+1} = \pi_C(x_k - \delta S_k)$$

where at each step k we independently generate $J_k \sim \text{Unif}\{1, \dots, m\}$ and define the stochastic gradient S_k by

$$S_k = c + 2A^\top(Ax_k + b) + d_{J_k} e^{d_{J_k}^\top x_k + r_{J_k}}$$

Here our oracle does not provide the f values, so the algorithm outputs a running average $\bar{x}_m = \frac{1}{m} \sum_{k=0}^{m-1} x_k$ as the number of iterations m increases. However, to compare performance, our graph in Fig. 5 plots the $f(\bar{x}_m)$ value versus m (using the true function value).

Fig. 5 compares performance for the strongly convex f in (52) for the case $C = [-1, 1]^n$, $n = 100$, $k = 20$, $m = 100$. The function f has coefficients of A, b, c, d, r that are randomly chosen i.i.d. $\text{Unif}[-1, 1]$. The simulations use one such randomly generated function f (the same f is used for both the gradient descent and stochastic gradient descent). Qualitatively similar behavior was observed for other random functions of this type. Fig. 5 has three curves: Deterministic gradient with adaptive stepsize (which converged to a stepsize of $1/128$ after 7 iterations); Deterministic gradient with fixed stepsize $1/128$; Stochastic gradient with fixed stepsize $1/128$. The deterministic curves are nearly identical (it should be noted that its stepsize $1/128$ was chosen only *after* observing the empirical stepsize the adaptive method converged to).

While the stochastic gradient method requires more iterations m for convergence, that extra complexity is *more than compensated for* by the fact that its oracle is significantly less complicated (roughly a factor of 100 less complex). Therefore, there is a net win by using a stochastic gradient in this case. As an implementation detail, note that it is not obvious how to use an adaptive stepsize for the stochastic gradient case. This is because the function values f are not computed. However, the deterministic adaptive stepsize method typically adjusts the stepsize to a fixed value after the first few iterations. It often makes sense to use a deterministic adaptive method for the first few iterations, then switch to using a stochastic gradient.

The next simulations (Figs. 6 and 7) show that the stochastic gradient method can have *unanticipated problems*. We use the smooth function f from (50), using the same A, b, c, d parameters from the previous simulation ($n = 100, k = 2, m = 100$). The only difference is that the $\|x\|^2$ term is removed from f . The corresponding stochastic gradient used is

$$S(x) = c + 2A^\top(Ax + b) + d_J e^{d_J^\top x + r_J} \quad (54)$$

Fig. 6 shows results for the f function from (50) for the deterministic gradient method with adaptive stepsize (which quickly converged to stepsize $1/128$); the deterministic gradient method with fixed stepsize $1/128$ (which shows a very similar curve);

the stochastic gradient method with fixed stepsize $1/128$. First note that removal of the strongly convex term $\|x\|^2$ increases the number of iterations needed even for the deterministic methods (the horizontal axis in Figs. 6 and 7 use 10000 iterations, as opposed to 1000 iterations from Fig. 5). As seen in Fig. 6, the stochastic method does not converge to optimality (in fact, it begins to increase). Fig 7 plots results when a reduced stepsize of $1/512$ is used for the stochastic gradient method (the plotted data from the deterministic methods is the same as the previous figure). The reduced stepsize helps the stochastic gradient descent in this example. However, even after 10000 iterations, the stochastic method does not converge to near optimality. Overall, deterministic descent wins in this example.

What is happening, and why the bizarre behavior when the stepsize $1/128$ is used for stochastic gradient descent in Fig. 6? One problem is that, while stochastic gradients have analytical bounds for small stepsizes for general (possibly nonsmooth) convex functions (see Exercise 42), we do not have the same analytical bounds and convergence behavior for fixed stepsizes $\delta \leq 1/\beta$ in the β -smooth regime (see Exercise 43). Also, the function f used in both (50) and (52) is considered over the 100-dimensional set $[-1, 1]^{100}$. This set is large and contains regions where the exponential terms make the f function and its smoothness parameter large. Therefore, the smoothness parameter β is larger than 128. Fortunately, the adaptive method makes steps that only require stepsize $1/128$, which means the *local smoothness parameter* is less than or equal to 128 in the region of the iterations explored. However, the stochastic method chooses random directions that explore regions of $[-1, 1]^{100}$ where the smoothness parameter is larger than 128. Therefore, the stepsize is too large for the stochastic method of Fig. 6. Performance improves when the stepsize is reduced to $1/512$ (Fig. 7). This phenomenon does not occur in Fig. 5 because the $\|x\|^2$ term tends to make even the stochastic method use iterates x that have components close to the origin, which avoids the extreme regions of $[-1, 1]^{100}$ where the smoothness parameter is worse.

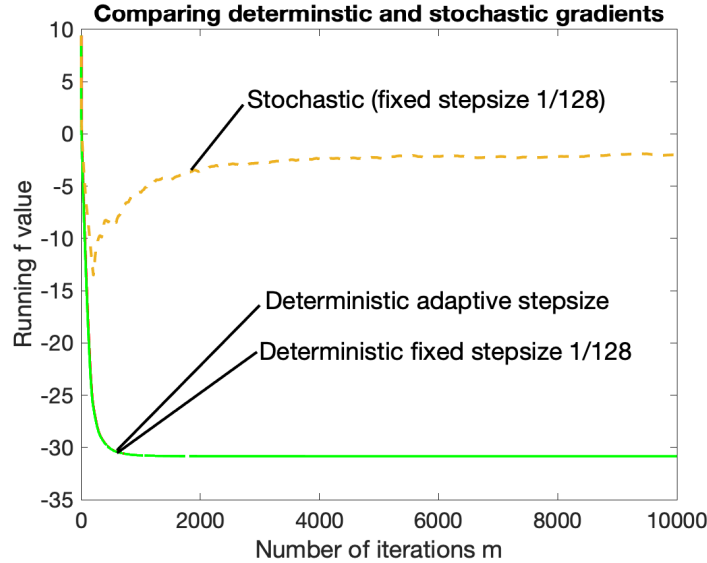


Fig. 6. Comparing deterministic and stochastic gradient descent for f in (50) ($n = 100, k = 20, m = 100$). The deterministic method with adaptive stepsize quickly converges to a stepsize of $1/128$, so this stepsize is used for the two fixed-stepsize curves. The stochastic gradient descent method performs poorly.

IX. NONCONVEX OPTIMIZATION

This section considers nonconvex optimization: We have a β -smooth but possibly nonconvex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that we want to minimize. Define

$$f^* = \inf_{x \in \mathbb{R}^n} \{f(x)\}$$

and assume f^* is finite. The search is over all $x \in \mathbb{R}^n$. Fix $x_0 \in \mathbb{R}^n$. Consider the following iteration:

$$x_{k+1} = x_k - \delta_k \nabla f(x_k) \quad \forall k \in \{0, 1, 2, \dots\} \quad (55)$$

for some stepsizes $\delta_k > 0$. Instead of an exact minimizer, also called a *global optimal point*, we seek an approximate local optimal point. In fact the goal is somewhat weaker than that: We seek a point $\tilde{x} \in \mathbb{R}^n$ that satisfies $\nabla f(\tilde{x}) \approx 0$. A point $x \in \mathbb{R}^n$ that satisfies $\nabla f(x) = 0$ is called a *stationary point*. Thus, we seek an approximate stationary point. Of course, a stationary point does not necessarily relate to a local minimum (it could be an inflection point, or a local *maximum*). However, we shall move in the direction of decreasing the function f , so our final point is likely to be desirable. Throughout this section we use $\|\cdot\|$ as the standard Euclidean norm.

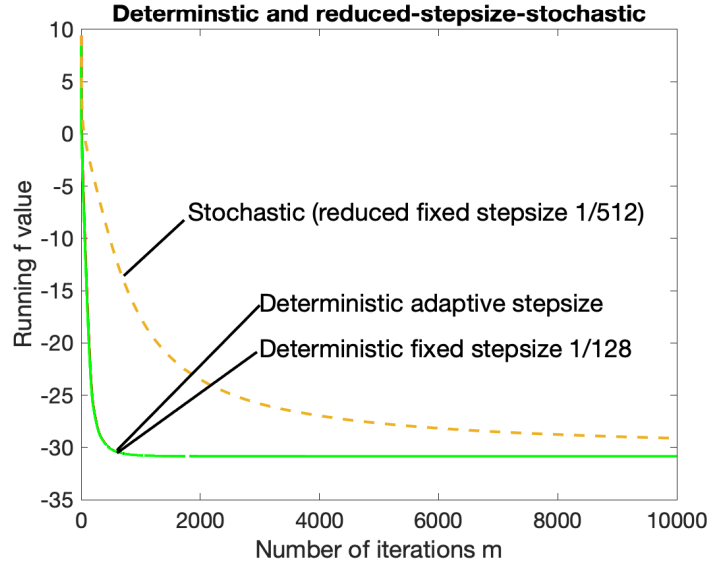


Fig. 7. This reduces the fixed stepsize for the stochastic gradient method from $1/128$ to $1/512$. The deterministic curves are the same as in Fig. 6. The same function f from (50) is used ($n = 100, k = 20, m = 100$).

A. Fixed stepsize $\delta < 2/\beta$

Here we consider a fixed stepsize $\delta_k = \delta$ for all k . Suppose we have an upper bound on β and so we choose a stepsize $\delta \in (0, 2/\beta)$. We use $\delta_k = \delta$ for all k . For each step $k \in \{0, 1, 2, \dots\}$ we have

$$\begin{aligned} f(x_{k+1}) &\stackrel{(a)}{\leq} f(x_k) + \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{\beta}{2} \|x_{k+1} - x_k\|^2 \\ &\stackrel{(b)}{=} f(x_k) + \nabla f(x_k)^\top (-\delta \nabla f(x_k)) + \frac{\beta \delta^2}{2} \|\nabla f(x_k)\|^2 \\ &= f(x_k) - \delta \|\nabla f(x_k)\|^2 (1 - \beta \delta / 2) \end{aligned} \quad (56)$$

where (a) holds by the descent Lemma 17; (b) holds by definition of x_{k+1} in (55) (using $\delta_k = \delta$ for all k). Since $(1 - \beta \delta / 2) > 0$, we can rearrange terms as

$$\|\nabla f(x_k)\|^2 \leq \frac{f(x_k) - f(x_{k+1})}{\delta(1 - \beta \delta / 2)} \quad \forall k \in \{0, 1, 2, \dots\}$$

It follows that our iterates are *always improving*:

$$f(x_{k+1}) \leq f(x_k) \quad \forall k \in \{0, 1, 2, \dots\}$$

and the only time the inequality holds with equality is when $\nabla f(x_k) = 0$ (which means we can terminate the iterations and use x_k as our stationary point). Fix m as a positive integer. Summing over $k \in \{0, 1, \dots, m-1\}$ and dividing by m gives

$$\begin{aligned} \frac{1}{m} \sum_{k=0}^{m-1} \|\nabla f(x_k)\|^2 &\leq \frac{f(x_0) - f(x_m)}{m\delta(1 - \beta \delta / 2)} \\ &\leq \frac{f(x_0) - f^*}{m\delta(1 - \beta \delta / 2)} \end{aligned}$$

It follows that

$$\min \{\|\nabla f(x_0)\|^2, \dots, \|\nabla f(x_{m-1})\|^2\} \leq \frac{f(x_0) - f^*}{m\delta(1 - \beta \delta / 2)} \quad \forall m \in \{1, 2, 3, \dots\} \quad (57)$$

and so the norm-square of the smallest magnitude gradient seen in the first m iterations decays like $O(1/m)$. Thus, the best-seen norm of the gradient decays like $O(1/\sqrt{m})$.

If the value of β is known, we can choose the stepsize δ to maximize the denominator term $\delta(1 - \beta \delta / 2)$ to obtain the best performance. The best stepsize to choose is

$$\delta^* = 1/\beta$$

from which we obtain (using $\delta = 1/\beta$):

$$\min \{\|\nabla f(x_0)\|^2, \dots, \|\nabla f(x_{m-1})\|^2\} \leq \frac{2\beta(f(x_0) - f^*)}{m} \quad \forall m \in \{1, 2, 3, \dots\} \quad (58)$$

This requires β to be fully known. The next subsection shows that we can come within roughly a factor of 2 of this performance by an adaptive stepsize method when β is unknown.

B. Adaptive stepsize for nonconvex minimization

Again we have a β -smooth but possibly nonconvex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that we want to minimize. However, the value β is unknown. Suppose we are at a point $x_k \in \mathbb{R}^n$ and we consider using some stepsize $\tilde{\delta} > 0$. Let $\tilde{x}$ be the next point we would get if we did choose that stepsize:

$$\tilde{x} = x_k - \tilde{\delta} \nabla f(x_k)$$

From the same argument as (56) we obtain (using the unknown value β):

$$f(\tilde{x}) \leq f(x_k) - \tilde{\delta} \|\nabla f(x_k)\|^2 (1 - \beta \tilde{\delta}/2)$$

In the case $\tilde{\delta} \leq 1/\beta$ then $(1 - \beta \tilde{\delta}/2) \geq 1/2$ and we obtain

$$f(\tilde{x}) \leq f(x_k) - (\tilde{\delta}/2) \|\nabla f(x_k)\|^2 \quad (59)$$

Therefore, if for our considered stepsize $\tilde{\delta}$ the inequality (59) fails, we know $\tilde{\delta} > 1/\beta$ and we can halve the stepsize. Specifically

Nonconvex gradient descent with adaptive stepsize: Fix $x_0 \in \mathbb{R}^n$ and $\delta_{-1} > 0$ (such as $x_0 = 0$ and $\delta_{-1} = 1$). Fix $tolerance > 0$ as a small value for tolerating numerical imprecision (such as $tolerance = 0.0000001$). From oracle, obtain

$$f(x_0), \nabla f(x_0)$$

Use these to initialize running bests for function value and norm-squared gradient. At the start of each step $k \in \{0, 1, 2, \dots\}$ do the following (using variables $\tilde{\delta}$ and $\tilde{x}$ that are changed potentially multiple times for step k):

1) Define $\tilde{\delta} := \delta_{k-1}$ and enter the following loop:

2) Loop:

- Define $\tilde{x} := x_k - \tilde{\delta} \nabla f(x_k)$
- Obtain $f(\tilde{x})$ from oracle. Test the condition

$$f(\tilde{x}) \leq f(x_k) - (\tilde{\delta}/2) \|\nabla f(x_k)\|^2 + tolerance \quad (60)$$

- If (60) holds then define $x_{k+1} = \tilde{x}$, $\delta_k = \tilde{\delta}$, and exit the loop.
- If (60) fails then define $\tilde{\delta} := \tilde{\delta}/2$ and repeat the loop.

3) Set $f(x_{k+1}) := f(\tilde{x})$ [already found by oracle in loop].

4) From oracle, obtain $\nabla f(x_{k+1})$.

5) Use $f(x_{k+1})$ and $\nabla f(x_{k+1})$ to update running bests for function value and norm-squared gradient.

6) Go to bullet 1 for step $k + 1$.

Observe that at each step k , a stepsize δ_k is chosen so that (60) holds, and so

$$f(x_{k+1}) \leq f(x_k) - (\delta_k/2) \|\nabla f(x_k)\|^2 + tolerance \quad \forall k \in \{0, 1, 2, \dots\} \quad (61)$$

To analyze the algorithm, we consider two cases:

- Case 1: $\delta_{-1} \leq 1/\beta$. In this case the condition (60) never fails, we have $\delta_k = \delta_{-1}$ always, and (61) becomes

$$f(x_{k+1}) \leq f(x_k) - (\delta_{-1}/2) \|\nabla f(x_k)\|^2 + tolerance \quad \forall k \in \{0, 1, 2, \dots\}$$

Fix m as a positive integer. Summing over $k \in \{0, \dots, m-1\}$ and dividing by m gives

$$\frac{f(x_m) - f(x_0)}{m} \leq -(\delta_{-1}/2) \frac{1}{m} \sum_{k=0}^{m-1} \|\nabla f(x_k)\|^2 + tolerance$$

and so

$$\min\{\|\nabla f(x_0)\|^2, \dots, \|\nabla f(x_{m-1})\|^2\} \leq \frac{2(f(x_0) - f^*)}{m\delta_{-1}} + \frac{2tolerance}{\delta_{-1}} \quad \forall m \in \{1, 2, 3, \dots\}$$

- Case 2: $\delta_{-1} > 1/\beta$. Recall that (60) can only fail if $\tilde{\delta} > 1/\beta$. Then it is possible for condition (60) to fail, but each time it fails the $\tilde{\delta}$ value is halved. It follows that we always have $\tilde{\delta} > 1/(2\beta)$, so $\delta_k > 1/(2\beta)$ for all $k \in \{0, 1, 2, \dots\}$ and (61) becomes

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{4\beta} \|\nabla f(x_k)\|^2 + tolerance \quad \forall k \in \{0, 1, 2, \dots\}$$

Fix m as a positive integer. Summing over $k \in \{0, \dots, m-1\}$ and dividing by m gives

$$\frac{f(x_m) - f(x_0)}{m} \leq -\frac{1}{4\beta} \frac{1}{m} \sum_{k=0}^{m-1} \|\nabla f(x_k)\|^2 + \text{tolerance}$$

and so

$$\min\{\|\nabla f(x_0)\|^2, \dots, \|\nabla f(x_{m-1})\|^2\} \leq \frac{4\beta(f(x_0) - f^*)}{m} + 4\beta \text{tolerance} \quad \forall m \in \{1, 2, 3, \dots\}$$

Moreover, the condition (60) can fail at most z times where $z = \left\lceil \frac{\log(\beta\delta)}{\log(2)} \right\rceil$.

Combining cases 1 and 2 we see that

$$\min\{\|\nabla f(x_0)\|^2, \dots, \|\nabla f(x_{m-1})\|^2\} \leq \max\{4\beta, 2/\delta_{-1}\} \left(\frac{(f(x_0) - f^*)}{m} + \text{tolerance} \right) \quad \forall m \in \{1, 2, 3, \dots\}$$

C. Simulation of adaptive algorithm for nonconvex example

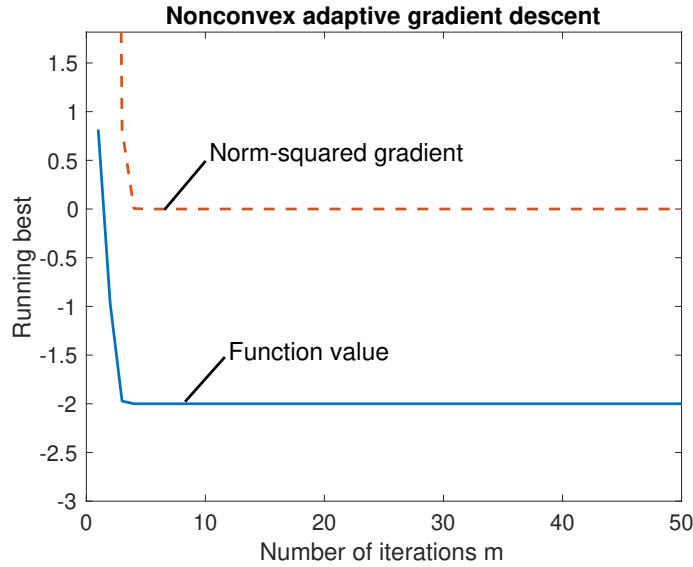


Fig. 8. Nonconvex gradient descent with adaptive stepsize ($x_0 = 0, \delta_{-1} = 1$): The vertical axis shows the running best $\|\nabla f(x_k)\|^2$ value (which converges to zero), and the running best $f(x_k)$ value (which converges to -2).

Fig. 8 shows the result of the adaptive stepsize method using $x_0 = 0, \delta_{-1} = 1$ and $f : \mathbb{R}^{40} \rightarrow \mathbb{R}$ given by the smooth but nonconvex function

$$f(x) = \sin(c^\top x) + \cos(a^\top x + b) \quad \forall x \in \mathbb{R}^{40}$$

where c, a are random vectors of size 40 with i.i.d. components that are $\text{Unif}[-1, 1]$, and b is independent and $\text{Unif}[-1, 1]$ (one such random function is used). The halving step occurs only 4 times. The resulting δ_k values generated by the adaptive algorithm are

$$[\delta_0, \delta_1, \delta_2, \delta_3] = [1/8, 1/8, 1/16, 1/16]$$

and $\delta_k = 1/16$ for all $k \geq 2$. The empirically-achieved minimum function value is -2, the empirically-achieved minimum norm-squared gradient is 2.5×10^{-31} . In this case, with high probability, there exists an $x \in \mathbb{R}^n$ such that $c^\top x = -\pi/2$ and $a^\top x + b = \pi$, which implies $f^* = -2$ for this example. This high probability event in fact occurs with probability 1 if true (infinite precision) uniform random variables are used. While no numerical scheme has infinite precision, the event indeed occurs for the randomly generated function in this example (all that is required is that the pair of vectors $\{a, c\}$ are linearly independent). Thus, in this special case nonconvex example, the algorithm converges closely to a global minimizer.

X. DUALITY AND LAGRANGE MULTIPLIERS

This section presents a theory of Lagrange multipliers and duality. It ends by computing a subgradient of a dual function to motivate the famous *dual subgradient algorithm* for constrained optimization problems.

A. Lagrange multipliers

Fix n, k, m as nonnegative integers (assume $n > 0, k + m > 0$). Fix $X \subseteq \mathbb{R}^n$ as a nonempty set (possibly nonconvex). Fix $f : X \rightarrow \mathbb{R}, g_i : X \rightarrow \mathbb{R}$ for $i \in \{1, \dots, k\}, h_j : X \rightarrow \mathbb{R}$ for $j \in \{1, \dots, m\}$ as real-valued functions (possibly nonconvex). We consider two types of problems: The first we shall call the *constrained problem* and the second the *unconstrained problem*. For notation, a hyphen between equation numbers denotes an inclusive “through,” so “problem (62)-(65)” is the problem defined by (62), (63), (64), (65).

Constrained problem: [given $c \in \mathbb{R}^k, d \in \mathbb{R}^m$]

$$\text{Minimize: } f(x) \quad (62)$$

$$\text{Subject to: } g_i(x) \leq c_i \quad \forall i \in \{1, \dots, k\} \quad (63)$$

$$h_j(x) = d_j \quad \forall j \in \{1, \dots, m\} \quad (64)$$

$$x \in X \quad (65)$$

Unconstrained problem: [given $\mu \in \mathbb{R}^k, \lambda \in \mathbb{R}^m, \mu \geq 0$]

$$\text{Minimize: } f(x) + \sum_{i=1}^k \mu_i g_i(x) + \sum_{j=1}^m \lambda_j h_j(x) \quad (66)$$

$$\text{Subject to: } x \in X \quad (67)$$

The constrained problem (62)-(65) is defined by parameters $c \in \mathbb{R}^k, d \in \mathbb{R}^m$. Different c, d vectors lead to different constrained problems. Likewise, the unconstrained problem (66)-(67) is defined by parameters $\mu \in \mathbb{R}^k, \lambda \in \mathbb{R}^m$, where $\mu \geq 0$. Different μ, λ vectors lead to different unconstrained problems. The parameters μ_i and λ_j are called *Lagrange multipliers*. The Lagrange multiplier μ_i for an inequality constraint must be nonnegative, but the Lagrange multiplier λ_j for an equality constraint can be any real number (possibly negative). The problem (66)-(67) is called the “unconstrained” problem because it removes the inequality and equality constraints (63), (64) (it still has the set constraint $x \in X$). What is the relationship between the two problems? Between the parameters (c, d) and (μ, λ) ? The following theorem is both useful and easy to prove. The idea is that if x^* is a solution to an unconstrained problem defined by some parameters (μ, λ) , then we can manufacture a constrained problem that x^* also solves. We build the constrained problem by specifying c, d parameters in terms of x^* .

Theorem 18: Fix $\mu \in \mathbb{R}^k, \lambda \in \mathbb{R}^m$ with $\mu \geq 0$. Suppose $x^* \in \mathbb{R}^n$ solves the unconstrained problem defined by (μ, λ) . Then a) x^* also solves the constrained problem for the special case when we define c_i and d_j by

$$\begin{aligned} c_i &= g_i(x^*) \quad \forall i \in \{1, \dots, k\} \\ d_j &= h_j(x^*) \quad \forall j \in \{1, \dots, m\} \end{aligned}$$

b) x^* also solves the constrained problem defined by any vectors $c \in \mathbb{R}^k, d \in \mathbb{R}^m$ that satisfy $d_j = h_j(x^*)$ for all $j \in \{1, \dots, m\}$ and $g_i(x^*) \leq c_i, \mu_i(g_i(x^*) - c_i) = 0$ for all $i \in \{1, \dots, k\}$. In particular, if $\mu_i = 0$, then we only require $g_i(x^*) \leq c_i$ (so equality is not required).

Proof: We prove only part (a) (the proof of (b) is similar). Fix vectors μ and λ with $\mu \geq 0$. Suppose x^* solves the unconstrained problem defined by parameters (μ, λ) . Define c_i and d_j as in the statement of the theorem. We want to show x^* satisfies the definition of a solution to the constrained problem associated with parameter vectors c, d .

i) (Constraints) Since x^* solves the unconstrained problem, it must satisfy $x^* \in X$ (this is a requirement of the unconstrained problem). So x^* satisfies constraint (65) of the constrained problem. Vector x^* also satisfies constraints (63) and (64) of the constrained problem by definition of c and d . So x^* satisfies all constraints of the constrained problem.

ii) (Optimality) Suppose $w \in \mathbb{R}^n$ is another vector that satisfies all constraints of the constrained problem, namely, $w \in X$ and $g_i(w) \leq c_i$ for all $i, h_j(w) = d_j$ for all j . We want to show $f(x^*) \leq f(w)$. Since x^* minimizes the unconstrained objective over all $x \in X$, and since $w \in X$, we have

$$\begin{aligned} f(x^*) + \sum_{i=1}^k \mu_i g_i(x^*) + \sum_{j=1}^m \lambda_j h_j(x^*) &\stackrel{(a)}{\leq} f(w) + \sum_{i=1}^k \mu_i g_i(w) + \sum_{j=1}^m \lambda_j h_j(w) \\ &\stackrel{(b)}{\leq} f(w) + \sum_{i=1}^k \mu_i c_i + \sum_{j=1}^m \lambda_j d_j \\ &\stackrel{(c)}{=} f(w) + \sum_{i=1}^k \mu_i g_i(x^*) + \sum_{j=1}^m \lambda_j h_j(x^*) \end{aligned}$$

where (a) holds by definition of x^* being a minimizer of the unconstrained problem; (b) holds because $\mu_i \geq 0$ and $g_i(w) \leq c_i$ for all i (and $h_j(w) = d_j$ for all j); (c) holds by definition of c_i, d_j . Canceling the common terms yields $f(x^*) \leq f(w)$. $\square$

B. Example 1

Theorem 18 often leads to exact solutions to problems with a single constraint. Consider the problem

$$\text{Maximize: } \sum_{i=1}^n w_i \sqrt{x_i} \quad (68)$$

$$\text{Subject to: } \sum_{i=1}^n x_i \leq \theta \quad (69)$$

$$x_i \geq 0 \quad \forall i \in \{1, \dots, n\} \quad (70)$$

where w_i and θ are given positive constants. Define $f(x) = -\sum_{i=1}^n w_i \sqrt{x_i}$, $g(x) = \sum_{i=1}^n x_i$. Define X as the set of all vectors x that satisfy (70). This problem has the structure (62)-(65) with $k = 1, m = 0$.

Fixing $\mu \geq 0$ as a real-valued Lagrange multiplier for the single inequality constraint (69), the corresponding unconstrained problem seeks to find an $x \in X$ to minimize $f(x) + \mu g(x)$, that is, we minimize

$$-\sum_{i=1}^n w_i \sqrt{x_i} + \mu \sum_{i=1}^n x_i$$

which is the same as solving

$$\text{Minimize: } \sum_{i=1}^n (-w_i \sqrt{x_i} + \mu x_i) \quad (71)$$

$$\text{Subject to: } x_i \geq 0 \quad \forall i \in \{1, \dots, n\} \quad (72)$$

It is clear that if $\mu = 0$, this unconstrained problem has no optimal solution because we can take $x_i \rightarrow \infty$. So we focus on the case $\mu > 0$. The unconstrained problem (71)-(72) is *separable*, meaning the objective is a separable sum over terms that involve only a single variable, and the constraints (72) allow each variable x_i to be chosen separately to minimize its corresponding term. For each $i \in \{1, \dots, n\}$, we choose $x_i \geq 0$ to minimize $-w_i \sqrt{x_i} + \mu x_i$. The solution is

$$x_i^*(\mu) = \frac{w_i^2}{4\mu^2} \quad \forall i \in \{1, \dots, n\}$$

where we use notation $x_i^*(\mu)$ to emphasize dependence on the Lagrange multiplier $\mu > 0$. By Theorem 18, we want to find a $\mu > 0$ that satisfies $\sum_{i=1}^n x_i^*(\mu) = \theta$. So

$$\sum_{i=1}^n \frac{w_i^2}{4\mu^2} = \theta \implies \frac{1}{4\mu^2} = \frac{\theta}{\sum_{i=1}^n w_i^2}$$

So we obtain as solution to the problem (68)-(70):

$$x_i^* = \frac{\theta w_i^2}{\sum_{j=1}^n w_j^2} \quad \forall i \in \{1, \dots, n\}$$

C. Example 2 (Water filling)

Consider the problem

$$\text{Maximize: } \sum_{i=1}^n \log(1 + \beta_i x_i) \quad (73)$$

$$\text{Subject to: } \sum_{i=1}^n x_i \leq \theta \quad (74)$$

$$x_i \geq 0 \quad \forall i \in \{1, \dots, n\} \quad (75)$$

where β_i and θ are given positive parameters. This is a famous problem of allocating power over multiple separate channels to maximize a sum transmission rate. Indeed, suppose we have n separate channels. Parameter β_i is a channel quality for channel i . Let x_i represent the amount of power allocated to channel i . The expression $\log(1 + \beta_i x_i)$ is the *Shannon capacity* (normalized to using a natural logarithm). The value θ is a constraint on the sum power used.

Define $f(x) = -\sum_{i=1}^n \log(1 + \beta_i x_i)$, $g(x) = \sum_{i=1}^n x_i$. Define X as the set of all x vectors that satisfy (75). This problem again has the structure (62)-(65) with $k = 1, m = 0$. Fix a real number $\mu > 0$. The corresponding unconstrained problem is to choose $x \in X$ to minimize

$$\sum_{i=1}^n (-\log(1 + \beta_i x_i) + \mu x_i)$$

which is a separable problem: For each $i \in \{1, \dots, n\}$, choose $x_i \geq 0$ to minimize $-\log(1 + \beta_i x_i) + \mu x_i$. The solution is (see Exercises 50 and 51):

$$x_i^*(\mu) = \left[\frac{1}{\mu} - \frac{1}{\beta_i} \right]_0^\infty \quad \forall i \in \{1, \dots, n\} \quad (76)$$

where $[z]_0^\infty = \max[z, 0]$ denotes a projection of the real number z onto the interval $[0, \infty)$.

By Theorem 18, we want to find a value $\mu > 0$ to solve

$$\sum_{i=1}^n \left[\frac{1}{\mu} - \frac{1}{\beta_i} \right]_0^\infty = \theta \quad (77)$$

The function $h(1/\mu) = \sum_{i=1}^n \left[\frac{1}{\mu} - \frac{1}{\beta_i} \right]_0^\infty$ is continuous and nondecreasing in the parameter $1/\mu$ and satisfies

$$\begin{aligned} h(1/\mu) &= 0 \quad \text{if } 1/\mu \leq 1/\beta_{\max} \\ \lim_{1/\mu \rightarrow \infty} h(1/\mu) &= \infty \end{aligned}$$

where $\beta_{\max} = \max\{\beta_1, \dots, \beta_n\}$. The intermediate value theorem ensures that for every real number $\theta > 0$, there is a value $\mu > 0$ such that $h(1/\mu) = \theta$. There are various ways to find such a $1/\mu \in [1/\beta_{\max}, \infty)$ (one method is bisection).

It is often useful to assume the channels are arranged in order of channel quality, so

$$\beta_{\max} = \beta_1 \geq \beta_2 \geq \dots \geq \beta_n > 0 \quad (78)$$

Then if $1/\mu = 1/\beta_3$, equality (76) yields $x_i^*(\mu) = 0$ for all $i \geq 3$ and

$$\begin{aligned} x_1^*(\mu) &= \frac{1}{\beta_3} - \frac{1}{\beta_1} \\ x_2^*(\mu) &= \frac{1}{\beta_3} - \frac{1}{\beta_2} \\ x_i^*(\mu) &= 0 \quad \forall i \in \{3, \dots, n\} \end{aligned}$$

and this vector $x^*(\mu)$ solves the constrained problem (73)-(75) for the special case

$$\theta = \left(\frac{1}{\beta_3} - \frac{1}{\beta_1} \right) + \left(\frac{1}{\beta_3} - \frac{1}{\beta_2} \right)$$

Further, this value of θ is the largest for which an optimal solution has $x_3^* = 0$. If total power exceeds $\left(\frac{1}{\beta_3} - \frac{1}{\beta_1} \right) + \left(\frac{1}{\beta_3} - \frac{1}{\beta_2} \right)$ then we must start placing some power on channel 3.

In general, if the β_i values are ordered according to (78) then there is some integer $k \in \{1, \dots, n\}$ such that $1/\mu \geq 1/\beta_i$ for all $i \in \{1, \dots, k\}$ but $1/\mu < 1/\beta_i$ for $i > k$. In that case we have $x_i^* = 0$ for all $i > k$ and we have from (77):

$$\sum_{i=1}^k \left(\frac{1}{\mu} - \frac{1}{\beta_i} \right) = \theta \implies \frac{1}{\mu} = \frac{\theta}{k} + \frac{1}{k} \sum_{i=1}^k \frac{1}{\beta_i}$$

which implies $x_i^* = 0$ for all $i > k$ and (from (76))

$$x_i^* = \left(\frac{\theta}{k} + \frac{1}{k} \sum_{j=1}^k \frac{1}{\beta_j} \right) - \frac{1}{\beta_i} \quad \forall i \in \{1, \dots, k\}$$

Of course, we do not know which $k \in \{1, \dots, n\}$ to use, so we must find an integer k (which exists) such that when we define $1/\mu = \frac{\theta}{k} + \frac{1}{k} \sum_{i=1}^k 1/\beta_i$, we have $1/\mu \geq 1/\beta_i$ for all $i \leq k$ and $1/\mu < 1/\beta_i$ for all $i > k$.

For these two examples, we were “lucky” to always be able to find a Lagrange multiplier $\mu \geq 0$ such that $\sum_{i=1}^n x_i^*(\mu) = \theta$. This is because the original constrained problem for both examples was a convex optimization problem. Exercises 52-57 also consider Lagrange multipliers for convex problems. For nonconvex problems, the Lagrange multiplier theorem still holds, but we may not be able to find Lagrange multiplier vectors μ, λ that lead to the desired constants c, d for our constrained problem.

D. Dual function

Fix n, k, m as nonnegative integers (assume $n > 0, k + m > 0$). Fix $X \subseteq \mathbb{R}^n$ as a nonempty set (possibly nonconvex). Fix $f : X \rightarrow \mathbb{R}$, $g_i : X \rightarrow \mathbb{R}$ for $i \in \{1, \dots, k\}$, $h_j : X \rightarrow \mathbb{R}$ for $j \in \{1, \dots, m\}$ as some real-valued functions (possibly nonconvex). Consider the following constrained optimization problem:

$$\text{Minimize: } f(x) \quad (79)$$

$$\text{Subject to: } g_i(x) \leq 0 \quad \forall i \in \{1, \dots, k\} \quad (80)$$

$$h_j(x) = 0 \quad \forall j \in \{1, \dots, m\} \quad (81)$$

$$x \in X \quad (82)$$

For convenience, this problem has the c_i and d_j constants all set to 0. So, for example, an inequality constraint of the form $\cos(x_1 + x_2) + x_4^2 \leq 8$ is modeled by $\cos(x_1 + x_2) + x_4^2 - 8 \leq 0$. Define the set $D \subseteq \mathbb{R}^{k+m}$ by

$$D = \{(\mu, \lambda) : \mu \in \mathbb{R}^k, \lambda \in \mathbb{R}^m, \mu \geq 0\}$$

Define the *dual function* $d : D \rightarrow \mathbb{R} \cup \{-\infty\}$ for the problem (79)-(82) by

$$d(\mu, \lambda) = \inf_{x \in X} \{f(x) + \mu^\top g(x) + \lambda^\top h(x)\} \quad \forall (\mu, \lambda) \in D$$

where for convenience we define vector-valued functions $g(x)$ and $h(x)$ by

$$g(x) = (g_1(x), \dots, g_k(x))$$

$$h(x) = (h_1(x), \dots, h_m(x))$$

The dual function is simply the optimal objective value of the unconstrained problem (66)-(67) treated as a function of (μ, λ) .

Theorem 19: (Weak duality) Suppose x^* is an optimal solution to problem (79)-(82). Then

$$d(\mu, \lambda) \leq f(x^*) \quad \forall (\mu, \lambda) \in D$$

and so

$$\sup_{(\mu, \lambda) \in D} \{d(\mu, \lambda)\} \leq f(x^*)$$

Proof: Fix $(\mu, \lambda) \in D$. Since $x^* \in X$, we have by definition of $\inf_{x \in X}$:

$$\begin{aligned} d(\mu, \lambda) &= \inf_{x \in X} \{f(x) + \mu^\top g(x) + \lambda^\top h(x)\} \\ &\stackrel{(a)}{\leq} f(x^*) + \mu^\top g(x^*) + \lambda^\top h(x^*) \\ &\stackrel{(b)}{=} f(x^*) + \mu^\top g(x^*) \\ &\stackrel{(c)}{\leq} f(x^*) \end{aligned}$$

where (a) holds by definition of $\inf_{x \in X}$; (b) holds because $h(x^*) = 0$ (recall that x^* satisfies the constraints (80) and (81)); (c) holds because $\mu \geq 0$ and $g(x^*) \leq 0$. $\square$

We define the *duality gap* for the problem (79)-(82) by

$$\text{duality gap} = f(x^*) - \sup_{(\mu, \lambda) \in D} \{d(\mu, \lambda)\}$$

The weak duality theorem above ensures the duality gap is always greater than or equal to zero (regardless of whether or not the problem (79)-(82) is convex). Unfortunately, in most cases of nonconvex problems the duality gap is strictly positive (there are some special cases of nonconvex problems with zero duality gap). Fortunately, for most convex problems, the duality gap is zero. A famous condition called the *Slater condition* gives a sufficient condition for a convex problem to have zero duality gap (see [2]). As a special case, it can be shown that all linear programs that have optimal solutions also have zero duality gap. A problem that has zero duality gap is said to satisfy *strong duality*.

E. Dual subgradient algorithm

Weak duality implies that $d(\mu, \lambda)$ is a lower bound on $f(x^*)$ for the problem (79)-(82). The best lower bound of this type is obtained by maximizing the dual function. Fortunately, the next lemma ensures the dual function is always concave. Further, a subgradient can often be computed. Thus, maximizing the dual function can be accomplished by projected gradient ascent.

Lemma 18: For all (possibly nonconvex) problems, the dual function is concave. Further, at a given $(\mu, \lambda) \in D$, if $\hat{x}$ is a minimizer of $f(x) + \mu^\top g(x) + \lambda^\top h(x)$ over all $x \in X$, then $(g(\hat{x}), h(\hat{x})) \in \mathbb{R}^{k+m}$ is a subgradient of the dual function at (μ, λ) , meaning (since d is concave rather than convex):

$$d(\tilde{\mu}, \tilde{\lambda}) \leq d(\mu, \lambda) + [g(\hat{x}), h(\hat{x})]^\top ((\tilde{\mu}, \tilde{\lambda}) - (\mu, \lambda)) \quad \forall (\tilde{\mu}, \tilde{\lambda}) \in D$$

Proof: See Exercises 58 and 59. □

This means that to find the best Lagrange multiplier vector (μ^*, λ^*) , we can use either projected subgradient descent on the convex function $-d$, or projected subgradient *ascent* on the concave function d . Using a fixed stepsize $\delta > 0$ and operating directly on the d function, we initialize $(\mu(0), \lambda(0)) \in D$ and do

$$(\mu(t+1), \lambda(t+1)) = \pi_D[(\mu(t), \lambda(t)) + \delta(g(x(t)), h(x(t)))] \quad \forall t \in \{0, 1, 2, \dots\}$$

where $x(t) \in X$ is a minimizer of $f(x) + \mu(t)^\top g(x) + \lambda(t)^\top h(x)$ over all $x \in X$. Since D is a Cartesian product of closed intervals, the projection onto D is trivial and leads to the famous *dual subgradient algorithm* for problem (79)-(82):

Dual subgradient algorithm: (fixed stepsize $\delta > 0$)

- 1) Fix $(\mu(0), \lambda(0)) \in D$ (such as $(\mu(0), \lambda(0)) = 0$).
- 2) Every step $t \in \{0, 1, 2, \dots\}$ do

- Define $x(t) \in X$ as a minimizer of

$$f(x) + \mu(t)^\top g(x) + \lambda(t)^\top h(x)$$

over all $x \in X$ (assuming a minimizer exists).

- Update the Lagrange multiplier iterates by

$$\begin{aligned} \mu_i(t+1) &= \max[\mu_i(t) + \delta g_i(x(t)), 0] \quad \forall i \in \{1, \dots, k\} \\ \lambda_j(t+1) &= \lambda_j(t) + \delta h_j(x(t)) \quad \forall j \in \{1, \dots, m\} \end{aligned}$$

The algorithm only applies when a minimizer $x(t) \in X$ can be found for each t . This can be ensured, for example, when X is a compact set and the f, g, h functions are continuous. The above algorithm uses a fixed stepsize $\delta > 0$. Of course, a vanishing stepsize can also be used, such as $\delta(t) = 1/\sqrt{t+1}$. Exercises 60 and 61 consider the dual subgradient algorithm for a convex program with two variables, and a linear program with n variables, respectively. Exercises 62 and 63 consider the dual subgradient algorithm for network utility maximization.

Strictly speaking, the dual subgradient algorithm is designed only to find optimal Lagrange multipliers, rather than to solve the problem (79)-(82) of interest. However, its iterations can be shown to lead to many desirable outcomes. A stochastic version of the dual subgradient algorithm for time-varying objective and constraint functions is known as the *drift-plus-penalty* algorithm [15]. For (deterministic) convex optimization problems with strongly convex objective functions f , the work [16][17] shows the dual subgradient algorithm yields iterates such that $\frac{1}{T} \sum_{t=0}^{T-1} x(t)$ converges to optimality in the sense that

$$f\left(\frac{1}{T} \sum_{t=0}^{T-1} x(t)\right) \leq f(x^*) + O(1/T) \tag{83}$$

$$g_i\left(\frac{1}{T} \sum_{t=0}^{T-1} x(t)\right) \leq O(1/T) \quad \forall i \in \{1, \dots, k\} \tag{84}$$

A slightly modified algorithm with similar structure to the dual subgradient algorithm is developed in [18] that achieves (83)-(84) for general (possibly nonsmooth) convex programs without requiring the objective function f to be strongly convex. Primal-dual versions of these algorithms also exist when f, g are smooth (these used linearized approximations of f and g based on their gradients). A Newton method that applies for twice-differentiable functions uses Hessians to make progress in a much smaller number of iterations, but each iteration involves a more complicated matrix inversion [2].

F. Dual function for a linear program

Consider the linear program that seeks $x = (x_1, \dots, x_n)$ to solve

$$\text{Minimize: } c^\top x \quad (85)$$

$$\text{Subject to: } Ax \leq b \quad (86)$$

$$x \geq 0 \quad (87)$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^k$, and A is a $k \times n$ matrix. Define $f(x) = c^\top x$, $g(x) = Ax - b$, $X = \{x \in \mathbb{R}^n : x_i \geq 0 \ \forall i\}$. Since there are no equality constraints, the dual function is defined over $\mu \in \mathbb{R}^k$ with $\mu \geq 0$ by

$$d(\mu) = \inf_{x \in X} \{c^\top x + \mu^\top (Ax - b)\} \quad \forall \mu \geq 0$$

We can explicitly compute $d(\mu)$. Fix $\mu \geq 0$. Then

$$\begin{aligned} d(\mu) &= \inf_{x \geq 0} \{(c^\top + \mu^\top A)x - \mu^\top b\} \\ &= \inf_{x \geq 0} \{q^\top x - \mu^\top b\} \end{aligned}$$

where we define $q = (q_1, \dots, q_n)$ by $q^\top = c^\top + \mu^\top A$. Thus

$$d(\mu) = \inf_{x \geq 0} \left\{ \sum_{i=1}^n q_i x_i - \mu^\top b \right\} = \begin{cases} -\mu^\top b & \text{if } q_i \geq 0 \ \forall i \in \{1, \dots, n\} \\ -\infty & \text{else} \end{cases}$$

which holds because if $q_i \geq 0$ for all i , then the infimum is achieved by using $x_i = 0$ for all i , while if there is at least one i such that $q_i < 0$ then we can let $x_i \rightarrow \infty$ to show the infimum diverges to $-\infty$. The definition of q means $q = c + A^\top \mu$. The condition $q_i \geq 0$ for all $i \in \{1, \dots, n\}$ is equivalent to the vector inequality $A^\top \mu \geq -c$. Note that $\mu^\top b = b^\top \mu$. Thus, the dual function for problem (85)-(87) is

$$d(\mu) = \begin{cases} -b^\top \mu & \text{if } A^\top \mu \geq -c \\ -\infty & \text{else} \end{cases} \quad \forall \mu \geq 0 \quad (88)$$

Assume the linear program (85)-(87) is feasible and has optimal solution $x^* \in \mathbb{R}^n$. Since strong duality holds for all linear programs that are feasible, we know that maximizing the dual function yields $\sup_{\mu \geq 0} \{d(\mu)\} = c^\top x^*$. Let us find a $\mu \geq 0$ that maximizes $d(\mu)$ in (88). We immediately see that we must choose a $\mu \geq 0$ to satisfy $A^\top \mu \geq -c$ (else $d(\mu) = -\infty$, which cannot be the maximum). Subject to the constraints $A^\top \mu \geq -c$ and $\mu \geq 0$, we seek a μ that maximizes $-b^\top \mu$. So we find a $\mu = (\mu_1, \dots, \mu_k)$ that solves the following linear program:

$$\text{Maximize: } -b^\top \mu \quad (89)$$

$$\text{Subject to: } A^\top \mu \geq -c \quad (90)$$

$$\mu \geq 0 \quad (91)$$

The problem 89-(91) is called the *dual linear program*; the problem (85)-(87) is called the *primal linear program*. Strong duality means that if μ^* is an optimal solution to the dual linear program then $-b^\top \mu^* = c^\top x^*$. The primal linear program has n variables and k inequality constraints. The dual linear program has k variables and n inequality constraints.

An example is shown below. The two separate linear programs have been solved in matlab. It can be seen that their optimal objective values are the same.

- Primal LP: [Solution: $objective^* = -10$, $x^* = (6, 0, 0, 4)$]

$$\begin{aligned} \text{Minimize: } & \begin{bmatrix} 1 & 0 & 2 & -4 \end{bmatrix} x \\ \text{Subject to: } & \begin{bmatrix} 1 & 1 & 0 & -1 \\ -1 & 0 & 3 & 2 \end{bmatrix} x \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\ & x \geq 0 \end{aligned}$$

- Dual LP: [Solution: $objective^* = -10$, $\mu^* = (2, 3)$]

$$\begin{aligned} \text{Maximize: } & - \begin{bmatrix} 2 & 2 \end{bmatrix} \mu \\ \text{Subject to: } & \begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 0 & 3 \\ -1 & 2 \end{bmatrix} \mu \geq - \begin{bmatrix} 1 \\ 0 \\ 2 \\ -4 \end{bmatrix} \\ & \mu \geq 0 \end{aligned}$$

The maximization of the dual $d(\mu)$ can be posed as a minimization of $-d(\mu)$. Then, for this linear program structure, it can be shown that the dual of the dual yields the original primal linear program. It can also be shown that

- 1) The primal LP has an optimal solution if and only if the dual LP has an optimal solution.
- 2) If feasible, the primal and dual LPs have the same optimal objective value.

Suppose we solve the dual LP and obtain an optimal solution μ^* . Does that enable us to easily find an optimal solution x^* to the primal LP? Unfortunately, no. While it tells us what the optimal objective value will be for the primal LP, it does not tell us what vector x^* solves it. One might attempt to find x^* from the optimal Lagrange multiplier vector μ^* by minimizing $c^\top x + \mu^{*\top}(Ax - b)$ subject to $x \geq 0$. While the optimal solution x^* is indeed a minimizer of $c^\top x + \mu^{*\top}(Ax - b)$ over all $x \geq 0$ (see Exercise 69), there can be infinitely many other minimizers, many of which do not even satisfy the desired constraints of the primal LP.

Exercises 65 and 66 consider example primal-dual LP pairs similar to the above. Exercise 67 computes the dual function $d(\mu, \lambda)$ for the linear program with the following structure:

$$\text{Minimize: } c^\top x \quad (92)$$

$$\text{Subject to: } Ax \leq b \quad (93)$$

$$Hx = d \quad (94)$$

$$x \in \mathbb{R}^n \quad (95)$$

and shows that maximizing $d(\mu, \lambda)$ over all $(\mu, \lambda) \in D$ leads to another dual linear program. Exercise 68 treats a different linear program structure that is closely related to problem (92)-(95).

G. A comment on utilizing optimal Lagrange multipliers

In general, if the (possibly nonconvex) problem (79)-(82) has an optimal solution x^* , and if this problem satisfies strong duality, and if the dual problem of maximizing $d(\mu, \lambda)$ over all $(\mu, \lambda) \in D$ has an optimal solution (μ^*, λ^*) (called an optimal Lagrange multiplier solution), one can say that $d(\mu^*, \lambda^*) = f(x^*)$ and

$$f(x) + \mu^{*\top}g(x) + \lambda^{*\top}h(x) \geq f(x^*) \quad \forall x \in X$$

with equality when $x = x^*$. One may attempt to find an optimal solution to the original problem (79)-(82) by using the optimal Lagrange multiplier vector (μ^*, λ^*) to solve:

$$\text{Minimize: } f(x) + \mu^{*\top}g(x) + \lambda^{*\top}h(x) \quad (96)$$

$$\text{Subject to: } x \in X \quad (97)$$

However, while it can be shown that all optimal solutions x^* to the original problem (79)-(82) are also optimal solutions to problem (96)-(97) (see Exercise 69), there can be many other optimal solutions to (96)-(97), many of which do not even satisfy the constraints of the original problem. We cannot expect to find an optimal solution to the original problem by solving (96)-(97). On the other hand, if X is a convex set and if the objective function in (96) is *strictly convex* over all $x \in X$, then problem (96)-(97) has at most one optimal solution. In this case, since we already know x^* solves (96)-(97), x^* must be the *unique solution* to (96)-(97). This is a situation when solving (96)-(97) indeed leads to an optimal solution x^* to the original problem of interest. This situation does not apply to linear programs because, even though linear programs are convex, they do not have the *strictly convex* property.

Nevertheless, the *concept* of duality and Lagrange multipliers is still useful, particularly for convex programs and linear programs, even though strict convexity may not hold. For example, the dual subgradient algorithm, and its faster variation in [18], both yield time averages of iterates $x(t)$ that do approach an optimal solution to the original constrained problem of interest for general convex programs, regardless of whether or not strict convexity holds.

XI. EXERCISES

Exercise 1: (The L_∞ norm is not strictly convex) Let $n \geq 2$ be an integer. We consider the L_∞ norm over $x \in \mathbb{R}^n$:

$$\|x\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\} \quad \forall x \in \mathbb{R}^n$$

- a) Show that $\|x\|_\infty$ is not strictly convex.
- b) Show that $\|x\|_\infty^2$ is not strictly convex.
- c) A norm $\|\cdot\|$ on $\mathbb{R}^n$ is an *inner product norm* if and only if $\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2$ for all $x, y \in \mathbb{R}^n$ (this equality is called the *parallelogram law*). Show that $\|x\|_\infty$ is not an inner product norm.

Exercise 2: (Jensen's inequality for convex functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. Let Y be a random vector that takes values in $\mathbb{R}^n$ and has finite expectation $\mathbb{E}[Y]$. Define $m = \mathbb{E}[Y]$.

a) Use the fact that $\partial f(m)$ is nonempty (Theorem 3) to conclude there is a subgradient $f'(m) \in \partial f(m)$ and so

$$f(Y) \geq f(m) + f'(m)^\top (Y - m)$$

b) Take expectations of part (a) to prove Jensen's inequality.

Exercise 3: (General Jensen's inequality) Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set and let $f : X \rightarrow \mathbb{R}$ be a convex function. Suppose Y is a random vector that takes values in X and has finite expectation $\mathbb{E}[Y]$. Define $m = \mathbb{E}[Y]$. It can be shown that every nonempty convex set X has a nonempty relative interior.

a) Let v be a point in the relative interior of X . Use Theorem 4 to conclude there is a $f'(v) \in \partial f(v)$ and so

$$f(Y) \geq f(v) + f'(v)^\top (Y - v)$$

b) Take expectations of part (a) to prove that $\mathbb{E}[f(Y)]$ is lower bounded by a finite constant. Hence, $\mathbb{E}[f(Y)]$ must exist (possibly being ∞) and cannot be $-\infty$.

c) It can be shown that if k is a positive integer, $D \subseteq \mathbb{R}^k$ is a convex set, and Z is a random vector that takes values in D and that has finite expectation $\mathbb{E}[Z]$, then $\mathbb{E}[Z] \in D$. We want to use this to prove Jensen's inequality: $\mathbb{E}[f(Y)] \geq f(\mathbb{E}[Y])$. If $\mathbb{E}[f(Y)] = \infty$ the result is trivially true. Part (b) implies $\mathbb{E}[f(Y)] \neq -\infty$. Now suppose $\mathbb{E}[f(Y)]$ is finite. Define $D = \{(x, y) \in \mathbb{R}^{n+1} : x \in X, y \in \mathbb{R}, y \geq f(x)\}$. Show D is a convex set. Define $Z = (X, f(X))$ and observe $Z \in D$. Use this to prove Jensen's inequality.

Exercise 4: Fix $X \subseteq \mathbb{R}^n$ as a nonempty convex set. Let $f : X \rightarrow \mathbb{R}$ and $g : X \rightarrow \mathbb{R}$ be convex functions. Fix $a, b \geq 0$. Define $h : X \rightarrow \mathbb{R}$ by

$$h(x) = af(x) + bg(x) \quad \forall x \in X$$

Fix $v \in X$ and suppose $f'(v) \in \partial f(v)$ and $g'(v) \in \partial g(v)$. Define $h'(v) = af'(v) + bg'(v)$. Show that $h'(v) \in \partial h(v)$.

Exercise 5: Let $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ be a convex function. For each $(x, y) \in \mathbb{R}^2$, let $f'(x, y)$ denote a subgradient of f at (x, y) . Define $g : \mathbb{R}^2 \rightarrow \mathbb{R}$ by $g(x, y) = 3f(x, y) + \cos(x + y^2)$. For each $(x, y) \in \mathbb{R}^2$, find a subgradient $g'(x, y)$.

Exercise 6: Fix $X \subseteq \mathbb{R}^n$ as a nonempty convex set. Let $f_i : X \rightarrow \mathbb{R}$ be convex functions for each $i \in \{1, \dots, k\}$. Define $h : X \rightarrow \mathbb{R}$ by

$$h(x) = \max\{f_1(x), \dots, f_k(x)\} \quad \forall x \in X$$

Fix $v \in X$ and suppose $f'_i(v) \in \partial f_i(v)$ for all $i \in \{1, \dots, k\}$. Define $h'(v)$ by $h'(v) = f'_{j(v)}(v)$, where $j(v)$ is any element of $\arg \max_{i \in \{1, \dots, k\}} \{f_i(v)\}$. Show that $h'(v) \in \partial h(v)$.

Exercise 7: Prove Lemma 2.

Exercise 8: Define the nonsmooth convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$f(x) = \exp(x_3 + x_4) + \sum_{i=1}^n w_i |x_i - a_i| \quad \forall x \in \mathbb{R}^n$$

where $n \geq 4$ is an integer, $a_1, \dots, a_n, w_1, \dots, w_n$ are real numbers, $w_i \geq 0$ for all $i \in \{1, \dots, n\}$. Find a subgradient $f'(x)$ for each $x \in \mathbb{R}^n$.

Exercise 9: Define the nonsmooth convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$f(x) = \|Ax - b\|_2 + \sum_{i=1}^n e^{c_i x_i} \quad \forall x \in \mathbb{R}^n$$

where k, n are positive integers, $c_1, \dots, c_n$ are given real numbers, A is a real $k \times n$ matrix, $b \in \mathbb{R}^k$. Find a subgradient $f'(x)$ for each $x \in \mathbb{R}^n$.

Exercise 10: Define the nonsmooth convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$f(x) = b^\top x + \max\{\|x - a\|_2, c^\top x + d\} \quad \forall x \in \mathbb{R}^n$$

where $a, b, c \in \mathbb{R}^n$ and $d \in \mathbb{R}$. Find a subgradient $f'(x)$ for each $x \in \mathbb{R}^n$.

Exercise 11: Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. Let x^* be a minimizer of f over all $x \in \mathbb{R}^n$. Suppose we test a point $w \in \mathbb{R}^n$ and the oracle returns $f'(w) \in \partial f(w)$. Suppose $f'(w) \neq 0$. Show that we can ensure x^* lies in a particular halfspace of $\mathbb{R}^n$ defined by a hyperplane that passes through w . Specifically, show that

$$f'(w)^\top (x^* - w) \leq 0$$

That is, the ideal-but-unknown direction $(x^* - w)$ has negative (or zero) inner product with the known direction $f'(w)$. This suggests that taking a small step away from w in the direction of $-f'(w)$ is a good idea.

Exercise 12: (How many iterations are needed?) Suppose we use subgradient descent with a fixed stepsize $\delta > 0$ under the conditions of Theorem 8. Suppose we know $B = 1$. We also choose a point $x_0 \in C$ that ensures $\|x_0 - x\|^2 \leq r^2$ for all $x \in C$, where r is some known positive constant. We use $\delta = \frac{1}{2 \cdot 10^3}$. How many iterations m are needed to ensure we can compute a vector $\tilde{x}_m \in C$ such that $f(\tilde{x}_m) \leq f(x^*) + \frac{1}{10^3}$?

Exercise 13: (Bounds for portfolio optimization) For online portfolio optimization: For our $g_k(x)$ function defined over $x \in P$, where $P = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x_i \geq 0 \ \forall i \in \{1, \dots, n\}\}$, show that for all steps $k \in \{0, 1, 2, \dots\}$ we have

$$\|\nabla g_k(x)\| \leq \frac{q_{\max} \sqrt{n}}{q_{\min}} \quad \forall x \in P$$

and so we have B -bounded gradients with $B = \frac{q_{\max} \sqrt{n}}{q_{\min}}$.

Exercise 14: (Fixed decisions for online portfolio) Consider online portfolio optimization with one stock and a savings account. So

- $x(k) \in [0, 1]$ is the fraction of wealth we put into the stock at time k .
- $1 - x(k) \in [0, 1]$ is the fraction of wealth we put into the savings account at time k .
- $q(k)$ = stock price at time k .
- $g_k(x(k)) = \log(x(k)q(k) + 1 \cdot (1 - x(k)))$ for $k \in \{0, 1, 2, \dots\}$.
- $0 < q_{\min} \leq q(k) \leq q_{\max}$ for $k \in \{0, 1, 2, \dots\}$.

For positive integers m define the convex function of one variable:

$$h_m(x) = \frac{1}{m} \sum_{k=0}^{m-1} \log(xq(k) + 1 - x) \quad \forall x \in [0, 1]$$

Define $h_m^* = \max\{x \in [0, 1] \mid h_m(x)\}$. Online portfolio optimization guarantees a growth exponent over the first m steps that is at least as good as h_m^* (minus some small error terms). We want to find x_m^* , the maximizer of h_m^* .

- Show that if $h'_m(0) \leq 0$ then $x_m^* = 0$ and $h_m^* = 0$.
- Show that if $\frac{1}{m} \sum_{k=0}^{m-1} q(k) \leq 1$ then $x_m^* = 0$ and $h_m^* = 0$.
- Argue that if $\frac{1}{m} \sum_{k=0}^{m-1} 1/q(k) \leq 1$ then $x_m^* = 1$.
- Argue that if both $\frac{1}{m} \sum_{k=0}^{m-1} q(k) > 1$ and $\frac{1}{m} \sum_{k=0}^{m-1} 1/q(k) > 1$ then $0 < x_m^* < 1$. Demonstrate a sequence of positive real numbers $\{\theta(0), \theta(1), \theta(2), \dots\}$ that satisfy this (where $q(k) = \theta(k+1)/\theta(k)$).
- Argue that h_m is strictly concave if and only if there is at least one $k \in \{0, \dots, m-1\}$ such that $q(k) \neq 1$.

Exercise 15: Change the online portfolio optimization algorithm to utilize mirror descent on the probability simplex P .

Exercise 16: (Projecting onto a probability simplex) Fix $w = (w_1, w_2)$ such that $w_1 \geq 0, w_2 \geq 0$. Define the probability simplex $P = \{(x_1, x_2) : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 = 1\}$. Find $\pi_P(w)$ as a function of (w_1, w_2) .

Exercise 17: (Summing strongly convex functions) Prove Lemma 13.

Exercise 18: Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. Let $f : X \rightarrow \mathbb{R}$ be a convex function. Fix $c \geq 0$ and $w \in \mathbb{R}^n$ and define the function $h : X \rightarrow \mathbb{R}$ by

$$h(x) = f(x) + \frac{c}{2} \|x - w\|^2 \quad \forall x \in X$$

Use Lemma 10 to prove that h is c -strongly convex.

Exercise 19: (Unique minimizer) Let $f : X \rightarrow \mathbb{R}$ be a strictly convex function. Suppose $x^* \in X$ is a minimizer of f . Show that x^* is the unique minimizer of f .

Exercise 20: Let $f : X \rightarrow \mathbb{R}$ be a convex function. Fix $x, y \in X$ and suppose $f'(x) \in \partial f(x)$ and $f'(y) \in \partial f(y)$.

a) Use the standard subgradient inequality to show

$$(f'(x) - f'(y))^\top (x - y) \geq 0$$

b) Suppose f is c -strongly convex for a given $c \geq 0$. Use the strengthened subgradient inequality to show

$$(f'(x) - f'(y))^\top (x - y) \geq c\|x - y\|^2$$

Exercise 21: (Objective bound for strongly convex case) Let f satisfy the conditions of Theorem 17, so f is c -strongly convex and β -smooth (for some $c > 0$, $\beta > 0$). Assume the stepsize satisfies $0 < \delta \leq 1/\beta$ and the projected gradient descent algorithm of Theorem 17 is used.

a) Use the descent lemma (Lemma 17) to show that for all $k \in \{0, 1, 2, \dots\}$ we have

$$f(x_k) \leq f(x^*) + \rho^k \|\nabla f(x^*)\| \cdot \|x^* - x_0\| + \frac{\rho^{2k}\beta}{2} \|x^* - x_0\|^2$$

b) Suppose $X = \mathbb{R}^n$. Use Lemma 1 and Theorem 3c to argue that $\nabla f(x^*) = 0$, and

$$f(x_k) \leq f(x^*) + \frac{\rho^{2k}\beta}{2} \|x^* - x_0\|^2$$

Exercise 22: Fix $c \geq 0$, $\beta \geq 0$. Suppose $X \subseteq \mathbb{R}^n$ is a convex set that consists of more than one point. Suppose $f : X \rightarrow \mathbb{R}$ is c -strongly convex and β -smooth. Show that $c \leq \beta$. Hint: Use the strengthened subgradient inequality.

Exercise 23: (Second derivative condition) Suppose $f : \mathbb{R} \rightarrow \mathbb{R}$ is twice differentiable. Fix $c \geq 0$. Use Lemma 10 to show that f is c -strongly convex if and only if $f''(x) \geq c$ for all $x \in \mathbb{R}$.

Exercise 24: (Proving one direction of Lemma 10) Fix $X \subseteq \mathbb{R}^n$ as a convex set. Fix $c \geq 0$. Suppose $f : X \rightarrow \mathbb{R}$ is c -strongly convex. Use (38) to prove the function $h : X \rightarrow \mathbb{R}$ defined by $h(x) = f(x) - \frac{c}{2}\|x\|^2$ for all $x \in X$ is convex.

Exercise 25: (Proving the other direction of Lemma 10) Fix $X \subseteq \mathbb{R}^n$ as a convex set. Let $f : X \rightarrow \mathbb{R}$ be some function. Fix $c \geq 0$ and define $h : X \rightarrow \mathbb{R}$ by $h(x) = f(x) - \frac{c}{2}\|x\|^2$ for all $x \in X$. Show that if h is convex, then f is c -strongly convex. Hint: Use (38).

Exercise 26: Fix $a \in \mathbb{R}$ and $\beta > 0$. Show that if $f : X \rightarrow \mathbb{R}$ is defined by $f(x) = \frac{\beta}{2}\|x - a\|^2$ then

a) f is β -smooth and β -strongly convex.

b) The algorithm of Theorem 17 finds the exact x^* in one step $k = 1$.

Exercise 27: (Scaling smooth functions) Show that if $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is β -smooth for some $\beta > 0$, then for any $w \geq 0$, any $a \in \mathbb{R}^n$, any $b \in \mathbb{R}$, the function $wf + a^\top x + b$ is $(w\beta)$ -smooth.

Exercise 28: (Projections shrink distances) This problem proves Lemma 8. Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set. Fix $x, y \in \mathbb{R}^n$. We want to prove Lemma 8 to show that $\|x - y\| \geq \|\pi_C(x) - \pi_C(y)\|$, where we use the Euclidean norm.

a) Use the normality condition (Theorem 6) to show

$$\begin{aligned} (x - \pi_C(x))^\top (\pi_C(y) - \pi_C(x)) &\leq 0 \\ (y - \pi_C(y))^\top (\pi_C(x) - \pi_C(y)) &\leq 0 \end{aligned}$$

b) Observe that

$$\begin{aligned} \|x - y\|^2 &= \|(x - \pi_C(x)) + \pi_C(x) - \pi_C(y) + (\pi_C(y) - y)\|^2 \\ &= \|\pi_C(x) - \pi_C(y)\|^2 + \|(x - \pi_C(x)) + (\pi_C(y) - y)\|^2 + 2(\pi_C(x) - \pi_C(y))^\top ((x - \pi_C(x)) + (\pi_C(y) - y)) \end{aligned}$$

c) Combine (a) and (b) to prove the result.

Exercise 29: Fix $n \geq 2$ and $d \in \mathbb{R}^n$ with $d \neq 0$. Define $f : \mathbb{R}^n \rightarrow \mathbb{R}$ by

$$f(x) = e^{d^\top x} \quad \forall x \in \mathbb{R}^n$$

a) Show that $\nabla f(x) = df(x)$ for all $x \in \mathbb{R}^n$.

b) Show that the Hessian is $\nabla^2 f(x) = dd^\top f(x)$ for all $x \in \mathbb{R}^n$.

c) Show the Hessian is positive semidefinite, but not positive definite. Hence, f is not strongly convex.

d) For $x, y \in \mathbb{R}^n$ show that $\|\nabla f(x) - \nabla f(y)\| = \|d\| \cdot |f(x) - f(y)|$. Argue that there is no $\beta > 0$ such that f is β -smooth over all $\mathbb{R}^n$. However, it can be shown that f is β -smooth over $x \in [-1, 1]^n$ with $\beta = \|d\|^2 e^{\|d\|_1}$, where $\|d\|_1$ is the L_1 norm.

Exercise 30: (At boundary points the subdifferential is either empty or unbounded) Let $X \subseteq \mathbb{R}^n$ be a convex set and $f : X \rightarrow \mathbb{R}$ a convex function. Fix $w \in X$ and suppose: (i) w is a boundary point of X ; (ii) $\partial f(w)$ is nonempty. That is, there is at least one subgradient $f'(w) \in \partial f(w)$. Show that $\partial f(w)$ is unbounded. Hint: Use the hyperplane separation theorem to show there is a hyperplane that passes through w and contains X on one side. Use this to augment the $f'(w)$ subgradient.

Exercise 31: ($\partial f(x)$ for a piecewise linear $f : \mathbb{R} \rightarrow \mathbb{R}$) Define the convex function $f : \mathbb{R} \rightarrow \mathbb{R}$ by

$$f(x) = \max\{-2x + 3, 4x + 1, 5x\} \quad \forall x \in \mathbb{R}$$

Find $\partial f(x)$ for all $x \in \mathbb{R}$.

Exercise 32: (When subgradient does not exist) Define the convex function $f : [0, \infty) \rightarrow \mathbb{R}$ by

$$f(x) = -\sqrt{x} \quad \forall x \geq 0$$

a) Give the set $\partial f(x)$ for all $x > 0$.

b) Show there is no $s \in \mathbb{R}$ such that $s \in \partial f(0)$. Thus, $\partial f(0) = \emptyset$. Hint: Suppose there is an $s \in \partial f(0)$. Write the subgradient inequality and get a contradiction.

Exercise 33: (Subgradient of $\|x\|_1$) Fix n as a positive integer. Let $g_i : \mathbb{R} \rightarrow \mathbb{R}$ be a convex functions with subgradients $g'_i(x) \in \partial g_i(x)$ for all $x \in \mathbb{R}$ and all $i \in \{1, \dots, n\}$.

a) Define $f(x) = \sum_{i=1}^n g_i(x_i)$ for all $x \in \mathbb{R}^n$. Define $f'(x) = \sum_{i=1}^n g'_i(x_i)e_i$ where $e_i \in \mathbb{R}^n$ is a unit vector that is 1 in entry i and 0 in all other entries. Show that $f'(x) \in \partial f(x)$ for all $x \in \mathbb{R}^n$.

b) Define the convex function $f(x) = \|x\|_1 = \sum_{i=1}^n |x_i|$ for $x \in \mathbb{R}^n$. Define

$$f'(x) = (s(x_1), s(x_2), \dots, s(x_n)) \quad \forall x \in \mathbb{R}^n$$

where

$$s(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \\ -1 & \text{if } t < 0 \end{cases}$$

Use part (a) to show $f'(x) \in \partial f(x)$ for all $x \in \mathbb{R}^n$.

Exercise 34: (Subgradient of $\|x\|_p$ for $p > 1$) Fix $p > 1$. Define the convex function

$$f(x) = \|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p} \quad \forall x \in \mathbb{R}^n$$

We want to compute a subgradient $f'(x) \in \partial f(x)$ for all $x \in \mathbb{R}^n$.

a) Explain why we can use $f'(0) = 0$.

b) Fix $x \neq 0$ and note that $\nabla f(x)$ exists. Compute $\nabla f(x)$.

Exercise 35: (Subgradient of composition) Let $g : \mathbb{R}^k \rightarrow \mathbb{R}$ be convex and entrywise nondecreasing with subgradients $f'(y)$ for $y \in \mathbb{R}^k$. For $i \in \{1, \dots, k\}$ let $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ be convex with subgradients $f'_i(x)$ for $x \in \mathbb{R}^n$. Define the function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ by $h(x) = g(f_1(x), f_2(x), \dots, f_k(x))$ for all $x \in \mathbb{R}^n$.

a) Show that h is a convex function.

b) Define $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$ by $F(x) = (f_1(x), \dots, f_k(x))$. Show that $F'(x)g'(F(x)) \in \partial h(x)$ for all $x \in \mathbb{R}^n$, where $F'(x) \sim n \times k$ and the i th column of $F'(x)$ is the column vector $f'_i(x)$ for $i \in \{1, \dots, k\}$. To start, fix $y \in \mathbb{R}^n$ and observe

$$h(y) = g(f_1(y), f_2(y), \dots, f_k(y)) \geq g(f_1(x) + f'_1(x)^\top(y - x), \dots, f_k(x) + f'_k(x)^\top(y - x)) = g(F(x) + F'(x)^\top(y - x))$$

Exercise 36: (Stochastic gradient) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex and differentiable function. Fix $x \in \mathbb{R}^n$ and consider

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right)$$

Computing $\nabla f(x)$ involves computing n partial derivatives. Let J be a random variable that is uniform over $\{1, \dots, n\}$. Consider the random vector

$$S = \left(0, 0, \dots, n \frac{\partial f(x)}{\partial x_J}, 0, \dots, 0 \right)$$

which is zero in all components except for component J . The random vector S is called a *stochastic gradient*. It is n times less expensive to compute because it only involves a single randomly chosen partial derivative.

a) Show that $\mathbb{E}[S] = \nabla f(x)$.

b) Show that $\mathbb{E}[||S||^2] = n||\nabla f(x)||^2$.

Exercise 37: (B -bounded subgradients $\implies B$ -Lipschitz) Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. Let $f : X \rightarrow \mathbb{R}$ be a convex function that has B -bounded subgradients for some $B \geq 0$. Show that f is B -Lipschitz continuous.

Exercise 38: (B -Lipschitz over $\mathbb{R}^n \implies B$ -bounded subgradients) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex and B -Lipschitz continuous function for some $B \geq 0$. Theorem 3 ensures f has at least one subgradient at each $x \in \mathbb{R}^n$. Fix $x \in X$ and $f'(x) \in \partial f(x)$. Show $\|f'(x)\| \leq B$. In particular, f has B -bounded subgradients. Hint: Use the subgradient inequality to compare $f(y)$ and $f(x)$, then consider $y = x + f'(x)$.

Exercise 39: (B -bounded subgradients $\iff B$ -Lipschitz [6][7]) Let $X \subseteq \mathbb{R}^n$ be a nonempty convex set. Let $f : X \rightarrow \mathbb{R}$ be a convex function that is B -Lipschitz continuous.

a) Show that for all $y \in \mathbb{R}^n$ the expression $\inf_{x \in X} \{f(x) + B\|y - x\|\}$ is finite. Hint: Fix $a \in X$ and observe that for any x we have $f(x) \geq f(a) - B\|a - x\|$.

b) It can be shown that our convex function f can be convexly extended to a function $\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ that is convex, B -Lipschitz, and satisfies $\tilde{f}(x) = f(x)$ for all $x \in X$ (McShane-Whitney extension theorem [6] and Theorem 2.1 in [7]). Use Exercise 38 to claim that $\tilde{f}$ has B -bounded subgradients. Use Lemma 2 to conclude that f has B -bounded subgradients, that is, for all $x \in X$ there is a subgradient $f'(x) \in \partial f(x)$ such that $\|f'(x)\| \leq B$.

Exercise 40: (A particular stochastic gradient) Fix $x \in \mathbb{R}^n$. The vector x is treated here as a nonrandom constant. Define the random vector $S(x)$ according to (53).

a) Show that $\mathbb{E}[S(x)] = \nabla f(x)$ for the function f in (52).

b) Compute $\mathbb{E}[\|S(x)\|^2]$ and $\text{Var}(\|S(x)\|)$.

Exercise 41: (Approximate subgradients) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set with diameter at most D , so $\|x - y\| \leq D$ for all $x, y \in C$. We want to minimize $f(x)$ over all $x \in C$. Suppose $x^* \in C$ is a solution. Fix $x_0 \in C$ and $\delta > 0$. We use a projected descent $x_{k+1} = \pi_C(x_k - \delta s_k)$ for $k \in \{0, 1, 2, \dots\}$. However, $s_k \in \mathbb{R}^n$ is only an *approximate* subgradient, so $s_k = f'(x_k) + z_k$ for all $k \in \{0, 1, 2, \dots\}$, where $f'(x_k) \in \partial f(x_k)$ and z_k is an error term. Suppose there are constants $B > 0$, $\epsilon > 0$, $S > 0$ such that

$$\|f'(x_k)\| \leq B, \quad \|z_k\| \leq \epsilon, \quad \|s_k\| \leq S \quad \forall k \in \{0, 1, 2, \dots\}$$

Provide an error bound analysis to obtain an approximate solution $\tilde{x}_m \in C$ after m iterations, where $f(\tilde{x}_m) \leq f(x^*) + \text{error}_m$ for some term error_m that depends on $D, B, \epsilon, S, m, \delta$, and such that $\lim_{m \rightarrow \infty} \text{error}_m = O(\delta) + O(\epsilon)$. Recall the Cauchy-Schwarz inequality $|a^\top b| \leq \|a\| \cdot \|b\|$ for all $a, b \in \mathbb{R}^n$.

Exercise 42: (Analysis of stochastic gradient method for nonsmooth convex functions) Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set. Let $f : C \rightarrow \mathbb{R}$ be a (possibly nonsmooth) convex function. Let $x^* \in C$ be a minimizer. We use a stochastic subgradient method with fixed stepsize $\delta > 0$: Fix $x_0 \in C$. Then

$$x_{k+1} = \pi_C(x_k - S_k \delta) \quad \forall k \in \{0, 1, 2, \dots\}$$

where S_k is a stochastic subgradient. Note that $\{x_k\}_{k=0}^\infty$ are now random vectors. Suppose that for each $k \in \{0, 1, 2, \dots\}$ the random vector S_k satisfies: (i) $\mathbb{E}[S_k | x_k] = f'(x_k)$; (ii) $\mathbb{E}[\|S_k\|^2] \leq G$, where $f'(x_k) \in \partial f(x_k)$, and where $G > 0$ is some value. This question follows the proof of Theorem 8.

a) Show that for each $k \in \{0, 1, 2, \dots\}$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 + \delta^2 \|S_k\|^2 + 2\delta S_k(x^* - x_k)$$

b) Take a conditional expectation given x_k to conclude

$$\mathbb{E}[\|x_{k+1} - x^*\|^2 | x_k] \leq \|x_k - x^*\|^2 + \delta^2 \mathbb{E}[\|S_k\|^2 | x_k] + 2\delta f'(x_k)(x^* - x_k)$$

Use the subgradient inequality to change the last term on the right-hand-side to something useful.

c) Use the law of iterated expectations ($\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]]$) to conclude that for all $k \in \{0, 1, 2, \dots\}$

$$\mathbb{E}[f(x_k)] \leq f(x^*) + \frac{\delta G}{2} + \frac{\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2]}{2\delta}$$

d) Use Jensen's inequality to prove

$$\mathbb{E}[f(\tilde{x}_m)] \leq f(x^*) + \frac{\delta G}{2} + \frac{\|x_0 - x^*\|^2}{2\delta m} \quad \forall m \in \{1, 2, 3, \dots\}$$

where $\tilde{x}_m = \frac{1}{m} \sum_{k=0}^{m-1} x_k$. Compare this with the deterministic method to observe the only difference on the right-hand-side is the B changes to G ; the only difference on the left-hand-side is that the result holds in expectation.

Exercise 43: (Difficulty of analyzing the stochastic gradient method for smooth functions) Exercise 42 shows the analysis of stochastic subgradients for general (possibly nonsmooth problems) is almost the same as that for deterministic subgradients. Unfortunately, that is not the case for smooth functions. This problem shows the difficulty. Fix $C \subseteq \mathbb{R}^n$ as a nonempty, closed, convex set. Suppose we have a β -smooth convex function f with domain that contains C . Suppose $x^* \in C$ is a minimizer of f over all $x \in C$. We use a stochastic gradient method with fixed stepsize $\delta > 0$: Fix $x_0 \in C$. Then

$$x_{k+1} = \pi_C(x_k - S_k \delta) \quad \forall k \in \{0, 1, 2, \dots\}$$

where S_k is a stochastic gradient. Note that $\{x_k\}_{k=0}^\infty$ are now random vectors. Suppose that for each $k \in \{0, 1, 2, \dots\}$ the random vector S_k satisfies $\mathbb{E}[S_k | x_k] = \nabla f(x_k)$ and $\mathbb{E}[|S_k|^2] \leq G$, where $G > 0$ is some value. This question follows the proof of Theorem 16.

a) Show that for all $k \in \{0, 1, 2, \dots\}$ we have

$$S_k^\top (x_{k+1} - x_k) + \frac{1}{2\delta} \|x_{k+1} - x_k\|^2 \leq S_k^\top (x^* - x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \|x^* - x_{k+1}\|^2$$

b) Take conditional expectations given x_k to show

$$\mathbb{E}[S_k^\top (x_{k+1} - x_k) | x_k] + \frac{1}{2\delta} \mathbb{E}[\|x_{k+1} - x_k\|^2 | x_k] \leq f(x^*) - f(x_k) + \frac{1}{2\delta} \|x^* - x_k\|^2 - \frac{1}{2\delta} \mathbb{E}[\|x^* - x_{k+1}\|^2 | x_k]$$

c) Good news: Explain why $\mathbb{E}[S_k^\top x_k | x_k] = \nabla f(x_k)^\top x_k$.

b) Bad news: Explain why we cannot conclude $\mathbb{E}[S_k^\top x_{k+1} | x_k] = \mathbb{E}[\nabla f(x_k)^\top x_{k+1} | x_k]$. This derails the proof and we cannot reach any conclusion.

Exercise 44: (Adaptive price) You want to buy a particular book. You will buy the book as long as the price is less than or equal to q , where q is some positive real number. The book is only available in a particular bookstore, you visit the store every day, and nobody else is interested in buying the book. The value q is a *secret threshold price* because the bookstore owner does not know q . The owner sets the initial price of the book at p . At the end of every day when the book does not sell, the owner reduces the price by a factor of 2 and uses that reduced price on the following day. Let $\text{sale}_q(p)$ be the final sale price of the book, given initial price p and secret threshold q . For example, if $q = 10$ and $p = 100$ then:

- Day 1: $p = 100$ (no sale)
- Day 2: $p = 50$ (no sale)
- Day 3: $p = 25$ (no sale)
- Day 4: $p = 12.5$ (no sale)
- Day 5: $p = 6.25$ ($6.25 \leq 10 \implies$ sell book)

so the book sells in 5 days with final sale price $\text{sale}_{10}(100) = 6.25$ dollars. Clearly $\text{sale}_q(p) = p$ whenever $p \leq q$. Assume $p > q$. Define $g(q) = \inf_{\{p: p > q\}} \{\text{sale}_q(p)\}$. The bookstore owner is guaranteed to receive at least $g(q)$ dollars upon eventual sale of the book.

- a) Find $g(q)$ as a function of q . Give its value for $q = 10$.
- b) In terms of p and q , how many days does it take to sell the book?
- c) What are the tradeoffs if the owner replaces the divide-by-2 rule with a divide-by-1.1 rule? Compare for $q = 10$.

Exercise 45: (Adaptive stepsize) We have a convex and β -smooth convex function. Assume $\beta > 1$, but the exact value of β is unknown. We use an adaptive stepsize δ_k for $k \in \{0, 1, 2, \dots\}$, with $\delta_0 = 1$. Let Condition_k be a particular condition that we can check based on information available on step k . Assume that for each k the following is satisfied:

$$(\text{Condition}_k \text{ is false}) \implies \delta_k > 1/\beta$$

On every step k we do:

- If Condition_k is true then $\delta_{k+1} = \delta_k$.
 - If Condition_k is false then $\delta_{k+1} = \delta_k/1.3$.
- a) Find a value $\delta_{\min} > 0$ such that $\delta_k \geq \delta_{\min}$ for all $k \in \{0, 1, 2, \dots\}$.
 - b) What is the largest number of time steps k for which Condition_k can be false?

Exercise 46: (Local minimizer) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a (possibly nonconvex) function. We say that a point $x^* \in \mathbb{R}^n$ is a *local minimizer* if there is an $\epsilon > 0$ such that x^* minimizes $f(x)$ over all $x \in \overline{B}_\epsilon(x^*)$.

a) Show that if f is a convex function and x^* is a local minimizer, then it is also a global minimizer, meaning $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$. Hint: Fix $w \in \mathbb{R}^n$. Compare $f(x^*)$ and $f(w)$ by considering $f(pw + (1-p)x^*)$.

b) Give an example differentiable function $f : \mathbb{R} \rightarrow \mathbb{R}$ and point $x^* \in \mathbb{R}$ that is a local but not global minimizer.

Exercise 47: (A monotone property for Lagrange multipliers) Let $X \subseteq \mathbb{R}^n$ and $f : X \rightarrow \mathbb{R}$, $g_i : X \rightarrow \mathbb{R}$ for $i \in \{1, \dots, k\}$. Define $g(x) = (g_1(x), \dots, g_k(x))$ for $x \in X$. Fix $\mu_1, \mu_2 \in \mathbb{R}^k$ such that $\mu_1, \mu_2 \geq 0$. Suppose x_1 minimizes $f(x) + \mu_1^\top g(x)$ over all $x \in X$, and x_2 minimizes $f(x) + \mu_2^\top g(x)$ over all $x \in X$.

a) Prove that $(\mu_1 - \mu_2)^\top (g(x_1) - g(x_2)) \leq 0$.

b) Suppose $k = 1$ and so $g(x) = g_1(x)$. Suppose $\mu_1 = 97.3$ and $g(x_1) = 50$. If $\mu_2 > 97.3$, show that $g(x_2) \leq 50$.

Exercise 48: (Lagrange multiplier theorem) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}$. Fix $\mu = 34.2$ and suppose x^* is a minimizer of $f(x) + 34.2g(x)$ over all $x \in \mathbb{R}^n$. Suppose $f(x^*) = 45.7$ and $g(x^*) = -1.1$. What constrained problem does x^* solve?

Exercise 49: (Lagrange multiplier theorem again) Fix $X \subseteq \mathbb{R}^n$, $f : X \rightarrow \mathbb{R}$, $g_i : X \rightarrow \mathbb{R}$ for $i \in \{1, 2\}$. Suppose x^* minimizes $f(x) + 8g_1(x) + 9.2g_2(x)$ subject to $x \in X$. Suppose $f(x^*) = -2.9$, $g_1(x^*) = 100$, $g_2(x^*) = -7.1$. What constrained problem does x^* solve?

Exercise 50: (Minimizing a convex function over an interval) Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a convex function. Suppose $x^* \in \mathbb{R}$ minimizes f over all $x \in \mathbb{R}$. Fix a, b as real numbers such that $a < b$. Prove that $[x^*]_a^b$ minimizes $f(x)$ over all $x \in [a, b]$, where $[x]_a^b$ projects the real number x onto the interval $[a, b]$:

$$[x]_a^b = \begin{cases} b & \text{if } x > b \\ x & \text{if } x \in [a, b] \\ a & \text{if } x < a \end{cases}$$

Hint: Treat the cases $x > b$ and $x \in [a, b]$ separately (the case $x < a$ is similar to $x > b$). Draw a picture to gain insight. [Note: This is a useful fact about convex functions of one variable. It does not generalize to multiple variables.]

Exercise 51: (Minimizing certain convex functions of one variable) Fix $a > 0$ and $\lambda > 0$. Problems of the following type often arise when solving separable unconstrained Lagrange multiplier problems. Solve these:

- Minimize $5x$ subject to $x \geq 0$.
- Minimize $-7x$ subject to $x \in [-3, 3]$.
- Minimize ax subject to $x \in [-3, 3]$.
- Minimize $(a - 8)x$ subject to $x \in [-3, 3]$.
- Minimize $40e^{-x} + x$ subject to $x \geq 0$.
- Minimize $e^{-x} + 40x$ subject to $x \geq 0$.
- Minimize $ae^{-x} + \lambda x$ subject to $x \geq 0$.
- Minimize $-\log(1 + ax) + \lambda x$ subject to $x \in [0, 4]$.

Exercise 52: (Water filling) Consider the problem

$$\begin{aligned} \text{Maximize: } & \sum_{i=1}^4 \log(1 + \beta_i x_i) \\ \text{Subject to: } & \sum_{i=1}^4 x_i \leq \theta \\ & x_i \geq 0 \quad \forall i \in \{1, \dots, 4\} \end{aligned}$$

where $\beta_1 = 9, \beta_2 = 7, \beta_3 = 4.5, \beta_4 = 3$. Note that $\log(\cdot)$ denotes the natural logarithm. The value θ is a positive real number. Let $x^* = (x_1^*, x_2^*, x_3^*, x_4^*)$ be an optimal solution.

- What is the largest value of θ for which $x_2^* = 0$?
- What is the largest value of θ for which $x_3^* = 0$?
- Find x^* when $\theta = 0.4$.
- Find x^* when $\theta = 1.8$.

Exercise 53: (Constrained water filling) Solve the problem of Exercise 52 using $(\beta_1, \beta_2, \beta_3, \beta_4) = (9, 7, 4.5, 3)$ and $\theta = 0.4$, but replacing the $x_i \geq 0$ constraint with $x_i \in [0, 0.15]$ for all $i \in \{1, 2, 3, 4\}$.

Exercise 54: (Lagrange multiplier for minimum distance) Fix $(w_1, \dots, w_n), (c_1, \dots, c_n) \in \mathbb{R}^n$ and $d \in \mathbb{R}$. Use a Lagrange multiplier $\lambda \in \mathbb{R}$ to find $x^* \in \mathbb{R}^n$ to minimize $\|x - w\|^2$ subject to $\sum_{i=1}^n c_i x_i = d$. Assume Euclidean norm and $(c_1, \dots, c_n) \neq 0$.

Exercise 55: (Lagrange multiplier for minimum distance with nonnegativity constraint) What changes in Exercise 54 if we add the constraint $x_i \geq 0$ for all $i \in \{1, \dots, n\}$? Define $X = \{x \in \mathbb{R}^n : x_i \geq 0 \quad \forall i \in \{1, \dots, n\}\}$.

- Show the problem is infeasible if $c_i \geq 0$ for all $i \in \{1, \dots, n\}$ and $d < 0$.
- Assume $d > 0$ and $c_i > 0$ for all $i \in \{1, \dots, n\}$. Explain how to find a $\lambda \in \mathbb{R}$ that solves the problem.

Exercise 56: (Lagrange multiplier with one equality constraint) Solve

$$\begin{aligned} \text{Minimize: } & \sum_{i=1}^n b_i e^{a_i x_i} \\ \text{Subject to } & \sum_{i=1}^n c_i x_i = 5 \\ & x \in \mathbb{R}^n \end{aligned}$$

under the assumption $a_i > 0, b_i > 0, c_i > 0$ for all i . What can go wrong if this assumption is removed?

Exercise 57: (Projection onto a probability simplex) Fix $w = (w_1, \dots, w_n) \in \mathbb{R}^n$. Define the probability simplex $P = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x_i \geq 0 \ \forall i\}$. Show how to use a Lagrange multiplier λ to get an algorithm for computing $x^* = \pi_P(w)$. Hint: Define $f(x) = \|x - w\|^2$, $g(x) = \sum_{i=1}^n x_i$, $X = \{x \in \mathbb{R}^n : x_i \geq 0 \ \forall i \in \{1, \dots, n\}\}$.

Exercise 58: (Dual function is concave) This proves the first half of Lemma 18. Prove the dual function is concave over all $(\mu, \lambda) \in D$. Start your proof: Fix (μ_1, λ_1) and $(\mu_2, \lambda_2) \in D$ and $p \in [0, 1]$. We want to show

$$d(p(\mu_1, \lambda_1) + (1-p)(\mu_2, \lambda_2)) \geq pd(\mu_1, \lambda_1) + (1-p)d(\mu_2, \lambda_2)$$

Then use the general fact $\inf_{x \in X} \{a(x) + b(x)\} \geq \inf_{x \in X} \{a(x)\} + \inf_{x \in X} \{b(x)\}$.

Exercise 59: (Subgradient of dual function) This proves the second half of Lemma 18. Fix $(\mu, \lambda) \in D$. Let $\hat{x} \in X$ minimize $f(x) + \mu^\top g(x) + \lambda^\top h(x)$ over all $x \in X$. Show that $(g(\hat{x}), h(\hat{x})) \in \mathbb{R}^{k+m}$ is a subgradient of the (concave) dual function d . Equivalently, $-(g(\hat{x}), h(\hat{x}))$ is a subgradient of the convex function $-d$. Start your proof: Fix $(\tilde{\mu}, \tilde{\lambda}) \in D$. We have $d(\tilde{\mu}, \tilde{\lambda}) \leq \dots$

Exercise 60: (Dual subgradient algorithm for a small convex program) Consider the problem:

$$\begin{aligned} \text{Minimize: } & e^{x_1} + e^{2x_2} \\ \text{Subject to: } & x_1 + x_2 \geq 4 \\ & x_1, x_2 \in [0, 4] \end{aligned}$$

a) Define $X = [0, 4]^2$, $f(x_1, x_2) = e^{x_1} + e^{2x_2}$, $g(x_1, x_2) = 4 - (x_1 + x_2)$. The dual subgradient algorithm uses a single Lagrange variable $\mu(t) \geq 0$ for $t \in \{0, 1, 2, \dots\}$. State the decisions $(x_1(t), x_2(t))$ for the dual subgradient algorithm at each time step $t \in \{0, 1, 2, \dots\}$, and the corresponding update $\mu_j(t+1)$.

b) Solve this problem exactly using the Lagrange multiplier theorem (Theorem 18).

c) Comment on similarities/differences between (a) and (b).

Exercise 61: (Dual subgradient algorithm for LP with compact box constraint) Consider the linear program that seeks a vector $x = (x_1, \dots, x_n)$ to solve

$$\begin{aligned} \text{Minimize: } & \sum_{i=1}^n c_i x_i \\ \text{Subject to: } & \sum_{i=1}^n a_{ji} x_i \leq b_j \quad \forall j \in \{1, \dots, k\} \\ & x_i \in [L_i, U_i] \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

where $c_i, a_{ji}, b_j, L_i, U_i$ are given real numbers (where $L_i < U_i$). Define $f(x) = \sum_{i=1}^n c_i x_i$ and $g_j(x) = (\sum_{i=1}^n a_{ji} x_i) - b_j$.

a) Using dual variables $\mu_j(t) \geq 0$ for all $j \in \{1, \dots, k\}$ and the compact box $X = \times_{i=1}^n [L_i, U_i]$, state the decisions for the dual subgradient algorithm at each time step $t \in \{0, 1, 2, \dots\}$. You should use separability to get a very simple and exact result for $x_i(t) \in [L_i, U_i]$ and $\mu_j(t+1)$ for each step t . [Note: For linear programs, this is considered to be a slow algorithm.]

b) Repeat for the case when the compact box constraints $x_i \in [L_i, U_i]$ for all i are removed and replaced with $x \in \mathbb{R}^n$. What goes wrong in this case?

Exercise 62: (Dual subgradient algorithm for a simple NUM problem with one link) Consider a network utility maximization (NUM) problem with n users that share a single link of capacity c . Let x_i be the transmission rate of user i . Let $w_i \log(x_i)$ be the utility of user i , where $w_i > 0$ and we use a natural log. The problem is

$$\begin{aligned} \text{Maximize: } & \sum_{i=1}^n w_i \log(x_i) \\ \text{Subject to: } & \sum_{i=1}^n x_i \leq c \\ & x_i \in [\epsilon, c] \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

where $\epsilon > 0$ is a constant that represents the min transmission rate that each user must be allocated (assume $n\epsilon \leq c$). The objective function is called a *weighted proportional fairness* objective. Define $f(x) = -\sum_{i=1}^n w_i \log(x_i)$, $g(x) = \sum_{i=1}^n x_i - c$. Define $X = \{x \in \mathbb{R}^n : x_i \in [\epsilon, c] \ \forall i\}$. Initialize $\mu(0) = 0$. For each $t \in \{0, 1, 2, \dots\}$, give the values of $x_i(t)$ and $\mu(t+1)$

iterates for the dual subgradient algorithm for this problem. The $\mu(t+1)$ update should resemble a queueing equation with constant service rate c and with arrivals coming from each user. [Note: Since this problem has only one inequality constraint, we could also directly use the Lagrange multiplier method to solve the problem. The next problem considers multi-link networks.]

Exercise 63: (Dual subgradient for multihop network) Consider a network with k links, which link $l \in \{1, \dots, k\}$ having capacity $c_l \geq 0$ (in units of Megabytes/sec). There are n users that are allocated flow rates $x = (x_1, \dots, x_n)$ (in units of Megabytes/sec). Each user $i \in \{1, \dots, n\}$ takes a fixed path through the network. Let $a_{l,i}$ be a binary value that is 1 if link l is contained on the path of user i , and 0 else. The network utility maximization problem is

$$\begin{aligned} \text{Maximize: } & \sum_{i=1}^n w_i \log(1 + \beta x_i) \\ \text{Subject to: } & \sum_{i=1}^n a_{l,i} x_i \leq c_l \quad \forall l \in \{1, \dots, k\} \\ & x_i \in [0, x_{\max}] \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

where $w_i, \beta, x_{\max}$ are positive parameters. This ensures the total flow on each link $l \in \{1, \dots, k\}$ is less than or equal to the link capacity c_l . The function $w_i \log(1 + \beta x_i)$ is a weighted function that measure the satisfaction of user i when that user is allocated a transmission rate of x_i . Define $X = [0, x_{\max}]^n$, $f(x) = -\sum_{i=1}^n w_i \log(1 + \beta x_i)$, $g_l(x) = \sum_{i=1}^n a_{l,i} x_i$ for $l \in \{1, \dots, k\}$. We use the dual subgradient algorithm with Lagrange multiplier parameters $\mu_j(t) \geq 0$ for $j \in \{1, \dots, k\}$.

a) For each step t , given $\mu(t) = (\mu_1(t), \dots, \mu_k(t))$, solve to find $x(t)$. The corresponding unconstrained minimization should be separable, so you can find $x_i(t) \in [0, x_{\max}]$ separately for each user $i \in \{1, \dots, n\}$. Does the decision for user i need to know the w_j values for other users $j \neq i$? Which $\mu_l(t)$ values does user i need to know?

b) State the corresponding $\mu_j(t+1)$ update equation for each $j \in \{1, \dots, k\}$ in terms of your solution $x(t)$.

Exercise 64: (Variations on the same problem) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $h : \mathbb{R}^n \rightarrow \mathbb{R}$ be functions. Consider the following three equivalent problems:

P1: Minimize: $f(x)$; Subject to: $h(x) = 7$, $x \in \mathbb{R}^n$.

P2: Minimize: $f(x)$; Subject to: $h(x) - 7 = 0$, $x \in \mathbb{R}^n$.

P3: Minimize: $f(x)$; Subject to: $h(x) - 7 \leq 0$, $7 - h(x) \leq 0$, $x \in \mathbb{R}^n$.

Their corresponding unconstrained problems are as follows:

Q1: (Fix $\lambda \in \mathbb{R}$) Minimize $f(x) + \lambda h(x)$ subject to $x \in \mathbb{R}^n$.

Q2: (Fix $\lambda \in \mathbb{R}$) Minimize $f(x) + \lambda(h(x) - 7)$ subject to $x \in \mathbb{R}^n$.

Q3: (Fix $\mu_1 \geq 0, \mu_2 \geq 0$) Minimize $f(x) + \mu_1(h(x) - 7) + \mu_2(7 - h(x))$ subject to $x \in \mathbb{R}^n$.

a) Explain why Q1 and Q2 are equivalent, in the sense that x^* solves Q1 if and only if it solves Q2.

b) Define $\lambda = \mu_1 - \mu_2$. Explain why λ can be positive, zero, or negative. Explain why Q3 reduces to minimizing $f(x) + \lambda h(x)$.

c) Explain why all three problems Q1, Q2, Q3 eventually fix $\lambda \in \mathbb{R}$ and minimize $f(x) + \lambda h(x)$. Let $x^*(\lambda)$ be the solution in terms of λ . To solve P1/P2/P3, we adjust $\lambda \in \mathbb{R}$ in an effort to obtain $h(x^*(\lambda)) = 7$.

Exercise 65: (Example primal-dual LP) Consider the primal linear program for decision vector $x \in \mathbb{R}^4$.

$$\begin{aligned} \text{Minimize: } & \begin{bmatrix} -1 & -1 & -20 & 1 \end{bmatrix} x \\ \text{Subject to: } & \begin{bmatrix} -2 & 1 & 0 & -1 \\ 1 & 0 & 3 & 2 \\ -1 & 0 & 1 & 0 \end{bmatrix} x \leq \begin{bmatrix} -3 \\ 12 \\ 5 \end{bmatrix} \\ & x \geq 0 \end{aligned}$$

a) State the dual linear program in terms of the variable $\mu \in \mathbb{R}^3$, where $\mu \geq 0$.

b) Use matlab to solve both the primal and dual linear programs. Find x^* and μ^* , and show the two programs have the same optimal objective value.

Exercise 66: (Another primal-dual LP) Consider the primal linear program for decision vector $x \in \mathbb{R}^3$.

$$\begin{aligned} \text{Minimize: } & \begin{bmatrix} 1 & 3 & -2 \end{bmatrix} x \\ \text{Subject to: } & \begin{bmatrix} -2 & 1 & 0 \\ 1 & 0 & 3 \\ -1 & 0 & 1 \\ 0 & 7 & 0 \\ -1 & -1 & -1 \end{bmatrix} x \leq \begin{bmatrix} 4 \\ 8 \\ 4 \\ 14 \\ -4 \end{bmatrix} \\ & x \geq 0 \end{aligned}$$

a) State the dual linear program in terms of the variable $\mu \in \mathbb{R}^5$, where $\mu \geq 0$.

b) Use matlab to solve both the primal and dual linear programs. Find x^* and μ^* , and show the two programs have the same optimal objective value.

Exercise 67: (Dual linear program with different structure) Show that the dual function for the linear program (92)-(95) is

$$d(\mu, \lambda) = \begin{cases} -b^\top \mu - d^\top \lambda & \text{if } A^\top \mu + H^\top \lambda = -c \\ -\infty & \text{else} \end{cases} \quad \forall (\mu, \lambda) \in D$$

where $D = \{(\mu, \lambda) : \mu \in \mathbb{R}^k, \lambda \in \mathbb{R}^m, \mu \geq 0\}$. Next, show that maximizing $d(\mu, \lambda)$ over all $(\mu, \lambda) \in D$ leads to another linear program, called the dual LP corresponding to the primal LP (92)-(95).

Exercise 68: (Another dual linear program) Consider the linear program

$$\begin{aligned} \text{Minimize:} \quad & c_1^\top x + c_2^\top y \\ \text{Subject to:} \quad & A_1 x + A_2 y = b \\ & y \geq 0 \\ & x \in \mathbb{R}^n, y \in \mathbb{R}^m \end{aligned}$$

Assume A_1 is a $k \times n$ matrix and A_2 is a $k \times m$ matrix. Define $X = \{(x, y) : x \in \mathbb{R}^n, y \in \mathbb{R}^m, y \geq 0\}$.

- Find the dual function $d(\lambda) = \inf_{(x,y) \in X} \{c_1^\top x + c_2^\top y + \lambda^\top (A_1 x + A_2 y - b)\}$ for all $\lambda \in \mathbb{R}^k$.
- Show that maximizing $d(\lambda)$ over all $\lambda \in \mathbb{R}^k$ leads to another linear program.

Exercise 69: (Utilizing optimal Lagrange multipliers) Suppose: (i) the (possibly nonconvex) problem (79)-(82) has an optimal solution x^* ; (ii) strong duality holds; (iii) the corresponding dual problem of maximizing $d(\mu, \lambda)$ over all $(\mu, \lambda) \in D$ has a maximizer $(\mu^*, \lambda^*) \in D$.

- Show $f(x) + \mu^{*\top} g(x) + \lambda^{*\top} h(x) \geq f(x^*)$ for all $x \in X$.
- Define $\phi : X \rightarrow \mathbb{R}$ by

$$\phi(x) = f(x) + \mu^{*\top} g(x) + \lambda^{*\top} h(x) \quad \forall x \in X$$

Prove that x^* is a minimizer of ϕ over all $x \in X$. Further, prove that $\mu^{*\top} g(x^*) = 0$, $\lambda^{*\top} h(x^*) = 0$, $\phi(x^*) = f(x^*)$.

- Show that if $z \in X$ is a minimizer of ϕ over all $x \in X$, and if z satisfies the constraints of problem (79)-(82), and if $\mu^{*\top} g(z) = 0$, then z is an optimal solution to problem (79)-(82).
- Argue that if X is a convex set and ϕ is a *strictly* convex function, then ϕ has a unique minimizer over all $x \in X$, and this unique minimizer is also the unique optimal solution to problem (79)-(82).

REFERENCES

- [1] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Boston, 2004.
- [2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [3] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *Proc. 20th International Conference on Machine Learning (ICML)*, 2003.
- [4] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.
- [5] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003.
- [6] Iosif Petrakis. McShane-Whitney extensions in constructive analysis. *Logical Methods in Computer Science*, Volume 16, Issue 1, February 2020.
- [7] Kamiar Asgari and Michael J. Neely. Projection-free non-smooth convex programming, 2023.
- [8] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM J. Optimization*, 2008.
- [9] X. Wei, H. Yu, and M. J. Neely. Online primal-dual mirror descent under stochastic constraints. *Proc. ACM Meas. Anal. Comput. Syst.*, 4(2), 2020.
- [10] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.
- [11] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, vol. 69, no. 2, pp. 169-192, Dec. 2007.
- [12] E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, vol. 15 (July):pp. 2489–2512, 2014.
- [13] T. M. Cover. Universal portfolios. *Mathematical Finance*, vol. 1, no. 1, pp. 1-29, Jan. 1991.
- [14] D. P. Bertsekas. *Nonlinear Programming*, 2nd ed. Athena Scientific, Belmont, MA, 1999.
- [15] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
- [16] I. Necoara and V. Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1232-1243, May 2014.
- [17] H. Yu and M. J. Neely. On the convergence time of dual subgradient methods for strongly convex programs. *IEEE Transactions on Automatic Control*, 63(4):1105–1112, April 2018.
- [18] H. Yu and M. J. Neely. A simple parallel algorithm with an $O(1/t)$ convergence rate for general convex programs. *SIAM Journal on Optimization*, 27(2):759–783, 2017.