6.2.1 Search Direction

To choose $\vec{v}_t$, we want to ensure that $f(\vec{x}_t + \eta\vec{v}_t) \leq f(\vec{x}_t)$ for some small η. This motivates taking $\vec{v}_t$ as the *direction of steepest descent*, i.e., the direction in which f decays the fastest. The degree (i.e., the steepness) of the rate of change can be characterized by the directional derivative $Df(\vec{x})[\vec{v}] = \vec{v}^\top[\nabla f(\vec{x})]$.

Theorem 148 (Negative Gradient is Direction of Steepest Descent)
Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, and let $\vec{x} \in \mathbb{R}^n$. Then

$$-\frac{\nabla f(\vec{x})}{\|\nabla f(\vec{x})\|_2} \in \operatorname*{argmin}_{\substack{\vec{v}\in\mathbb{R}^n \\ \|\vec{v}\|_2=1}} Df(\vec{x})[\vec{v}]. \tag{6.8}$$

Proof. Left as exercise. □

We also want to use the norm of the gradient in our update. For example, if $f(x) = x^2$ then $f'(x) = 2x$, which has large norm when x is far from the optimal point $x^\star = 0$. In this way, if the gradient is large then we are usually far away from an optimum, and we want our update $\eta\vec{v}_t$ to be large. This motivates choosing $\vec{v}_t = -\nabla f(\vec{x}_t)$.

6.2.2 Defining Gradient Descent

This choice of $\vec{v}_t$ gives the famous gradient descent iteration scheme:

$$\vec{x}_{t+1} = \vec{x}_t - \eta\nabla f(\vec{x}_t), \qquad \forall t = 0, 1, 2, \ldots. \tag{6.9}$$

We can formalize this in an algorithm which terminates after T iterations for a user-set T.

Algorithm 3 Gradient Descent.

```
1: function GradientDescent($f, \vec{x}_0, \eta, T$)
2:     for $t = 0, 1, \ldots, T-1$ do
3:         $\vec{x}_{t+1} \leftarrow \vec{x}_t - \eta\nabla f(\vec{x}_t)$
4:     end for
5:     return $\vec{x}_T$
6: end function
```

It is important to note that with the choice $\vec{v}_t = -\nabla f(\vec{x}_t)$, *descent is not guaranteed.* Namely, for a fixed $\eta > 0$, it is *not* true in general that

$$f(\vec{x}_{t+1}) = f(\vec{x}_t - \eta\nabla f(\vec{x}_t)) \leq f(\vec{x}_t). \tag{6.10}$$

Rather, from the proof of the above theorem, we only have that, for any given t there *exists* $\eta_t > 0$ such that

$$f(\vec{x}_t - \eta_t\nabla f(\vec{x}_t)) \leq f(\vec{x}_t). \tag{6.11}$$

This η_t, in general, is very small, and depends heavily on t and the local geometry of f around $\vec{x}_t$. None of these details are known by any realistic implementation of the gradient descent algorithm. In what follows, we will study gradient descent with a constant step size; this setting is most common in practice.

6.2.3 Convergence Analysis of Gradient Descent

When applying GD algorithm to a problem we need to make sure that we're making progress and that we eventually converge to the optimal solution. This question is very related to the choice of the step size. For the reminder of the section we will show that, for certain classes of functions $f\colon \mathbb{R}^n \to \mathbb{R}$ and certain chosen step sizes $\eta > 0$, the gradient descent algorithm is guaranteed to make progress in every iteration and converge to the optimal solution. We will first explore these questions through a familiar example.

Example 149 (Gradient Descent for Least Squares)**.** In this example we explore the convergence properties of the gradient descent algorithm by applying it to the least squares problem. Let $A \in \mathbb{R}^{m\times n}$ have full column rank, and $\vec{y} \in \mathbb{R}^m$.

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 . \tag{6.12}$$

Recall that, in this setting, the least squares problem has the unique closed form solution

$$\vec{x}^\star = (A^\top A)^{-1} A^\top \vec{y}. \tag{6.13}$$

We will use our knowledge of the true solution to analyze the convergence of the gradient descent algorithm. To apply gradient descent to this problem, let us first compute the gradient, which we see as

$$\nabla f(\vec{x}) = 2A^\top (A\vec{x} - \vec{b}). \tag{6.14}$$

Now let $\vec{x}_0 \in \mathbb{R}^n$ be the initial guess. For t a non-negative integer, we can write the gradient descent step:

$$\begin{aligned}
\vec{x}_{t+1} &= \vec{x}_t - \eta \nabla f(\vec{x}_t) && (6.15)\\
&= \vec{x}_t - 2\eta A^\top (A\vec{x}_t - \vec{y}) && (6.16)\\
&= (I - 2\eta A^\top A)\vec{x}_t + 2\eta A^\top \vec{y}. && (6.17)
\end{aligned}$$

We aim to set η which achieves the following two desired properties for the gradient descent iterates $\vec{x}_t$:

- We make progress, i.e., in every iteration we get closer to the optimal solution:

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2 \leq \|\vec{x}_t - \vec{x}^\star\|_2 . \tag{6.18}$$

- We eventually converge to the optimal solution:

$$\lim_{t\to\infty} \vec{x}_t = \vec{x}^\star \iff \lim_{t\to\infty} \|\vec{x}_t - \vec{x}^\star\|_2 = 0. \tag{6.19}$$

To study this, let us write write out the relationship between $\vec{x}_{t+1} - \vec{x}^\star$ and $\vec{x}_t - \vec{x}^\star$. We do this by subtracting $\vec{x}^\star$ from both sides of Equation (6.17) and do some algebraic manipulations.

$$\begin{aligned}
\vec{x}_{t+1} - \vec{x}^\star &= (I - 2\eta A^\top A)\vec{x}_t + 2\eta A^\top \vec{y} - \vec{x}^\star && (6.20)\\
&= (I - 2\eta A^\top A)\vec{x}_t + 2\eta A^\top \vec{y} - (A^\top A)^{-1} A^\top \vec{y} && (6.21)\\
&= (I - 2\eta A^\top A)\vec{x}_t + 2\eta (A^\top A)(A^\top A)^{-1} A^\top \vec{y} - (A^\top A)^{-1} A^\top \vec{y} && (6.22)\\
&= (I - 2\eta A^\top A)\vec{x}_t + 2\eta A^\top A\vec{x}^\star - \vec{x}^\star && (6.23)\\
&= (I - 2\eta A^\top A)\vec{x}_t - (I - 2\eta A^\top A)\vec{x}^\star && (6.24)\\
&= (I - 2\eta A^\top A)(\vec{x}_t - \vec{x}^\star). && (6.25)
\end{aligned}$$

Here we used the trick of introducing $I = (A^\top A)(A^\top A)^{-1}$; this is because we wanted to group terms with $A^\top A$, and we also wanted to introduce instances of $\vec{x}^\star = (A^\top A)^{-1}A^\top \vec{y}$. Thus, while this was a trick, it was a motivated trick. In the end we get the following relationship:

$$\vec{x}_{t+1} - \vec{x}^\star = (I - 2\eta A^\top A)(\vec{x}_t - \vec{x}^\star). \tag{6.26}$$

If we take the norm of both sides, we have

$$\begin{aligned} \|\vec{x}_{t+1} - \vec{x}^\star\|_2 &= \left\|(I - 2\eta A^\top A)(\vec{x}_t - \vec{x}^\star)\right\|_2 & (6.27)\\ &\le \left\|I - 2\eta A^\top A\right\|_2 \|\vec{x}_t - \vec{x}^\star\|_2 & (6.28)\\ &= \sigma_{\max}\{I - 2\eta A^\top A\} \|\vec{x}_t - \vec{x}^\star\|_2 . & (6.29)\end{aligned}$$

Thus, if $\sigma_{\max}\{I - 2\eta A^\top A\} < 1$, then we have

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2 \le \sigma_{\max}\{I - 2\eta A^\top A\} \|\vec{x}_t - \vec{x}^\star\|_2 < \|\vec{x}_t - \vec{x}^\star\|_2 . \tag{6.30}$$

Thus we are guaranteed to make progress at each step. For the convergence guarantee, we recursively apply Equation (6.29) to get

$$\begin{aligned} \|\vec{x}_{t+1} - \vec{x}^\star\|_2 &\le \sigma_{\max}\{I - 2\eta A^\top A\} \|\vec{x}_t - \vec{x}^\star\|_2 & (6.31)\\ &\le \sigma_{\max}\{I - 2\eta A^\top A\}^2 \|\vec{x}_{t-1} - \vec{x}^\star\|_2 & (6.32)\\ &\le \cdots & (6.33)\\ &\le \sigma_{\max}\{I - 2\eta A^\top A\}^{t+1} \|\vec{x}_0 - \vec{x}^\star\|_2 . & (6.34)\end{aligned}$$

Thus taking the limit we get

$$\begin{aligned} \lim_{t\to\infty} \|\vec{x}_t - \vec{x}^\star\|_2 &\le \lim_{t\to\infty} \sigma_{\max}\{I - 2\eta A^\top A\}^t \|\vec{x}_0 - \vec{x}^\star\|_2 & (6.35)\\ &= \|\vec{x}_0 - \vec{x}^\star\|_2 \cdot \underbrace{\lim_{t\to\infty} \sigma_{\max}\{I - 2\eta A^\top A\}^t}_{=0} & (6.36)\\ &= 0. & (6.37)\end{aligned}$$

Thus our convergence guarantee holds as well, so long as $\sigma_1\{I - 2\eta A^\top A\} < 1$.

Let us now generalize this analysis to a larger class of functions that includes least squares, as well as more general functions. We will focus our attention on the class of L-smooth and μ-strongly convex functions. Similar to what we did for least squares, we will use the optimal solution $\vec{x}^\star$ in our analysis. For a general function f, the quantity $\vec{x}^\star$ is almost always not known. But in the case of L-smooth and μ-strongly convex functions, a unique global minimizer exists. We just need the fact that it exists to show that for some small enough choice of the step size η, the gradient descent algorithm converges to this optimal solution. Before we formally state and prove this, we want to introduce one property of L-smooth functions that will become useful in our following proof.

Lemma 150 (Gradient Bound for L-Smooth Functions)
Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be an L-smooth function. For all $\vec{x} \in \mathbb{R}^n$, we have

$$\|\nabla f(\vec{x})\|_2^2 \le 2L\left(f(\vec{x}) - \min_{\vec{x}' \in \mathbb{R}^n} f(\vec{x}')\right). \tag{6.38}$$

This lemma says that the magnitude of the gradient gets smaller as we get closer to the optimal solution.

Proof. Recall the definition of L-smooth functions:

$$f(\vec{y}) \leq f(\vec{x}) + [\nabla f(\vec{x})]^\top (\vec{y} - \vec{x}) + \frac{L}{2} \|\vec{y} - \vec{x}\|_2^2 . \tag{6.39}$$

This is true for all points $\vec{x}, \vec{y} \in \mathbb{R}^n$, so let us fix $\vec{x}$ and set $\vec{y} = \vec{x} - \frac{\nabla f(\vec{x})}{L}$. We get

$$\begin{aligned} f\left(\vec{x} - \frac{\nabla f(\vec{x})}{L}\right) &\leq f(\vec{x}) + [\nabla f(\vec{x})]^\top \left(-\frac{\nabla f(\vec{x})}{L}\right) + \frac{L}{2}\left\|-\frac{\nabla f(\vec{x})}{L}\right\|_2^2 && (6.40)\\ &= f(\vec{x}) - \frac{1}{L}\|\nabla f(\vec{x})\|_2^2 + \frac{1}{2L}\|\nabla f(\vec{x})\|_2^2 && (6.41)\\ &= f(\vec{x}) - \frac{1}{2L}\|\nabla f(\vec{x})\|_2^2 . && (6.42)\end{aligned}$$

Now we can use the fact that $\min_{\vec{x}'} f(\vec{x}') \leq f(\vec{z})$ for all $\vec{z} \in \mathbb{R}^n$ to get

$$\begin{aligned} \min_{\vec{x}' \in \mathbb{R}^n} f(\vec{x}') &\leq f\left(\vec{x} - \frac{\nabla f(\vec{x})}{L}\right) && (6.43)\\ &\leq f(\vec{x}) - \frac{1}{2L}\|\nabla f(\vec{x})\|_2^2 . && (6.44)\end{aligned}$$

Rearranging, we have

$$\|\nabla f(\vec{x})\|_2^2 \leq 2L\left(f(\vec{x}) - \min_{\vec{x}' \in \mathbb{R}^n} f(\vec{x}')\right) \tag{6.45}$$

as desired. □

Now we have all the needed tools to prove the following property of gradient descent: for any μ-strongly convex and L-smooth function, the gradient descent algorithm converges exponentially fast to the true solution $\vec{x}^\star$.

Theorem 151 (Convergence of Gradient Descent for Smooth Strongly Convex Functions)
Let $\mu, L > 0$. Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be an L-smooth, μ-strongly convex function. Consider the following optimization problem:

$$p^\star = \min_{\vec{x} \in \mathbb{R}^n} f(\vec{x}) \tag{6.46}$$

which has optimal solution $\vec{x}^\star$. Then the constant step size $\eta = \frac{1}{L}$ is such that, if applying the gradient descent algorithm generates the sequence of points

$$\vec{x}_{t+1} = \vec{x}_t - \eta \nabla f(\vec{x}_t), \tag{6.47}$$

then for all initializations $\vec{x}_0$ and non-negative integers t,

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2^2 \leq \left(1 - \frac{\mu}{L}\right)\|\vec{x}_t - \vec{x}^\star\|_2^2 . \tag{6.48}$$

Proof. We want to upper bound $\|\vec{x}_{t+1} - \vec{x}^\star\|_2^2$, so let us write it out as

$$\begin{aligned} \|\vec{x}_{t+1} - \vec{x}^\star\|_2^2 &= \|\vec{x}_t - \eta\nabla f(\vec{x}_t) - \vec{x}^\star\|_2^2 && (6.49)\\ &= \|[\vec{x}_t - \vec{x}^\star] - \eta\nabla f(\vec{x}_t)\|_2^2 && (6.50)\\ &= \|\vec{x}_t - \vec{x}^\star\|_2^2 - 2\eta[\nabla f(\vec{x}_t)]^\top(\vec{x}_t - \vec{x}^\star) + \eta^2\|\nabla f(\vec{x}_t)\|_2^2 && (6.51)\end{aligned}$$

$$= \|\vec{x}_t - \vec{x}^\star\|_2^2 + 2\eta[\nabla f(\vec{x}_t)]^\top(\vec{x}^\star - \vec{x}_t) + \eta^2 \|\nabla f(\vec{x}_t)\|_2^2 . \tag{6.52}$$

Because f is L-smooth, the lemma gives

$$\|\nabla f(\vec{x}_t)\|_2^2 \leq 2L(f(\vec{x}_t) - f(\vec{x}^\star)). \tag{6.53}$$

Because f is μ-strongly convex, we can write

$$f(\vec{x}^\star) \geq f(\vec{x}_t) + [\nabla f(\vec{x}_t)]^\top(\vec{x}^\star - \vec{x}_t) + \frac{\mu}{2}\|\vec{x}^\star - \vec{x}_t\|_2^2 \tag{6.54}$$

$$\implies [\nabla f(\vec{x}_t)]^\top(\vec{x}_t - \vec{x}^\star) \geq f(\vec{x}_t) - f(\vec{x}^\star) + \frac{\mu}{2}\|\vec{x}_t - \vec{x}^\star\|_2^2 \tag{6.55}$$

$$\implies [\nabla f(\vec{x}_t)]^\top(\vec{x}^\star - \vec{x}_t) \leq f(\vec{x}^\star) - f(\vec{x}_t) - \frac{\mu}{2}\|\vec{x}_t - \vec{x}^\star\|_2^2 . \tag{6.56}$$

Plugging these two estimates in, we get

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2^2 = \|\vec{x}_t - \vec{x}^\star\|_2^2 + 2\eta[\nabla f(\vec{x}_t)]^\top(\vec{x}^\star - \vec{x}_t) + \eta^2\|\nabla f(\vec{x}_t)\|_2^2 \tag{6.57}$$

$$\leq \|\vec{x}_t - \vec{x}^\star\|_2^2 + 2\eta\left[f(\vec{x}^\star) - f(\vec{x}_t) - \frac{\mu}{2}\|\vec{x}_t - \vec{x}^\star\|_2^2\right] + \eta^2[2L(f(\vec{x}_t) - f(\vec{x}^\star))] \tag{6.58}$$

$$= (1 - \eta\mu)\|\vec{x}_t - \vec{x}^\star\|_2^2 + 2\eta(\eta L - 1)(f(\vec{x}_t) - f(\vec{x}^\star)). \tag{6.59}$$

So as to make the second term 0, we choose $\eta = \frac{1}{L}$. This gives

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2^2 \leq \left(1 - \frac{\mu}{L}\right)\|\vec{x}_t - \vec{x}^\star\|_2^2 . \tag{6.60}$$

□

Corollary 152. *Consider the setting of Theorem 151. We have that $0 \leq 1 - \frac{\mu}{L} < 1$, and consequently:*

(a) (Descent at every step.) $\|\vec{x}_{t+1} - \vec{x}^\star\|_2 \leq \|\vec{x}_t - \vec{x}^\star\|_2$ for all non-negative integers t and initializations $\vec{x}_0$.

(b) (Convergence.) $\lim_{t\to\infty} \vec{x}_t = \vec{x}^\star$.

Proof. We need to show that $1 - \frac{\mu}{L} \in [0, 1)$, or in other words that $0 < \frac{\mu}{L} \leq 1$. By assumption $\mu > 0$ so $\frac{\mu}{L} > 0$. The upper bound is given by the fact that f is μ-strongly convex and L-smooth, so

$$f(\vec{x}) + [\nabla f(\vec{x})]^\top(\vec{y} - \vec{x}) + \frac{\mu}{2}\|\vec{y} - \vec{x}\|_2^2 \leq f(\vec{y}) \leq f(\vec{x}) + [\nabla f(\vec{x})]^\top(\vec{y} - \vec{x}) + \frac{L}{2}\|\vec{y} - \vec{x}\|_2^2 \tag{6.61}$$

which implies that

$$f(\vec{x}) + [\nabla f(\vec{x})]^\top(\vec{y} - \vec{x}) + \frac{\mu}{2}\|\vec{y} - \vec{x}\|_2^2 \leq f(\vec{x}) + [\nabla f(\vec{x})]^\top(\vec{y} - \vec{x}) + \frac{L}{2}\|\vec{y} - \vec{x}\|_2^2 \tag{6.62}$$

which implies that

$$\frac{\mu}{2}\|\vec{y} - \vec{x}\|_2^2 \leq \frac{L}{2}\|\vec{y} - \vec{x}\|_2^2 \tag{6.63}$$

which finally implies that $\mu \leq L$, i.e., $0 < \frac{\mu}{L} \leq 1$. This immediately yields the first claim since $1 - \frac{\mu}{L} \in [0, 1)$ so $\sqrt{1 - \frac{\mu}{L}} \in [0, 1)$, therefore

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2 \leq \sqrt{1 - \frac{\mu}{L}}\|\vec{x}_t - \vec{x}^\star\|_2 < \|\vec{x}_t - \vec{x}^\star\|_2 . \tag{6.64}$$

The second claim follows since

$$\|\vec{x}_t - \vec{x}^\star\|_2^2 \leq \left(1 - \frac{\mu}{L}\right)\|\vec{x}_{t-1} - \vec{x}^\star\|_2^2 \tag{6.65}$$

$$\leq \left(1 - \frac{\mu}{L}\right)^2 \|\vec{x}_{t-2} - \vec{x}^\star\|_2^2 \tag{6.66}$$

$$\leq \cdots \tag{6.67}$$

$$\leq \left(1 - \frac{\mu}{L}\right)^t \|\vec{x}_0 - \vec{x}^\star\|_2^2 \tag{6.68}$$

and so

$$\lim_{t\to\infty} \|\vec{x}_t - \vec{x}^\star\|_2^2 = \lim_{t\to\infty} \left(1 - \frac{\mu}{L}\right)^t \|\vec{x}_0 - \vec{x}^\star\|_2^2 \tag{6.69}$$

$$= \|\vec{x}_0 - \vec{x}^\star\|_2^2 \cdot \underbrace{\lim_{t\to\infty} \left(1 - \frac{\mu}{L}\right)^t}_{=0} \tag{6.70}$$

$$= 0. \tag{6.71}$$

□

In this case the quantity $c = \sqrt{1 - \frac{\mu}{L}}$ is important. If we wanted to run gradient descent for T iterations to within accuracy ϵ, we would have

$$\|\vec{x}_T - \vec{x}^\star\|_2 \leq c^T \|\vec{x}_0 - \vec{x}^\star\|_2 . \tag{6.72}$$

If we set $D \doteq \|\vec{x}_0 - \vec{x}^\star\|$, then we can ensure that $\|\vec{x}_T - \vec{x}^\star\|_2 \leq \epsilon$ by bounding the right-hand side $c^T \|\vec{x}_0 - \vec{x}^\star\|_2 = c^T D$ by ϵ, getting

$$c^T D \leq \epsilon \tag{6.73}$$

$$\implies c^T \leq \frac{\epsilon}{D} \tag{6.74}$$

$$\implies T\log(c) \leq \log(\epsilon) - \log(D) \tag{6.75}$$

$$\implies T \geq \frac{\log(\epsilon) - \log(D)}{\log(c)} \tag{6.76}$$

$$= \frac{\log(1/\epsilon) + \log(D)}{\log(1/c)}. \tag{6.77}$$

Thus, the lower the value of c is, the faster we get convergence towards a given accuracy.

Finally, it is important to point out that this is not the only class of functions for which the gradient descent algorithm converges. For example for the class of functions that are L-smooth and convex (i.e., not μ-strictly convex), the gradient descent algorithm does converge; in particular it converges to within accuracy ϵ after $T \geq O(1/\epsilon)$ iterations. This is vastly slower than the convergence achieved for μ-strongly convex functions.

6.3 Variations: Stochastic Gradient Descent

In this section we will cover the stochastic gradient descent (SGD) algorithm. This variant is motivated by optimization problems in which computing the gradient can be computationally expensive. The idea behind SGD is to use random sampling to find an estimate of the gradient that is easy to compute. Let us consider the class of unconstrained differentiable optimization problems that take the form

$$p^\star = \min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) = \min_{\vec{x}\in\mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} f_i(\vec{x}). \tag{6.78}$$

The gradient of the objective function is given by

$$\nabla f(\vec{x}) = \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(\vec{x}), \tag{6.79}$$

and the gradient descent step is given by

$$\vec{x}_{t+1} = \vec{x}_t - \eta \cdot \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(\vec{x}), \tag{6.80}$$

This class of functions is very common in applications that involve learning from data, one example being the least squares problem. Recall that we can write the least squares problem as

$$\min_{\vec{x}\in\mathbb{R}^n} \underbrace{\frac{1}{m}\|A\vec{x}-\vec{y}\|_2^2}_{=f(\vec{x})} = \min_{\vec{x}\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^{m}\underbrace{(\vec{a}_i^\top\vec{x}-b_i)^2}_{=f_i(\vec{x})} \tag{6.81}$$

Note that we multiplied the objective function with the constant $\frac{1}{m}$, which does not change the solutions to the optimization problem. We will see shortly that this constant is important in the SGD setting.

If the number of functions m is large and if each gradient $\nabla f_i(\vec{x})$ is complex to evaluate, computing the full gradient in Equation (6.79) can become prohibitively expensive. To overcome this, SGD proposes to take a random sample from the set of functions $\{f_1, f_2, \ldots, f_m\}$, evaluate the gradient at that sample only, and take the step

$$\vec{x}_{t+1} = \vec{x}_t - \eta\nabla f_i(\vec{x}). \tag{6.82}$$

If this index i is drawn uniformly at random, we can see that the expected value of the estimated gradient is equal to the full gradient.

$$\mathbb{E}[\nabla f_i(\vec{x})] = \sum_{i=1}^{m}\frac{1}{m}\nabla f_i(\vec{x}) \tag{6.83}$$

$$= \nabla f(\vec{x}). \tag{6.84}$$

Note that the direction $-\nabla f_i(\vec{x})$ is not guaranteed to be the direction of steepest descent of the function $f(\vec{x})$. In fact, it might not be a descent direction at all, and the value of the function f might even *increase* by taking a step in this direction. Further, it is not guaranteed to satisfy the first order optimality conditions; that is, when $\nabla f(\vec{x}) = \vec{0}$, the gradient of a single function $\nabla f_i(\vec{x})$ is generally not zero. This already tells us that the notion of convergence we studied (which is called "last-iterate convergence") is practically impossible to achieve — and in particular, the type of convergence proof that we studied for gradient descent will not work here. Instead the type of convergence guarantees we can hope for in this setting is for the *averaged* point across all iterations will converge to the optimal solution, i.e.,

$$\lim_{T\to\infty} f\left(\frac{1}{T}\sum_{t=1}^{T}\vec{x}_t\right) = \min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}). \tag{6.85}$$

Further, to prove convergence of SGD we need to use a variable step size η_t. Using a fixed step size (like we did for gradient descent) doesn't guarantee convergence. The intuition behind this is that the gradient directions $\nabla f_i(\vec{x})$ for different i might be competing with each other and want to pull the solution in different directions. This will cause the sequence $\vec{x}_t$ to bounce back and forth. Using a variable step size η_t such that $\lim_{t\to\infty}\eta_t = 0$ makes it possible to converge to a single point. It is not trivial to show that the point that the algorithm will converge to is the optimal solution, but this can be proven under some assumptions on the objective function. For formal proofs, a good reference is [6].

Example 153 (Finding the Centroid Using SGD)**.** Fix points $\vec{p}_1, \ldots, \vec{p}_m \in \mathbb{R}^n$. Let us consider the problem of finding their *centroid*, which we formulate as

$$\min_{\vec{x} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{2} \|\vec{x} - \vec{p}_i\|_2^2}_{=f_i(\vec{x})}. \tag{6.86}$$

The optimal solution for this problem is just the average of the points

$$\vec{x}^\star = \frac{1}{m} \sum_{i=1}^{m} \vec{p}_i. \tag{6.87}$$

Let us apply SGD on this problem with the time varying step size $\eta_t = \frac{1}{t}$ and initial guess $\vec{x}_0 = \vec{0}$. We first compute the gradients

$$\nabla f_i(\vec{x}) = \vec{x} - \vec{p}_i. \tag{6.88}$$

To simplify the notation we will apply SGD by selecting the indices in order.

$$\text{Iteration 1}: \quad \vec{x}_1 = \vec{x}_0 - \frac{1}{1}(\vec{x}_0 - \vec{p}_1) = \vec{p}_1 \tag{6.89}$$

$$\text{Iteration 2}: \quad \vec{x}_2 = \vec{x}_1 - \frac{1}{2}(\vec{x}_1 - \vec{p}_2) = \frac{\vec{p}_1 + \vec{p}_2}{2} \tag{6.90}$$

$$\text{Iteration 3}: \quad \vec{x}_3 = \vec{x}_3 - \frac{1}{3}(\vec{x}_2 - \vec{p}_3) = \frac{\vec{p}_1 + \vec{p}_2 + \vec{p}_3}{3} \tag{6.91}$$

$$\vdots \tag{6.92}$$

$$\text{Iteration } m: \quad \vec{x}_m = \vec{x}_{m-1} - \frac{1}{m}(\vec{x}_{m-1} - \vec{p}_m) = \frac{\sum_{i=1}^{m} \vec{p}_i}{m}. \tag{6.93}$$

So the SGD algorithm which takes a step along the gradient of a single $\nabla f_i(\vec{x})$ in every iteration converges to the true solution in m iterations. It is important to note that this is a toy example that is used to illustrate the application of SGD. As we discussed earlier, this type of convergence behavior is not common or guaranteed when applying SGD to more complicated real world problems.

Finally, we note that this set up of the SGD algorithm allows us to think of the optimization scheme as an online algorithm. Instead of randomly selecting an index to compute the gradient, assume that the data points (i.e., in the above example the $\vec{p}_i$) that define the functions $f_i(\vec{x})$ arrive one at a time. Following the same idea of SGD, we can perform an optimization step as each new data point becomes available. In this setting it is important to normalize by the number of data points m in the objective function; otherwise the objective function will keep growing as we get more data, irregardless of whether our optimization algorithm finds a good solution.

6.4 Variations: Gradient Descent for Constrained Optimization

So far we have seen how gradient descent can be applied to different problems all of which are unconstrained. There are variants of the gradient descent algorithm that can be used to solve problems with *simple* constraints. Let $\Omega \subseteq \mathbb{R}^n$ be a *convex* set, let $f \colon \Omega \to \mathbb{R}$ be a differentiable function, and consider the problem

$$p^\star = \min_{\vec{x} \in \Omega} f(\vec{x}). \tag{6.94}$$

For these types of problems we want to limit our search to points inside the set Ω. If we start with an initial guess $\vec{x}_0 \in \Omega$ and apply the (unconstrained) gradient descent algorithm

$$\vec{x}_{t+1} = \vec{x}_t - \eta \nabla f(\vec{x}), \tag{6.95}$$

the point $\vec{x}_{t+1}$ might end up outside of the feasible set Ω, even when $\vec{x}_t$ is in Ω. In this section we will consider two variants of the gradient descent algorithm that propose techniques to deal with constraints.

6.4.1 Projected Gradient Descent

This first variant is projected gradient descent, proposes the idea of *projecting* the point back into the feasible set.

Definition 154 (Projection onto a Convex Set)
Let Ω be a closed[a] convex set. Let $\vec{y} \in \mathbb{R}^n$ be any vector. We define the projection of the vector $\vec{y}$ onto the set Ω as

$$\operatorname{proj}_\Omega(\vec{y}) = \operatorname*{argmin}_{\vec{x}\in\Omega} \|\vec{x} - \vec{y}\|_2^2 . \tag{6.96}$$

[a]A closed set is one that contains its boundary points.

One can show that, since Ω is closed and convex, the projection is unique. In words, $\operatorname{proj}_\Omega(\vec{y})$ is the closest point in Ω to $\vec{y}$. From this definition one sees that if $\vec{y} \in \Omega$ then $\operatorname{proj}_\Omega(\vec{y}) = \vec{y}$. Using this definition we can write the step of the projected gradient descent algorithm as

$$\vec{x}_{t+1} = \operatorname{proj}_\Omega(\vec{x}_t - \eta\nabla f(\vec{x}_t)). \tag{6.97}$$

Algorithm 4 Projected Gradient Descent

```
1: function ProjectedGradientDescent($f, \vec{x}_0, \Omega, \eta, T$)
2:     for $t = 0, 1, \ldots, T-1$ do
3:         $\vec{x}_{t+1} \leftarrow \operatorname{proj}_\Omega(\vec{x}_t - \eta\nabla f(\vec{x}_t))$
4:     end for
5:     return $\vec{x}_T$
6: end function
```

Note that the projection operator itself solves a convex optimization problem. It is only meaningful to consider the projected gradient descent algorithm if this projection problem is simple enough to be solved in every iteration. We have seen examples of projections onto simple sets. For example, the least squares problem computes the projection of a vector onto the subspace $\mathcal{R}(A)$, and we know how to efficiently solve this projection problem. However, that's not always the case, and the projection problem might be nearly as difficult to solve as our original optimization problem.

6.4.2 Conditional Gradient Descent

We introduce another variant of the gradient descent algorithm for constrained problems called conditional gradient descent (also known as Frank-Wolfe method). Recall that in our development of the choice of the "best" search direction for the (regular) gradient descent algorithm, we considered the quantity

$$D_{\vec{v}_t} f(\vec{x}_t) = \lim_{\eta\to 0} \frac{f(\vec{x}_t + \eta\vec{v}_t) - f(\vec{x}_t)}{\eta} = [\nabla f(\vec{x}_t)]^\top \vec{v}_t. \tag{6.98}$$

This quantity, known as the directional derivative, is the (instantaneous) rate of change of the function f at point $\vec{x}_t$ along direction $\vec{v}_t$. We also saw that the choice $\vec{v}_t = -\nabla f(\vec{x}_t)$ is the direction that minimizes this rate of change. This is a reasonable choice for the unconstrained case since we can freely move in any direction. The idea behind the conditional gradient descent algorithm is to limit our search direction to the feasible set Ω and find a vector inside the set that minimizes $\nabla\left[f(\vec{x}_t)\right]^\top \vec{v}_t$. Thus, given $\vec{x}_t \in \Omega$, we can define the search direction of the conditional gradient descent as

$$\vec{v}_t = \underset{\vec{v}\in\Omega}{\operatorname{argmin}}[\nabla f(\vec{x}_t)]^\top \vec{v}. \tag{6.99}$$

Once we find this direction $\vec{v}_t$, we need to use it in a way that ensures that we don't leave the set. Taking a step along this direction $\vec{x}_{t+1} = \vec{x}_t + \eta\vec{v}_t$ doesn't guarantee this. But we can do something different and take the convex combination between the current point and the search direction. That is, for $\delta_t \in [0, 1]$ we can update

$$\vec{x}_{t+1} = (1 - \delta_t)\vec{x}_t + \delta_t\vec{v}_t \tag{6.100}$$

By convexity of the set, this guarantees that if we start with a feasible initial guess $\vec{x}_0 \in \Omega$, we get a sequence of points $\vec{x}_t \in \Omega$ for all $t \geq 0$. While this property holds for any choice of $\delta_t \in [0, 1]$, to prove convergence of the algorithm we require $\lim_{t\to\infty} \delta_t = 0$; a conventional choice is $\delta_t = \frac{1}{t}$.

Algorithm 5 Conditional Gradient Descent (Frank-Wolfe)

```
1: function ConditionalGradientDescent($f, \vec{x}_0, \Omega, \delta, T$)
2:     for $t = 0, 1, \ldots, T-1$ do
3:         $\vec{v}_t \leftarrow \operatorname{argmin}_{\vec{v}\in\Omega}[\nabla f(\vec{x}_t)]^\top \vec{v}$
4:         $\vec{x}_{t+1} \leftarrow (1-\delta_t)\vec{x}_t + \delta_t\vec{v}_t$
5:     end for
6:     return $\vec{x}_T$
7: end function
```

As a last note, we point out that the problem of finding a search direction $\vec{v}_t$ is a constrained optimization problem within Ω.

Chapter 7

Duality

Relevant sections of the textbooks:

- [1] Chapter 5.
- [2] Section 8.5.

7.1 Lagrangian

This chapter develops the theory of duality for optimization problems. This is a technique that can help us solve or bound the optimal values for constrained optimization problems by solving the related dual problem. The dual problem might not always give us a direct solution to our original ("primal") problem, but it will always give us a bound. For this, we first define the Lagrangian for a problem.

Let us start with our generic constrained optimization problem, which we will call problem $\mathcal{P}$ (which stands for *primal*):

$$\begin{aligned} \text{problem } \mathcal{P}: \qquad p^\star = \min_{\vec{x} \in \mathbb{R}^n} \quad & f_0(\vec{x}) \\ \text{s.t.} \quad & f_i(\vec{x}) \leq 0, \qquad \forall i \in \{1, \ldots, m\} \\ & h_j(\vec{x}) = 0, \qquad \forall j \in \{1, \ldots, p\}. \end{aligned} \tag{7.1}$$

Let us denote its feasible set by

$$\Omega \doteq \left\{ \vec{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} f_i(\vec{x}) \leq 0, & \forall i \in \{1, \ldots, m\} \\ h_j(\vec{x}) = 0, & \forall j \in \{1, \ldots, p\} \end{array} \right\} \qquad \text{so that} \qquad p^\star = \min_{\vec{x} \in \Omega} f_0(\vec{x}). \tag{7.2}$$

We know how to algorithmically approach unconstrained problems, say for instance using gradient descent, which we discussed in the previous chapter. Thus, our question is whether there is a way to incorporate the constraints of the problem $\mathcal{P}$ into the objective function itself. Trying to understand if this is possible is one motivation for the Lagrangian.

To this end, we construct the *indicator function* $\mathbb{1}[\cdot]$, which for a condition $C(\vec{x})$ defined as

$$\mathbb{1}[C(\vec{x})] \doteq \begin{cases} 0, & \text{if } C(\vec{x}) \text{ is true,} \\ +\infty, & \text{if } C(\vec{x}) \text{ is false.} \end{cases} \tag{7.3}$$

Then the problem $\mathcal{P}$ in (7.1) is equivalent to the following *unconstrained* problem:

$$
\begin{aligned}
p^\star &= \min_{\vec{x}\in\Omega} f_0(\vec{x}) && (7.4)\\
&= \min_{\vec{x}\in\mathbb{R}^n} \{f_0(\vec{x}) + \mathbb{1}[\vec{x}\in\Omega]\} && (7.5)\\
&= \min_{\vec{x}\in\mathbb{R}^n} \left\{ f_0(\vec{x}) + \sum_{i=1}^{m} \mathbb{1}[f_i(\vec{x}) \le 0] + \sum_{j=1}^{p} \mathbb{1}[h_j(\vec{x}) = 0] \right\}. && (7.6)
\end{aligned}
$$

To explain the chain of equalities leading to (7.6), say we consider the function

$$
F_0(\vec{x}) \doteq f_0(\vec{x}) + \sum_{i=1}^{m} \mathbb{1}[f_i(\vec{x}) \le 0] + \sum_{j=1}^{p} \mathbb{1}[h_j(\vec{x}) = 0] \qquad (7.7)
$$

evaluated at an $\vec{x} \notin \Omega$, i.e., there is some i such that $f_i(\vec{x}) > 0$ or some j such that $h_j(\vec{x}) \neq 0$. Then $F_0(\vec{x}) = \infty$. Therefore no solution to the minimization in (7.6) will ever be outside of Ω, i.e., all solutions to (7.6) will be contained in Ω. And for $\vec{x} \in \Omega$, we have $F_0(\vec{x}) = f_0(\vec{x})$, so that $\min_{\vec{x}\in\Omega} f_0(\vec{x}) = \min_{\vec{x}\in\mathbb{R}^n} F_0(\vec{x})$ (and the argmins are equal too). Hence the problem in (7.6) is equivalent to the original problem $\mathcal{P}$.

Thus through this reformulation, we have removed the explicit constraints of the problem and incorporated them into the objective function of the problem, and we can nominally write the problem as an unconstrained problem. If we had some magic algorithm which allowed us to solve *all* unconstrained problems, then this reduction would let us solve constrained problems as well.

Unfortunately, we do *not* have such a magic algorithm. Our usual algorithms for solving unconstrained problems, such as gradient descent, require differentiable objective functions that are well-defined. Thus, it falls to us to express our indicator functions in a differentiable way.

For this we consider *approximating* the indicator functions by other functions that are differentiable.

The key idea here is that, for a given indicator $\mathbb{1}[f_i(\vec{x}) \le 0]$, we can express it as

$$
\mathbb{1}[f_i(\vec{x}) \le 0] = \max_{\lambda_i \in \mathbb{R}_+} \lambda_i f_i(\vec{x}), \qquad (7.8)
$$

where $\mathbb{R}_+$ is the set of non-negative real numbers. Why does this equality hold? Let's break it up into cases. Suppose that $f_i(\vec{x}) > 0$. Then λ_i being more positive would make $\lambda_i f_i(\vec{x})$ more positive, so the maximization will send $\lambda_i \to \infty$, making $\lambda_i f_i(\vec{x}) = \infty$. Now suppose that $f_i(\vec{x}) \le 0$. Then λ_i being more positive would make $\lambda_i f_i(\vec{x})$ more negative, so the maximization will keep λ_i as its lower bound 0, making $\lambda_i f_i(\vec{x}) = 0$. For a similar reason, for an indicator $\mathbb{1}[h_j(\vec{x}) = 0]$, we have

$$
\mathbb{1}[h_j(\vec{x}) = 0] = \max_{\nu_j \in \mathbb{R}} \nu_j h_j(\vec{x}). \qquad (7.9)
$$

Thus, we can rewrite the problem $\mathcal{P}$ as

$$
\begin{aligned}
p^\star &= \min_{\vec{x}\in\mathbb{R}^n} \left\{ f_0(\vec{x}) + \sum_{i=1}^{m} \max_{\lambda_i\in\mathbb{R}_+} \lambda_i f_i(\vec{x}) + \sum_{j=1}^{p} \max_{\nu_j\in\mathbb{R}} \nu_j h_j(\vec{x}) \right\} && (7.10)\\
&= \min_{\vec{x}\in\mathbb{R}^n} \left\{ f_0(\vec{x}) + \max_{\vec{\lambda}\in\mathbb{R}_+^m} \sum_{i=1}^{m} \lambda_i f_i(\vec{x}) + \max_{\vec{\nu}\in\mathbb{R}^p} \sum_{j=1}^{p} \nu_j h_j(\vec{x}) \right\} && (7.11)\\
&= \min_{\vec{x}\in\mathbb{R}^n} \max_{\substack{\vec{\lambda}\in\mathbb{R}_+^m \\ \vec{\nu}\in\mathbb{R}^p}} \left\{ f_0(\vec{x}) + \sum_{i=1}^{m} \lambda_i f_i(\vec{x}) + \sum_{j=1}^{p} \nu_j h_j(\vec{x}) \right\}. && (7.12)
\end{aligned}
$$

This interior function is called the *Lagrangian*, and is much easier to work with than the original unconstrained problem with indicator functions. Thus we have made our constrained problem into a (mostly) unconstrained problem,[1] at the expense of adding an extra $\max$. We will see how to cope with this extra difficulty shortly.

More formally, we have the following definition:

Definition 155 (Lagrangian)
The *Lagrangian* of problem $\mathcal{P}$ in (7.1) is the function $L\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ given by

$$L(\vec{x}, \vec{\lambda}, \vec{\nu}) \doteq f_0(\vec{x}) + \sum_{i=1}^{m} \lambda_i f_i(\vec{x}) + \sum_{j=1}^{p} \nu_j h_j(\vec{x}). \tag{7.13}$$

Additionally, $\lambda_i, \nu_j \in \mathbb{R}$ are called *Lagrange multipliers*.

The important intuition behind the Lagrange multipliers is that they are *penalties* for violating their corresponding constraint. In particular, just like how the indicator function $\mathbb{1}[f_i(\vec{x}) \leq 0]$ assigns an infinite penalty for violating the constraint $f_i(\vec{x}) \leq 0$, the term $\lambda_i f_i(\vec{x})$ assigns a penalty $\lambda_i f_i(\vec{x})$ for violating the constraint $f_i(\vec{x}) \leq 0$. One can show that $\lambda_i f_i(\vec{x}) \leq \mathbb{1}[f_i(\vec{x}) \leq 0]$, so the Lagrangian penalty is a smooth lower bound to the indicator penalty, which is something like a hard-threshold. We can do similar analysis for the equality constraints $h_j(\vec{x}) = 0$.

As derived before, the primal problem can be expressed in terms of the Lagrangian as

$$p^\star = \min_{\vec{x} \in \mathbb{R}^n} \max_{\substack{\vec{\lambda} \in \mathbb{R}^m_+ \\ \vec{\nu} \in \mathbb{R}^p}} L(\vec{x}, \vec{\lambda}, \vec{\nu}). \tag{7.14}$$

We conclude with two very important properties of the Lagrangian.

Proposition 156
For every $\vec{x} \in \mathbb{R}^n$, the function $(\vec{\lambda}, \vec{\nu}) \mapsto L(\vec{x}, \vec{\lambda}, \vec{\nu})$ is an affine function, and hence a concave function. (We also say that L is affine (resp. concave) in $\vec{\lambda}$ and $\vec{\nu}$.)

Proof. The proof follows from the definition of the Lagrangian:

$$L(\vec{x}, \vec{\lambda}, \vec{\nu}) = f_0(\vec{x}) + \sum_{i=1}^{m} \lambda_i f_i(\vec{x}) + \sum_{j=1}^{p} \nu_j h_j(\vec{x}) \tag{7.15}$$

is affine in $\vec{\lambda}$ and $\vec{\nu}$. □

Example 157. Consider the optimization problem

$$\begin{aligned} p^\star = \min_{x \in \mathbb{R}} \quad & 3x^2 \\ \text{s.t.} \quad & 2x^3 \leq 8. \end{aligned} \tag{7.16}$$

This problem is not convex, but we can still find its Lagrangian as

$$L(x, \lambda) = 3x^2 + \lambda(2x^3 - 8). \tag{7.17}$$

[1]Not quite — the $\max$ is actually over $\mathbb{R}^m_+ \times \mathbb{R}^p$ which is technically a constrained set. But this non-negativity constraint is usually much easier to handle than arbitrary constraints, in part because the constraint set is convex.

7.2 Weak Duality

Our foray into duality starts by considering the primal problem:

$$p^\star = \min_{\vec{x}\in\mathbb{R}^n} \max_{\substack{\vec{\lambda}\in\mathbb{R}^m_+ \\ \vec{\nu}\in\mathbb{R}^p}} L(\vec{x}, \vec{\lambda}, \vec{\nu}). \tag{7.18}$$

The *dual problem* is obtained by swapping the min and max:

$$d^\star = \max_{\substack{\vec{\lambda}\in\mathbb{R}^m_+ \\ \vec{\nu}\in\mathbb{R}^p}} \min_{\vec{x}\in\mathbb{R}^n} L(\vec{x}, \vec{\lambda}, \vec{\nu}). \tag{7.19}$$

In general, the quantities $p^\star$ and $d^\star$ *do not have to be equal*! And swapping the min and the max is the generic definition of "duality".

Definition 158 (Dual Problem)
For a primal problem $\mathcal{P}$ as described in (7.1), its *dual problem* $\mathcal{D}$ is defined as

$$\begin{aligned} \text{problem } \mathcal{D}: \qquad d^\star = \max_{\substack{\lambda\in\mathbb{R}^m \\ \nu\in\mathbb{R}^p}} \quad & g(\vec{\lambda}, \vec{\nu}) \\ \text{s.t.} \quad & \lambda_i \geq 0, \qquad \forall i \in \{1, \ldots, m\}. \end{aligned} \tag{7.20}$$

Here the function $g\colon \mathbb{R}^m_+ \times \mathbb{R}^p \to \mathbb{R}$ is the *dual function* and defined as

$$g(\vec{\lambda}, \vec{\nu}) = \min_{\vec{x}\in\mathbb{R}^n} L(\vec{x}, \vec{\lambda}, \vec{\nu}). \tag{7.21}$$

Thus, $g(\vec{\lambda}, \vec{\nu})$ can be computed as an unconstrained optimization problem over $\vec{x}$, in particular $g(\vec{\lambda}, \vec{\nu})$ is the minimum value of $L(\vec{x}, \vec{\lambda}, \vec{\nu})$ over all $\vec{x}$.

There are several important properties of the dual problem and dual function, which we summarize below.

Proposition 159
The dual function g is a concave function of $(\vec{\lambda}, \vec{\nu})$, regardless of any properties of $\mathcal{P}$.

Proof. We already showed that $L(\vec{x}, \vec{\lambda}, \vec{\nu})$ is an affine (and thus concave) function of $\vec{\lambda}$ and $\vec{\nu}$. Thus the function

$$g(\vec{\lambda}, \vec{\nu}) = \min_{\vec{x}\in\mathbb{R}^n} L(\vec{\lambda}, \vec{\nu}) \tag{7.22}$$

is a pointwise minimum of concave functions and is thus concave. □

Corollary 160. *The dual problem $\mathcal{D}$ is always a convex problem, no matter what the primal problem $\mathcal{P}$ is.*

Note that the minimization of a convex function or the maximization of a concave function are both considered "convex" problems.

This means we have analytic and algorithmic ways to solve the dual problem $\mathcal{D}$. If we can connect the solutions to the dual problem $\mathcal{D}$ to the primal problem $\mathcal{P}$, this means that we have reduced the process of solving $\mathcal{P}$ to the process of solving a convex optimization problem, and this is something we know how to do. In the rest of this chapter, we discuss when this reduction is possible.

Example 161. Let us compute the dual of the optimization problem

$$\begin{aligned} p^\star = \min_{x \in \mathbb{R}} \quad & 5x^2 \\ \text{s.t.} \quad & 3x \leq 5. \end{aligned} \tag{7.23}$$

The Lagrangian is

$$L(x, \lambda) = 5x^2 + \lambda(3x - 5). \tag{7.24}$$

The dual function is

$$g(\lambda) = \min_x L(x, \lambda) = \min_x \left\{5x^2 + \lambda(3x - 5)\right\}. \tag{7.25}$$

For each $\lambda > 0$, the function $L(x, \lambda) = 5x^2 + \lambda(3x - 5)$ is bounded below (in x), convex, and differentiable, so to minimize it we set its derivative to 0. The optimal $x^\star$ is a function of the Lagrange multiplier λ, so we write it as $x^\star(\lambda)$. We have

$$\begin{aligned} 0 &= \nabla_x L(x^\star(\lambda), \lambda) \qquad (7.26) \\ &= 10x^\star(\lambda) + 3\lambda \qquad (7.27) \\ \implies x^\star(\lambda) &= -\frac{3}{10}\lambda. \qquad (7.28) \end{aligned}$$

Thus we can plug this optimal point back in to get g:

$$\begin{aligned} g(\lambda) &= L(x^\star(\lambda), \lambda) \qquad (7.29) \\ &= 5(x^\star(\lambda))^2 + \lambda(3x^\star(\lambda) - 5) \qquad (7.30) \\ &= \frac{5 \cdot 9}{100}\lambda^2 + \lambda\left(-\frac{3 \cdot 3}{10}\lambda - 5\right) \qquad (7.31) \\ &= -\frac{9}{20}\lambda^2 - 5\lambda. \qquad (7.32) \end{aligned}$$

Thus the dual problem is

$$d^\star = \max_{\lambda \in \mathbb{R}_+} \left(-\frac{9}{20}\lambda^2 - 5\lambda\right). \tag{7.33}$$

One may solve this problem directly to obtain $\lambda^\star = 0$ and $d^\star = 0$.

We also have some more bounds for the Lagrangian in terms of f_0, g, $p^\star$, and $d^\star$.

Proposition 162

Let $\vec{x} \in \Omega$, let $\vec{\lambda} \in \mathbb{R}^m_+$, and let $\vec{\nu} \in \mathbb{R}^p$, so that $\vec{x}$ is feasible for $\mathcal{P}$ and $(\vec{\lambda}, \vec{\nu})$ is feasible for $\mathcal{D}$. Then we have:

(a) $f_0(\vec{x}) \geq p^\star$ and $g(\vec{\lambda}, \vec{\nu}) \leq d^\star$;

(b) $f_0(\vec{x}) \geq L(\vec{x}, \vec{\lambda}, \vec{\nu}) \geq g(\vec{\lambda}, \vec{\nu})$;

(c) $f_0(\vec{x}) \geq d^\star$ and $g(\vec{\lambda}, \vec{\nu}) \leq p^\star$.

Proof. The inequalities in (a) follow from the characterizations

$$p^\star = \min_{\vec{x}' \in \Omega} f_0(\vec{x}') \leq f_0(\vec{x}) \tag{7.34}$$

$$d^\star = \max_{\substack{\vec{\lambda}' \in \mathbb{R}^m_+ \\ \vec{\nu}' \in \mathbb{R}^p}} g(\vec{\lambda}', \vec{\nu}') \geq g(\vec{\lambda}, \vec{\nu}). \tag{7.35}$$

The inequalities in (b) follow from the characterizations

$$g(\vec{\lambda}, \vec{\nu}) = \min_{\vec{x}' \in \Omega} L(\vec{x}', \vec{\lambda}, \vec{\nu}) \leq L(\vec{x}, \vec{\lambda}, \vec{\nu}) \tag{7.36}$$

$$f_0(\vec{x}) = \max_{\substack{\vec{\lambda}' \in \mathbb{R}^m_+ \\ \vec{\nu}' \in \mathbb{R}^p}} L(\vec{x}, \vec{\lambda}', \vec{\nu}') \geq L(\vec{x}, \vec{\lambda}, \vec{\nu}). \tag{7.37}$$

The inequalities in (c) follow from (b), in the sense that

$$f_0(\vec{x}) \geq g(\vec{\lambda}, \vec{\nu}) \qquad \forall \vec{\lambda} \in \mathbb{R}^m_+ \qquad \forall \vec{\nu} \in \mathbb{R}^p \tag{7.38}$$

$$\implies f_0(\vec{x}) \geq \max_{\substack{\vec{\lambda}' \in \mathbb{R}^m_+ \\ \vec{\nu}' \in \mathbb{R}^p}} g(\vec{\lambda}', \vec{\nu}') = d^\star, \tag{7.39}$$

and additionally

$$f_0(\vec{x}) \geq g(\vec{\lambda}, \vec{\nu}) \qquad \forall \vec{x} \in \Omega \tag{7.40}$$

$$\implies p^\star = \min_{\vec{x}' \in \Omega} f_0(\vec{x}') \geq g(\vec{\lambda}, \vec{\nu}). \tag{7.41}$$

□

From all these inequalities, the relationship between $p^\star$ and $d^\star$ is totally unclear. Actually, their relationship is very simple; it is captured in a condition called "weak duality".

Definition 163 (Types of Duality)
Let $\mathcal{P}$ be a primal problem with optimum $p^\star$. Let $\mathcal{D}$ be the corresponding dual problem with optimum $d^\star$.

(a) If $p^\star \geq d^\star$ then we say that *weak duality* holds.

(b) If $p^\star = d^\star$ then we say that *strong duality* holds.

(c) The quantity $p^\star - d^\star$ is called the *duality gap*.

Proposition 164 (Weak Duality Always Holds)
For *any* problem, weak duality holds, i.e., the duality gap is non-negative.

The fact that weak duality always holds is actually a corollary of a slightly more generic result called the minimax inequality, which we now state and prove.

Proposition 165 (Minimax Inequality)
Let X and Y be *any* sets, and $F\colon X \times Y \to \mathbb{R}$ be *any* function. Then

$$\min_{x \in X} \max_{y \in Y} F(x, y) \geq \max_{y \in Y} \min_{x \in X} F(x, y). \tag{7.42}$$

Proof. For any $x \in X$ and $y \in Y$, we have

$$F(x, y) \geq \min_{x' \in X} F(x', y). \tag{7.43}$$

Taking the maximum over y on both sides (we will discuss why we can do this at the end of the proof), we have

$$\max_{y' \in Y} F(x, y') \geq \max_{y' \in Y} \min_{x' \in X} F(x', y'). \tag{7.44}$$

Finally, the left hand side is a function of x but the right hand side is a constant, so we might as well take the minimum in x on the left hand side to get

$$\min_{x' \in X} \max_{y' \in Y} F(x', y') \geq \max_{y' \in Y} \min_{x' \in X} F(x', y'). \tag{7.45}$$

This concludes the (main part of the) proof.

However, we still need to discuss why we can take maximums on both sides in Equation (7.44). We prove this in the case where the maximums are attained, though it is true in general (and can be proved via a slightly technical modification to this argument). Fix $x \in X$. Let $f\colon Y \to \mathbb{R}$ be defined as $f(y) \doteq F(x, y)$, and let $g\colon Y \to \mathbb{R}$ be defined as $g(y) \doteq \min_{x' \in X} F(x', y)$. Then we had shown that $f(y) \geq g(y)$ for all $y \in Y$, and we want to justify why this means that $\max_{y' \in Y} f(y') \geq \max_{y' \in Y} g(y')$. Towards this end, let $\widetilde{y} \in \operatorname{argmax}_{y' \in Y} g(y')$. Then we have

$$\max_{y' \in Y} f(y') \geq f(\widetilde{y}) \geq g(\widetilde{y}) = \max_{y' \in Y} g(y') \implies \max_{y' \in Y} F(x, y') \geq \max_{y' \in Y} \min_{x' \in X} F(x', y') \text{ for all } x \in X, \tag{7.46}$$

as desired. □

Here is a game theoretic interpretation of the minimax inequality.

Suppose X and Y are two players choosing actions x and y. Player X chooses action x trying to *minimize* the value of the function $F(x, y)$. Player Y chooses action y trying to *maximize* the value of the function $F(x, y)$. The two players are perfectly rational and take turns — that is, the players choose their actions one after the other, and the second player has full knowledge of the first player's action (which is locked in) as they choose their action. (In game theory this is called a two-player zero-sum sequential game.) The value of the game, given actions x and y, is $F(x, y)$.

The outer optimization corresponds to the player going first, and the inner minimization corresponds to the player going second. Why? Consider the optimization problem $\min_{x \in X} \max_{y \in Y} F(x, y)$. Then defining $G(x) = \max_{y \in Y} F(x, y)$, we see that given a particular x, $G(x)$ is the maximum value obtained by varying y of $F(x, y)$. That is, it corresponds to the best play by player Y given that player X picks action x. This means that the inner minimization corresponds to the second player Y's actions given that the first player X chooses action x. And so the outer optimization corresponds to the first player X's action. The situation is analogous on the right-hand side.

What the minimax inequality says is that *going second is better*. That is, being the second player and choosing your action with full knowledge of the first player's action leads to a better outcome for you than going first with no knowledge of the second player's action. When X is the second player, the value of the game is lower than when X is the first player; when Y is the second player, the value of the game is higher than when Y is the first player.

Given the minimax inequality, the proof that weak duality always holds is only one line.

Proof that weak duality always holds. We have

$$p^\star = \min_{\vec{x} \in \mathbb{R}^n} \max_{\substack{\vec{\lambda} \in \mathbb{R}^m_+ \\ \vec{\nu} \in \mathbb{R}^p}} L(\vec{x}, \vec{\lambda}, \vec{\nu}) \geq \max_{\substack{\vec{\lambda} \in \mathbb{R}^m_+ \\ \vec{\nu} \in \mathbb{R}^p}} \min_{\vec{x} \in \mathbb{R}^n} L(\vec{x}, \vec{\lambda}, \vec{\nu}) = d^\star. \tag{7.47}$$

□

Weak duality is useful because it gives a bound on how close to optimum a given decision variable is. More specifically, we have that for all $\vec{x} \in \Omega$ and all $(\vec{\lambda}, \vec{\nu}) \in \mathbb{R}^m_+ \times \mathbb{R}^p$ that

$$f_0(\vec{x}) \geq p^\star \geq d^\star \geq g(\vec{\lambda}, \vec{\nu}) \implies f_0(\vec{x}) - g(\vec{\lambda}, \vec{\nu}) \geq f_0(\vec{x}) - p^\star. \tag{7.48}$$

Thus, if $f_0(\vec{x}) - g(\vec{\lambda}, \vec{\nu}) \leq \epsilon$ then we know that $f_0(\vec{x}) - p^\star \leq \epsilon$. This is called a *certificate of optimality*. We can use this certificate as a stopping condition for algorithms such as gradient descent or a broad class of algorithms known as *primal-dual* algorithms.

7.3 Strong Duality

Since weak duality holds for all problems, it necessarily would not have too many critical implications. The stronger condition of *strong duality* turns out to hold the keys to efficiently computing minima for constrained optimization problems.

First, a natural question is: when does strong duality hold? It does not *always* hold, and in particular it is a stronger condition than weak duality. You will see some example problems where strong duality does not hold in discussion.

There are many conditions under which strong duality holds. In this course, we discuss one such condition which is relatively simple to evaluate and broadly applicable. This condition is called *Slater's condition*.

Theorem 166 (Slater's Condition)
Suppose that $f_0, f_1, \ldots, f_m \colon \mathbb{R}^n \to \mathbb{R}$ are convex functions, and $h_1, \ldots, h_p \colon \mathbb{R}^n \to \mathbb{R}$ are affine functions. Consider our primal problem $\mathcal{P}$, which has feasible set Ω. If there exists any $\vec{x} \in \operatorname{relint}(\Omega)$[a] which is *strictly feasible*, i.e., such that all of the following hold:

- for all $i \in \{1, \ldots, m\}$ such that f_i is an *affine* function, we have $f_i(\vec{x}) \leq 0$;
- for all $i \in \{1, \ldots, m\}$ such that f_i is *not* an affine function, we have $f_i(\vec{x}) < 0$;
- and for all $j \in \{1, \ldots, p\}$, we have $h_j(\vec{x}) = 0$;

then strong duality holds for $\mathcal{P}$ and its dual $\mathcal{D}$, i.e., the duality gap is 0.

[a]Recall that this notation means the *relative interior* of the feasible set Ω, defined formally in Definition 102. Since the formal definition of the relative interior is out of scope for this semester, you may just consider it to be the interior of Ω, i.e., points not on the boundary of Ω, and we will not give you problems where the distinction matters.

This result is sometimes called *refined Slater's condition* as it is a slight modification of another condition which is also called Slater's condition.

We will not prove this in class; a proof sketch is in [1]. It uses the separating hyperplane theorem we proved earlier. This result is actually one of the main payoffs of the separating hyperplane theorem that we will see in this course.

Again, observe that we *only* need a *single* point which fulfills the three conditions in order to declare that Slater's condition holds. Moreover, this point *need not be optimal* for the problem — any point at all which satisfies the conditions will do.

Slater's condition has a geometric interpretation: if there is a point in the *relative interior* of the feasible set which is *strictly feasible*, then strong duality holds. In problems we are able to visualize, this makes it relatively easy to determine whether Slater's condition holds.

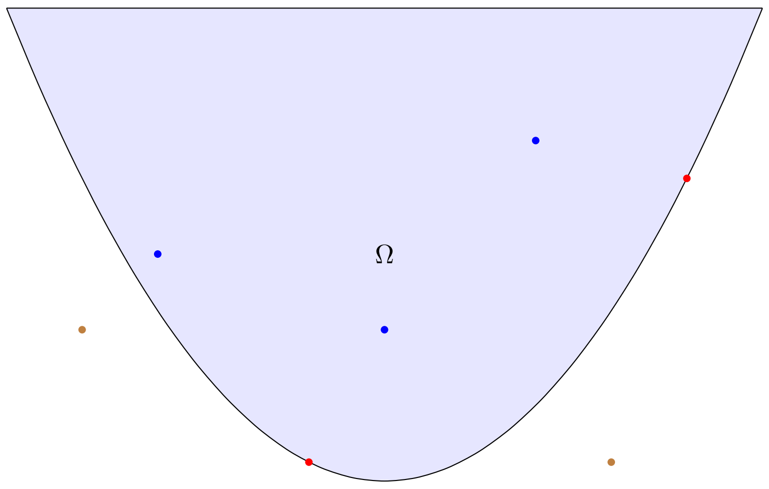

Figure 7.1: Points in the convex set $\Omega = \{(x, y) \in \mathbb{R}^2 \colon -5 \le x \le 5, \frac{25}{4} \ge y \ge \frac{1}{4}x^2\}$. Blue points are feasible points which fulfill the conditions of Slater's condition (thus ensuring that Slater's condition holds for the problem $\min_{\vec{x}\in\Omega} f_0(\vec{x})$ provided that f_0 is convex). Red points are feasible points which do not fulfill the conditions of Slater's condition. Brown points are infeasible points.

Example 167. In this example, we consider the equality-constrained minimum-norm problem

$$\begin{aligned} p^\star = \min_{\vec{x}\in\mathbb{R}^n} \quad & \|\vec{x}\|_2^2 \\ \text{s.t.} \quad & A\vec{x} = \vec{y}. \end{aligned} \tag{7.49}$$

This problem only has equality constraints, and so it only has the Lagrange multiplier $\vec{\nu}$. The Lagrangian of this problem is

$$L(\vec{x}, \vec{\nu}) = \|\vec{x}\|_2^2 + \sum_{j=1}^{p} \nu_j (A\vec{x} - \vec{y})_j \tag{7.50}$$

$$= \|\vec{x}\|_2^2 + \vec{\nu}^\top (A\vec{x} - \vec{y}). \tag{7.51}$$

The dual function is

$$g(\vec{\nu}) = \min_{\vec{x}\in\mathbb{R}^n} L(\vec{x}, \vec{\nu}) \tag{7.52}$$

$$= \min_{\vec{x}\in\mathbb{R}^n} \left\{ \|\vec{x}\|_2^2 + \vec{\nu}^\top (A\vec{x} - \vec{y}) \right\}. \tag{7.53}$$

The Lagrangian is convex in $\vec{x}$ and bounded below, and so it is minimized wherever its gradient in $\vec{x}$ is $\vec{0}$. The optimal $\vec{x}^\star$ is a function of $\vec{\nu}$, so we write it as $\vec{x}^\star(\vec{\nu})$. We have

$$\vec{0} = \nabla_{\vec{x}} L(\vec{x}^\star(\vec{\nu}), \vec{\nu}) \tag{7.54}$$

$$= 2\vec{x}^\star(\vec{\nu}) + A^\top \vec{\nu} \tag{7.55}$$

$$\implies \vec{x}^\star(\vec{\nu}) = -\frac{1}{2} A^\top \vec{\nu}. \tag{7.56}$$

Thus we can plug this optimal point back in to get g:

$$g(\vec{\nu}) = L(\vec{x}^\star(\vec{\nu}), \vec{\nu}) \tag{7.57}$$

$$= \|\vec{x}^\star(\vec{\nu})\|_2^2 + \vec{\nu}^\top (A\vec{x}^\star(\vec{\nu}) - \vec{y}) \tag{7.58}$$

$$= \left\| -\frac{1}{2}A^\top \vec{\nu} \right\|_2^2 + \vec{\nu}^\top \left(A\left(-\frac{1}{2}A^\top \vec{\nu}\right) - \vec{y} \right) \tag{7.59}$$

$$= \frac{1}{4}\vec{\nu}^\top AA^\top \vec{\nu} - \frac{1}{2}\vec{\nu}^\top AA^\top \vec{\nu} - \vec{\nu}^\top \vec{y} \tag{7.60}$$

$$= -\frac{1}{4}\vec{\nu}^\top AA^\top \vec{\nu} - \vec{\nu}^\top \vec{y}. \tag{7.61}$$

Thus the dual problem is

$$d^\star = \max_{\vec{\nu} \in \mathbb{R}^p} \left\{ -\frac{1}{4}\vec{\nu}^\top AA^\top \vec{\nu} - \vec{\nu}^\top \vec{y} \right\} = -\min_{\vec{\nu} \in \mathbb{R}^p} \left\{ \frac{1}{4}\vec{\nu}^\top AA^\top \vec{\nu} + \vec{\nu}^\top \vec{y} \right\}. \tag{7.62}$$

There are no constraints on the dual problem, because there are no inequality constraints on the primal problem. Since this dual problem is a convex unconstrained problem whose objective value is bounded below, it can be solved by setting the gradient of the objective to $\vec{0}$ and solving for the optimal dual variable. This yields

$$\vec{0} = \nabla g(\vec{\nu}^\star) \tag{7.63}$$

$$= \frac{1}{2}AA^\top \vec{\nu}^\star + \vec{y} \tag{7.64}$$

$$\implies \vec{\nu}^\star = -2(AA^\top)^{-1}\vec{y}. \tag{7.65}$$

Our original primal problem is convex, and all constraints are affine. As long as there is a feasible point $\vec{x}$, i.e., a point $\vec{x}$ such that $A\vec{x} = \vec{y}$, then Slater's condition implies that strong duality holds. Suppose that there is such a feasible point (i.e., the primal problem is feasible). Then strong duality holds. Because the primal problem is convex and strong duality holds, we can recover the optimal primal variable $\vec{x}^\star$ from the optimal dual variable $\vec{\nu}^\star$ as

$$\vec{x}^\star = \vec{x}^\star(\vec{\nu}^\star) = -\frac{1}{2}A^\top(-2(AA^\top)^{-1}\vec{y}) = A^\top(AA^\top)^{-1}\vec{y}. \tag{7.66}$$

This corresponds to the familiar minimum-norm solution.

Example 168. Consider a so-called *linear program*:

$$\begin{aligned} p^\star = \min_{\vec{x} \in \mathbb{R}^n} \quad & \vec{c}^\top \vec{x} \\ \text{s.t.} \quad & A\vec{x} = \vec{y} \\ & \vec{x} \geq \vec{0} \end{aligned} \tag{7.67}$$

where the last constraint means $x_i \geq 0$ for all i. Slater's condition implies that strong duality holds for this problem provided there is a feasible point. The Lagrangian is

$$L(\vec{x}, \vec{\lambda}, \vec{\nu}) = \vec{c}^\top \vec{x} + \sum_{i=1}^{n} \lambda_i(-x_i) + \sum_{j=1}^{m} \nu_j (A\vec{x} - \vec{y})_j \tag{7.68}$$

$$= \vec{c}^\top \vec{x} - \vec{\lambda}^\top \vec{x} + \vec{\nu}^\top (A\vec{x} - \vec{y}) \tag{7.69}$$

$$= \left(\vec{c} + A^\top \vec{\nu} - \vec{\lambda}\right)^\top \vec{x} - \vec{\nu}^\top \vec{y}. \tag{7.70}$$

This function is affine in $\vec{x}$, hence convex in $\vec{x}$, so its dual function is

$$g(\vec{\lambda}, \vec{\nu}) = \min_{\vec{x} \in \mathbb{R}^n} L(\vec{x}, \vec{\lambda}, \vec{\nu}) \tag{7.71}$$

$$= \min_{\vec{x} \in \mathbb{R}^n} \left\{ \left(\vec{c} + A^\top \vec{\nu} - \vec{\lambda}\right)^\top \vec{x} - \vec{\nu}^\top \vec{y} \right\} \tag{7.72}$$