Chapter 2
Mathematical Tools

This section will introduce concepts, algorithmic tools, and analysis techniques used in the design and analysis of optimization algorithms. It will also explore simple convex optimization problems which will serve as a warm-up exercise.

2.1 Convex Analysis

We recall some basic definitions in convex analysis. Studying these will help us appreciate the structural properties of non-convex optimization problems later in the monograph. For the sake of simplicity, unless stated otherwise, we will assume that functions are continuously differentiable. We begin with the notion of a convex combination.

Definition 2.1 (Convex Combination). A convex combination of a set of $n$ vectors $x_i \in \mathbb{R}^p$, $i = 1 \ldots n$ in an arbitrary real space is a vector $x_{\theta} := \sum_{i=1}^{n} \theta_i x_i$ where $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, $\theta_i \geq 0$ and $\sum_{i=1}^{n} \theta_i = 1$.

A set that is closed under arbitrary convex combinations is a convex set. A standard definition is given below. Geometrically speaking, convex sets are those that contain all line segments that join two points inside the set. As a result, they cannot have any inward “bulges”.

Definition 2.2 (Convex Set). A set $C \in \mathbb{R}^p$ is considered convex if, for every $x, y \in C$ and $\lambda \in [0, 1]$, we have $(1 - \lambda) \cdot x + \lambda \cdot y \in C$ as well.

Figure 2.1 gives visual representations of prototypical convex and non-convex sets. A related notion is that of convex functions which have a unique behavior under convex combinations. There are several definitions of convex functions, those that are more basic and general, as well as those that are restrictive but easier to use. One of the simplest definitions of convex functions, one that does not involve notions of derivatives, defines convex functions $f : \mathbb{R}^p \to \mathbb{R}$ as those for which, for every $x, y \in \mathbb{R}^p$ and every $\lambda \in [0, 1]$, we have $f((1-\lambda) \cdot x + \lambda \cdot y) \leq (1-\lambda) \cdot f(x) + \lambda \cdot f(y)$.

For continuously differentiable functions, a more usable definition follows.

Definition 2.3 (Convex Function). A continuously differentiable function $f : \mathbb{R}^p \to \mathbb{R}$ is considered convex if for every $x, y \in \mathbb{R}^p$ we have $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$, where $\nabla f(x)$ is the gradient of $f$ at $x$.

A more general definition that extends to non-differentiable functions uses the notion of subgradient to replace the gradient in the above expression. A special class of convex functions is the class of strongly convex and strongly smooth functions. These are critical to the study of algorithms for non-convex optimization. Figure 2.2 provides a handy visual representation of these classes of functions.
2.1. CONVEX ANALYSIS

Figure 2.1: A convex set is closed under convex combinations. The presence of even a single uncontained convex combination makes a set non-convex. Thus, a convex set cannot have inward “bulges”. In particular, the set of sparse vectors is non-convex.

Figure 2.2: A convex function is lower bounded by its own tangent at all points. Strongly convex and smooth functions are, respectively, lower and upper bounded in the rate at which they may grow, by quadratic functions and cannot, again respectively, grow too slowly or too fast. In each figure, the shaded area describes regions the function curve is permitted to pass through.

Definition 2.4 (Strongly Convex/Smooth Function). A continuously differentiable function $f : \mathbb{R}^p \to \mathbb{R}$ is considered $\alpha$-strongly convex (SC) and $\beta$-strongly smooth (SS) if for every $x, y \in \mathbb{R}^p$, we have

$$\frac{\alpha}{2} \|x - y\|_2^2 \leq f(y) - f(x) - \langle \nabla f(x), y - x \rangle \leq \frac{\beta}{2} \|x - y\|_2^2.$$

It is useful to note that strong convexity places a quadratic lower bound on the growth of the function at every point – the function must rise up at least as fast as a quadratic function. How fast it rises is characterized by the SC parameter $\alpha$. Strong smoothness similarly places a quadratic upper bound and does not let the function grow too fast, with the SS parameter $\beta$ dictating the upper limit.

We will soon see that these two properties are extremely useful in forcing optimization algorithms to rapidly converge to optima. Note that whereas strongly convex functions are definitely convex, strong smoothness does not imply convexity\(^1\). Strongly smooth functions

\(^1\)See Exercise 2.1.
may very well be non-convex. A property similar to strong smoothness is that of Lipschitzness which we define below.

**Definition 2.5 (Lipschitz Function).** A function \( f : \mathbb{R}^p \rightarrow \mathbb{R} \) is \( B \)-Lipschitz if for every \( x, y \in \mathbb{R}^p \), we have

\[
| f(x) - f(y) | \leq B \cdot \| x - y \|_2.
\]

Notice that Lipschitzness places a upper bound on the growth of the function that is linear in the perturbation i.e., \( \| x - y \|_2 \), whereas strong smoothness (SS) places a quadratic upper bound. Also notice that Lipschitz functions need not be differentiable. However, differentiable functions with bounded gradients are always Lipschitz\(^2\). Finally, an important property that generalizes the behavior of convex functions on convex combinations is the Jensen’s inequality.

**Lemma 2.1 (Jensen’s Inequality).** If \( X \) is a random variable taking values in the domain of a convex function \( f \), then

\[
\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])
\]

This property will be useful while analyzing iterative algorithms.

### 2.2 Convex Projections

The projected gradient descent technique is a popular method for constrained optimization problems, both convex as well as non-convex. The projection step plays an important role in this technique. Given any closed set \( C \subset \mathbb{R}^p \), the projection operator \( \Pi_C(\cdot) \) is defined as

\[
\Pi_C(z) := \arg \min_{x \in C} \| x - z \|_2.
\]

In general, one need not use only the \( L^2 \)-norm in defining projections but is the most commonly used one. If \( C \) is a convex set, then the above problem reduces to a convex optimization problem. In several useful cases, one has access to a closed form solution for the projection.

For instance, if \( C = B_2(1) \) i.e., the unit \( L^2 \) ball, then projection is equivalent\(^3\) to a normalization step

\[
\Pi_{B_2(1)}(z) = \begin{cases} 
  z/\|z\|_2 & \text{if } \|z\| > 1 \\
  z & \text{otherwise}
\end{cases}
\]

For the case \( C = B_1(1) \), the projection step reduces to the popular soft thresholding operation. If \( \hat{z} := \Pi_{B_1(1)}(z) \), then \( \hat{z}_i = \max\{z_i - \theta, 0\} \), where \( \theta \) is a threshold that can be decided by a sorting operation on the vector [see Duchi et al., 2008, for details].

Projections onto convex sets have some very useful properties which come in handy while analyzing optimization algorithms. In the following, we will study three properties of projections. These are depicted visually in Figure 2.3 to help the reader gain an intuitive appeal.

**Lemma 2.2 (Projection Property-O).** For any set (convex or not) \( C \subset \mathbb{R}^p \) and \( z \in \mathbb{R}^p \), let \( \hat{z} := \Pi_C(z) \). Then for all \( x \in C \), \( \| \hat{z} - z \|_2 \leq \| x - z \|_2 \).

This property follows by simply observing that the projection step solves the optimization problem \( \min_{x \in C} \| x - z \|_2 \). Note that this property holds for all sets, whether convex or not. However, the following two properties necessarily hold only for convex sets.

**Lemma 2.3 (Projection Property-I).** For any convex set \( C \subset \mathbb{R}^p \) and any \( z \in \mathbb{R}^p \), let \( \hat{z} := \Pi_C(z) \). Then for all \( x \in C \), \( \langle x - \hat{z}, z - \hat{z} \rangle \leq 0 \).

\(^2\)See Exercise 2.2.

\(^3\)See Exercise 2.3.
2.2. CONVEX PROJECTIONS

![Diagram of projection operators and their properties.](image)

**Figure 2.3:** A depiction of projection operators and their properties. Projections reveal a closest point in the set being projected onto. For convex sets, projection property I ensures that the angle $\theta$ is always non-acute. Sets that satisfy projection property I also satisfy projection property II. Projection property II may be violated by non-convex sets. Projecting onto them may take the projected point $z$ closer to certain points in the set (for example, $\tilde{z}$) but farther from others (for example, $x$).

**Proof.** To prove this, assume the contra-positive. Suppose for some $x \in C$, we have $\langle x - \tilde{z}, z - \tilde{z} \rangle > 0$. Now, since $C$ is convex and $\tilde{z}, x \in C$, for any $\lambda \in [0, 1]$, we have $x_\lambda := \lambda \cdot x + (1 - \lambda) \cdot \tilde{z} \in C$. We will now show that for some value of $\lambda \in [0, 1]$, it must be the case that $\|z - x_\lambda\|_2 < \|z - \tilde{z}\|_2$. This will contradict the fact that $\tilde{z}$ is the closest point in the convex set to $z$ and prove the lemma. All that remains to be done is to find such a value of $\lambda$. The reader can verify that any value of $0 < \lambda < \min \left\{ 1, \frac{2\langle x - \tilde{z}, z - \tilde{z} \rangle}{\|x - \tilde{z}\|_2^2} \right\}$ suffices. Since we assumed $\langle x - \tilde{z}, z - \tilde{z} \rangle > 0$, any value of $\lambda$ chosen this way is always in $(0, 1]$.

Projection Property-I can be used to prove a very useful contraction property for convex projections. In some sense, a convex projection brings a point closer to all points in the convex set simultaneously.

**Lemma 2.4** (Projection Property-II). *For any convex set $C \subset \mathbb{R}^p$ and any $z \in \mathbb{R}^p$, let $\tilde{z} := \Pi_C(z)$. Then for all $x \in C$, $\|\tilde{z} - x\|_2 \leq \|z - x\|_2$.*

**Proof.** We have the following elementary inequalities

$$\|z - x\|_2^2 = \|(\tilde{z} - x) - (z - \tilde{z})\|_2^2$$

$$= \|\tilde{z} - x\|_2^2 + \|z - \tilde{z}\|_2^2 - 2 \langle \tilde{z} - x, z - \tilde{z} \rangle$$

$$\geq \|\tilde{z} - x\|_2^2 + \|z - \tilde{z}\|_2^2$$

(Projection Property-I)

Note that Projection Properties-I and II are also called first order properties and can be violated if the underlying set is non-convex. However, Projection Property-O, often called a zeroth order property, always holds, whether the underlying set is convex or not.
Algorithm 1 Projected Gradient Descent (PGD)

Input: Convex objective $f$, convex constraint set $C$, step lengths $\eta_t$
Output: A point $\tilde{x} \in C$ with near-optimal objective value

1: $x^1 \leftarrow 0$
2: for $t = 1, 2, \ldots, T$ do
3: \[ z^{t+1} = x^t - \eta_t \cdot \nabla f(x^t) \]
4: \[ x^{t+1} = \Pi_C(z^{t+1}) \]
5: end for
6: (OPTION 1) return $\tilde{x}_{\text{final}} = x^T$
7: (OPTION 2) return $\tilde{x}_{\text{avg}} = (\sum_{t=1}^T x^t) / T$
8: (OPTION 3) return $\tilde{x}_{\text{best}} = \arg\min_{t \in [T]} f(x^t)$

2.3 Projected Gradient Descent

We now move on to study the projected gradient descent algorithm. This is an extremely simple and efficient technique that can effortlessly scale to large problems. Although we will apply this technique to non-convex optimization tasks later, we first look at its behavior on convex optimization problems as a warm up exercise. We warn the reader that the proof techniques used in the convex case do not apply directly to non-convex problems. Consider the following optimization problem:

\[
\min_{x \in \mathbb{R}^p} f(x) \quad \text{s.t. } x \in C. \tag{CVX-OPT}
\]

In the above optimization problem, $C \subset \mathbb{R}^p$ is a convex constraint set and $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is a convex objective function. We will assume that we have oracle access to the gradient and projection operators, i.e., for any point $x \in \mathbb{R}^p$ we are able to access $\nabla f(x)$ and $\Pi_C(x)$.

The projected gradient descent algorithm is stated in Algorithm 1. The procedure generates iterates $x^t$ by taking steps guided by the gradient in an effort to reduce the function value locally. Finally it returns either the final iterate, the average iterate, or the best iterate.

2.4 Convergence Guarantees for PGD

We will analyze PGD for objective functions that are either a) convex with bounded gradients, or b) strongly convex and strongly smooth. Let $f^* = \min_{x \in C} f(x)$ be the optimal value of the optimization problem. A point $\tilde{x} \in C$ will be said to be an $\epsilon$-optimal solution if $f(\tilde{x}) \leq f^* + \epsilon$.

2.4.1 Convergence with Bounded Gradient Convex Functions

Consider a convex objective function $f$ with bounded gradients over a convex constraint set $C$ i.e., $\|f(x)\|_2 \leq G$ for all $x \in C$.

**Theorem 2.5.** Let $f$ be a convex objective with bounded gradients and Algorithm 1 be executed for $T$ time steps with step lengths $\eta_t = \eta = \frac{1}{\sqrt{T}}$. Then, for any $\epsilon > 0$, if $T = O\left(\frac{1}{\epsilon^2}\right)$, then

\[
\frac{1}{T} \sum_{t=1}^T f(x^t) \leq f^* + \epsilon.
\]

We see that the PGD algorithm in this setting ensures that the function value of the iterates approaches $f^*$ on an average. We can use this result to prove the convergence of the PGD
2.4. CONVERGENCE GUARANTEES FOR PGD

algorithm. If we use OPTION 3, i.e., \( \hat{x}_{\text{best}} \), then since by construction, we have \( f(\hat{x}_{\text{best}}) \leq f(x^t) \) for all \( t \), by applying Theorem 2.5, we get

\[
f(\hat{x}_{\text{best}}) \leq \frac{1}{T} \sum_{t=1}^{T} f(x^t) \leq f^* + \epsilon,
\]

If we use OPTION 2, i.e., \( \hat{x}_{\text{avg}} \), which is cheaper since we do not have to perform function evaluations to find the best iterate, we can apply Jensen’s inequality (Lemma 2.1) to get the following

\[
f(\hat{x}_{\text{avg}}) = f\left( \frac{1}{T} \sum_{t=1}^{T} x^t \right) \leq \frac{1}{T} \sum_{t=1}^{T} f(x^t) \leq f^* + \epsilon.
\]

Note that the Jensen’s inequality may be applied only when the function \( f \) is convex. Now, whereas OPTION 1 i.e., \( \hat{x}_{\text{final}} \), is the cheapest and does not require any additional operations, \( \hat{x}_{\text{final}} \) does not converge to the optimum for convex functions in general and may oscillate close to the optimum. However, we shall shortly see that \( \hat{x}_{\text{final}} \) does converge if the objective function is strongly smooth. Recall that strongly smooth functions may not grow at a faster-than-quadratic rate.

The reader would note that we have set the step length to a value that depends on the total number of iterations \( T \) for which the PGD algorithm is executed. This is called a horizon-aware setting of the step length. In case we are not sure what the value of \( T \) would be, a horizon-oblivious setting of \( \eta_t = \frac{1}{\sqrt{t}} \) can also be shown to work\(^4\).

**Proof (of Theorem 2.5).** Let \( x^* \in \arg\min_{x \in \mathcal{C}} f(x) \) denote any point in the constraint set where the optimum function value is achieved. Such a point always exists if the constraint set is closed and the objective function continuous. We will use the following potential function \( \Phi_t = f(x^t) - f(x^*) \) to track the progress of the algorithm. Note that \( \Phi_t \) measures the sub-optimality of the \( t \)-th iterate. Indeed, the statement of the theorem is equivalent to claiming that \( \frac{1}{T} \sum_{t=1}^{T} \Phi_t \leq \epsilon \).

(Apply Convexity) We apply convexity to upper bound the potential function at every step. Convexity is a global property and very useful in getting an upper bound on the level of sub-optimality of the current iterate in such analyses.

\[
\Phi_t = f(x^t) - f(x^*) \leq \langle \nabla f(x^t), x^t - x^* \rangle
\]

We now do some elementary manipulations

\[
\langle \nabla f(x^t), x^t - x^* \rangle = \frac{1}{\eta} \langle \eta \cdot \nabla f(x^t), x^t - x^* \rangle = \frac{1}{2\eta} \left( \| x^t - x^* \|^2 + \| \eta \cdot \nabla f(x^t) \|^2 - \| x^t - \eta \cdot \nabla f(x^t) - x^* \|^2 \right) \\
= \frac{1}{2\eta} \left( \| x^t - x^* \|^2 + \| \eta \cdot \nabla f(x^t) \|^2 - \| z^{t+1} - x^* \|^2 \right) \\
\leq \frac{1}{2\eta} \left( \| x^t - x^* \|^2 + \eta^2 G^2 - \| z^{t+1} - x^* \|^2 \right),
\]

where the first step applies the identity \( 2ab = a^2 + b^2 - (a+b)^2 \), the second step uses the update step of the PGD algorithm that sets \( z^{t+1} \leftarrow x^t - \eta_t \cdot \nabla f(x^t) \), and the third step uses the fact that the objective function \( f \) has bounded gradients.

\(^4\)See Exercise 2.4.
We apply Lemma 2.4 to get
\[ \|z^{t+1} - x^*\|_2^2 \geq \|x^{t+1} - x^*\|_2^2 \]
Putting all these together gives us
\[ \Phi_t \leq \frac{1}{2\eta} \left( \|x^t - x^*\|_2^2 - \|x^{t+1} - x^*\|_2^2 \right) + \frac{\eta G^2}{2} \]
The above expression is interesting since it tells us that, apart from the \( \eta G^2 / 2 \) term which is small as \( \eta = \frac{1}{\sqrt{T}} \), the current sub-optimality \( \Phi_t \) is small if the consecutive iterates \( x^t \) and \( x^{t+1} \) are close to each other (and hence similar in distance from \( x^* \)).

This observation is quite useful since it tells us that once PGD stops making a lot of progress, it actually converges to the optimum! In hindsight, this is to be expected. Since we are using a constant step length, only a vanishing gradient can cause PGD to stop progressing. However, for convex functions, this only happens at global optima. Summing the expression up across time steps, performing telescopic cancellations, using \( x^1 = 0 \), and dividing throughout by \( T \) gives us
\[
\frac{1}{T} \sum_{t=1}^{T} \Phi_t \leq \frac{1}{2\eta T} \left( \|x^*\|_2^2 - \|x^{T+1} - x^*\|_2^2 \right) + \frac{\eta G^2}{2}
\]
where in the second step, we have used the fact that \( \|x^{t+1} - x^*\|_2 \geq 0 \) and \( \eta = 1/\sqrt{T} \). This gives us the claimed result.

### 2.4.2 Convergence with Strongly Convex and Smooth Functions

We will now prove a stronger guarantee for PGD when the objective function is strongly convex and strongly smooth (see Definition 2.4).

**Theorem 2.6.** Let \( f \) be an objective that satisfies the \( \alpha \)-SC and \( \beta \)-SS properties. Let Algorithm 1 be executed with step lengths \( \eta_t = \eta = \frac{1}{\beta} \). Then after at most \( T = O \left( \frac{\beta}{\alpha} \log \frac{\beta}{\epsilon} \right) \) steps, we have \( f(x^T) \leq f(x^*) + \epsilon \).

This result is particularly nice since it ensures that the final iterate \( x_{\text{final}} = x^T \) converges, allowing us to use OPTION 1 in Algorithm 1 when the objective is SC/SS. A further advantage is the accelerated rate of convergence. Whereas for general convex functions, PGD requires \( O \left( \frac{1}{\epsilon} \right) \) iterations to reach an \( \epsilon \)-optimal solution, for SC/SS functions, it requires only \( O \left( \log \frac{1}{\epsilon} \right) \) iterations.

The reader would notice the insistence on the step length being set to \( \eta = \frac{1}{\beta} \). In fact the proof we show below crucially uses this setting. In practice, for many problems, \( \beta \) may not be known to us or may be expensive to compute which presents a problem. However, as it turns out, it is not necessary to set the step length exactly to \( 1/\beta \). The result can be shown to hold even for values of \( \eta < 1/\beta \), which are nevertheless large enough, but the proof becomes more involved. In practice, the step length is tuned globally by doing a grid search over several \( \eta \) values, or per-iteration using line search mechanisms, to obtain a step length value that assures good convergence rates.
2.4. CONVERGENCE GUARANTEES FOR PGD

Proof (of Theorem 2.6). This proof is a nice opportunity for the reader to see how the SC/SS properties are utilized in a convergence analysis. As with convexity in the proof of Theorem 2.5, the strong convexity property is a global property that will be useful in assessing the progress made so far by relating the optimal point \( x^* \) with the current iterate \( x^t \). Strong smoothness on the other hand, will be used locally to show that the procedure makes significant progress between iterates.

We will prove the result by showing that after at most \( T = O \left( \frac{n}{\epsilon} \log \frac{1}{\epsilon} \right) \) steps, we will have \( \left\| x^T - x^* \right\|_2^2 \leq \frac{2\epsilon}{\beta} \). This already tells us that we have reached very close to the optimum. However, we can use this to show that \( x^T \) is \( \epsilon \)-optimal in function value as well. Since we are very close to the optimum, it makes sense to apply strong smoothness to upper bound the sub-optimality as follows

\[
 f(x^T) \leq f(x^*) + \left\langle \nabla f(x^*), x^T - x^* \right\rangle + \frac{\beta}{2} \left\| x^T - x^* \right\|_2^2.
\]

Now, since \( x^* \) is an optimal point for the constrained optimization problem with a convex constraint set \( C \), the first order optimality condition [see Bubeck, 2015, Proposition 1.3] gives us \( \left\langle \nabla f(x^*), x - x^* \right\rangle \leq 0 \) for any \( x \in C \). Applying this condition with \( x = x^T \) gives us

\[
 f(x^T) - f(x^*) \leq \frac{\beta}{2} \left\| x^T - x^* \right\|_2^2 \leq \epsilon,
\]

which proves that \( x^T \) is an \( \epsilon \)-optimal point. We now show \( \left\| x^T - x^* \right\|_2^2 \leq \frac{2\epsilon}{\beta} \). Given that we wish to show convergence in terms of the iterates, and not in terms of the function values, as we did in Theorem 2.5, a natural potential function for this analysis is \( \Phi_t = \left\| x^t - x^* \right\|_2^2 \).

(Apply Strong Smoothness) As discussed before, we use it to show that PGD always makes significant progress in each iteration.

\[
 f(x^{t+1}) - f(x^t) \leq \left\langle \nabla f(x^t), x^{t+1} - x^t \right\rangle + \frac{\beta}{2} \left\| x^t - x^{t+1} \right\|_2^2
 = \left\langle \nabla f(x^t), x^{t+1} - x^* \right\rangle + \left\langle \nabla f(x^t), x^* - x^t \right\rangle + \frac{\beta}{2} \left\| x^t - x^{t+1} \right\|_2^2
 = \frac{1}{\eta} \left\langle x^t - z^{t+1}, x^{t+1} - x^* \right\rangle + \left\langle \nabla f(x^t), x^* - x^t \right\rangle + \frac{\beta}{2} \left\| x^t - x^{t+1} \right\|_2^2
\]

(Apply Projection Rule) The above expression contains an unwieldy term \( z^{t+1} \). Since this term only appears during projection steps, we eliminate it by applying Projection Property-I (Lemma 2.3) to get

\[
 \left\langle x^t - z^{t+1}, x^{t+1} - x^* \right\rangle \leq \left\langle x^t - z^{t+1}, x^{t+1} - x^* \right\rangle
 = \frac{\left\| x^t - x^* \right\|_2^2 - \left\| x^t - x^{t+1} \right\|_2^2 - \left\| x^{t+1} - x^* \right\|_2^2}{2}
\]

Using \( \eta = 1/\beta \) and combining the above results gives us

\[
 f(x^{t+1}) - f(x^t) \leq \left\langle \nabla f(x^t), x^* - x^t \right\rangle + \frac{\beta}{2} \left( \left\| x^t - x^* \right\|_2^2 - \left\| x^{t+1} - x^* \right\|_2^2 \right)
\]

(Apply Strong Convexity) The above expression is perfect for a telescoping step but for the inner product term. Fortunately, this can be eliminated using strong convexity.

\[
 \left\langle \nabla f(x^t), x^* - x^t \right\rangle \leq f(x^*) - f(x^t) - \frac{\alpha}{2} \left\| x^t - x^* \right\|_2^2
\]

23
Combining with the above this gives us
\[
f(x^{t+1}) - f(x^*) \leq \frac{\beta - \alpha}{2} \|x^t - x^*\|^2 - \frac{\beta}{2} \|x^{t+1} - x^*\|^2.
\]

The above form seems almost ready for a telescoping exercise. However, something much stronger can be said here, especially due to the \(\frac{\alpha}{2} \|x^t - x^*\|^2\) term. Notice that we have

\[
f(x^{t+1}) \geq f(x^*).
\]

This means

\[
\frac{\beta}{2} \|x^{t+1} - x^*\|^2 \leq \frac{\beta - \alpha}{2} \|x^t - x^*\|^2,
\]

which can be written as

\[
\Phi_{t+1} \leq \left(1 - \frac{\alpha}{\beta}\right) \Phi_t \leq \exp\left(-\frac{\alpha t}{\beta}\right) \Phi_t,
\]

where we have used the fact that \(1 - x \leq \exp(-x)\) for all \(x \in \mathbb{R}\). What we have arrived at is a very powerful result as it assures us that the potential value goes down by a constant fraction at every iteration! Applying this result recursively gives us

\[
\Phi_{t+1} \leq \exp\left(-\frac{\alpha t}{\beta}\right) \Phi_1 = \exp\left(-\frac{\alpha t}{\beta}\right) \|x^*\|^2,
\]

since \(x^1 = 0\). Thus, we deduce that \(\Phi_T = \|x^T - x^*\|^2 \leq \frac{2\beta}{\alpha} \log \frac{3}{\beta}\) steps which finishes the proof.

We notice that the convergence of the PGD algorithm is of the form \(\|x^{t+1} - x^*\|^2 \leq \exp\left(-\frac{\alpha t}{\beta}\right) \|x^t\|^2\). The number \(\kappa := \frac{\beta}{\alpha}\) is the condition number of the optimization problem. The concept of condition number is central to numerical optimization. Below we give an informal and generic definition for the concept. In later sections we will see the condition number appearing repeatedly in the context of the convergence of various optimization algorithms for convex, as well as non-convex problems. The exact numerical form of the condition number (for instance here it is \(\beta/\alpha\)) will also change depending on the application at hand. However, in general, all these definitions of condition number will satisfy the following property.

**Definition 2.6 (Condition Number - Informal).** The condition number of a function \(f : \mathcal{X} \rightarrow \mathbb{R}\) is a scalar \(\kappa \in \mathbb{R}\) that bounds how much the function value can change relative to a perturbation of the input.

Functions with a small condition number are stable and changes to their input do not affect the function output values too much. However, functions with a large condition number can be quite jumpy and experience abrupt changes in output values even if the input is changed slightly. To gain a deeper appreciation of this concept, consider a differentiable function \(f\) that is also \(\alpha\)-SC and \(\beta\)-SS. Consider a stationary point for \(f\) i.e., a point \(x\) such that \(\nabla f(x) = 0\).

For a general function, such a point can be a local optima or a saddle point. However, since \(f\) is strongly convex, \(x\) is the (unique) global minima\(^5\) of \(f\). Then we have, for any other point \(y\)

\[
\frac{\alpha}{2} \|x - y\|^2 \leq f(y) - f(x) \leq \frac{\beta}{2} \|x - y\|^2.
\]

Dividing throughout by \(\frac{\alpha}{2} \|x - y\|^2\) gives us

\[
\frac{f(y) - f(x)}{\frac{\alpha}{2} \|x - y\|^2} \in \left[1, \frac{\beta}{\alpha}\right] := [1, \kappa].
\]

---

\(^5\)See Exercise 2.5.
2.5. EXERCISES
Thus, upon perturbing the input from the global minimum \( \mathbf{x} \) to a point \( \| \mathbf{x} - \mathbf{y} \|_2 =: \epsilon \) distance away, the function value does change much – it goes up by an amount at least \( \frac{\alpha \epsilon^2}{2} \) but at most \( \kappa \cdot \frac{\alpha \epsilon^2}{2} \). Such well behaved response to perturbations is very easy for optimization algorithms to exploit to give fast convergence.

The condition number of the objective function can significantly affect the convergence rate of algorithms. Indeed, if \( \kappa = \frac{\beta}{\alpha} \) is small, then \( \exp \left( -\frac{\alpha}{\beta} \right) = \exp \left( -\frac{1}{\kappa} \right) \) would be small, ensuring fast convergence. However, if \( \kappa \gg 1 \) then \( \exp \left( -\frac{1}{\kappa} \right) \approx 1 \) and the procedure might offer slow convergence.

2.5 Exercises

Exercise 2.1. Show that strong smoothness does not imply convexity by constructing a non-convex function \( f : \mathbb{R}^p \to \mathbb{R} \) that is 1-SS.

Exercise 2.2. Show that if a differentiable function \( f \) has bounded gradients i.e., \( \| \nabla f(\mathbf{x}) \|_2 \leq G \) for all \( \mathbf{x} \in \mathbb{R}^d \), then \( f \) is Lipschitz. What is its Lipschitz constant?
Hint: use the mean value theorem.

Exercise 2.3. Show that for any point \( \mathbf{z} \notin \mathcal{B}_2(r) \), the projection onto the ball is given by \( \Pi_{\mathcal{B}_2(r)}(\mathbf{z}) = \frac{r}{\| \mathbf{z} \|_2} \cdot \mathbf{z} \).

Exercise 2.4. Show that a horizon-oblivious setting of \( \eta_t = \frac{1}{\sqrt{t}} \) while executing the PGD algorithm with a convex function with bounded gradients also ensures convergence.
Hint: the convergence rates may be a bit different for this setting.

Exercise 2.5. Show that if \( f : \mathbb{R}^p \to \mathbb{R} \) is a strongly convex function that is differentiable, then there is a unique point \( \mathbf{x}^* \in \mathbb{R}^p \) that minimizes the function value \( f \) i.e., \( f(\mathbf{x}^*) = \min_{\mathbf{x} \in \mathbb{R}^p} f(\mathbf{x}) \).

Exercise 2.6. Show that the set of sparse vectors \( \mathcal{B}_0(s) \subset \mathbb{R}^p \) is non-convex for any \( s < p \). What happens when \( s = p \)?

Exercise 2.7. Show that \( \mathcal{B}_{\text{rank}}(r) \subseteq \mathbb{R}^{n \times n} \), the set of \( n \times n \) matrices with rank at most \( r \), is non-convex for any \( r < n \). What happens when \( r = n \)?

Exercise 2.8. Consider the Cartesian product set \( \mathcal{C} = \mathbb{R}^{m \times r} \times \mathbb{R}^{n \times r} \). Show that it is convex.

Exercise 2.9. Consider a least squares optimization problem with a strongly convex and smooth objective. Show that the condition number of this problem is equal to the condition number of the Hessian matrix of the objective function.

Exercise 2.10. Show that if \( f : \mathbb{R}^p \to \mathbb{R} \) is a strongly convex function that is differentiable, then optimization problems with \( f \) as an objective and a convex constraint set \( \mathcal{C} \) always have a unique solution i.e., there is a unique point \( \mathbf{x}^* \in \mathcal{C} \) that is a solution to the optimization problem \( \arg \min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x}) \).
This generalizes the result in Exercise 2.5.
Hint: use the first order optimality condition (see proof of Theorem 2.6)