Chapter 3

We have introduced gradient descent and studied its performance under Lipschitz gradient continuity. This lecture introduces the basic notions of convexity in optimization. We will discuss convex functions, convex constraints, and whether gradient descent is benefited by convexity. Apart from standard convexity, we will also introduce the notion of *strong convexity* and discuss its effect in practice and theory.
This chapter continues to evolve around convergence rates and contains some discussion about lower bounds on such rates.

Convexity | Gradient Descent | Strong convexity | Other global assumptions | Projection onto convex sets

(*The discussion in this chapter will primarily focus on the unconstrained case:* $\min_x f(x)$ *unless otherwise stated. Also, the constants found through proofs might not be the tightest possible, but the related analysis conveys the same message as the tightest ones.*)

Optimization via only linear algebra concepts. Before we delve into the main topic of this chapter, which is convexity, let us first focus on a simple –but significant– problem instance: that of *unconstrained quadratic form minimization.* Quadratic forms appear in the literature *i*) either because the problem is described as such: e.g., the principal component analysis problem involves the minimization/maximization of a quadratic form; *ii*) or because one utilizes local quadratic form approximations of a (potentially) complicated function, through Taylor expansions. In any case, quadratic forms are simple and complicated enough to provide insights for many of our decisions in general optimization theory.

In math terms, let us define the following function:

$$f(x) = \tfrac{1}{2} x^\top Q x - b^\top x + r,$$

where $x \in \mathbb{R}^p$, $Q \in \mathbb{R}^{p\times p}$ is a symmetric matrix, $b \in \mathbb{R}^p$ is a vector and r is a scalar. We are interested in the following problem:

$$\min_{x\in\mathbb{R}^p} f(x).$$

Following the discussion in the previous chapters, we will use gradients as a tool for our solution. Let us first compute the gradient of $f(\cdot)$ at a point x:

$$\nabla f(x) = Qx - b \in \mathbb{R}^p.$$

One could argue that, based on gradient descent theory, we are looking for points where $\nabla f(x) = 0$. I.e., we are looking for points where:

$$Qx = b.$$

Thus far, we have not made any assumptions about the matrix Q other than symmetry. Let us first assume that Q is full-rank; in that case, the system of linear equations $Qx = b$ has a unique solution that satisfies this expression. This further implies a unique point in $f(x)$ such that the gradient of $f(x)$ is zero.

This is great thus far! We have found that, given full-rank Q, we can find a point of interest if we solve the linear system of equations $Qx = b$. Yet, what is this point? Is it a minimum? Is it a maximum? Is it a saddle point? To continue our discussion (and to connect with convexity), we will further assume that Q is a symmetric matrix with real eigenvalues $L := \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p := \mu > 0$ (without loss of generality, in descending order). In other words, we say that the eigenvalues of Q "live" in the interval $[\mu, L]$, for some constants $\mu < L \in \mathbb{R}$. The fact that the eigenvalues are all non-zero originates from the fact that Q is full-rank; the additional assumption is that *all eigenvalues are positive*.

One way to find a solution for $Qx = b$ or $\nabla f(x) = 0$ (other than computing and applying the inverse Q^{-1}) is gradient descent. In particular, gradient descent iteratively points in directions that potentially lead to regions with small gradients. Based on the previous chapter, we have:

$$x_{t+1} = x_t - \eta \nabla f(x_t) = x_t - \eta \left(Q x_t - b\right)$$

Let $x^\star$ be the stationary point such that $Qx^\star = b$. Then, the above recursion becomes:

$$x_{t+1} = x_t - \eta \left(Q x_t - b\right) = x_t - \eta \left(Q x_t - Q x^\star\right).$$

By adding $x^\star$ on both sides of this expression, we get the following recursion:

$$x_{t+1} - x^\star = (I - \eta Q) \cdot (x_t - x^\star).$$

Here, it is important to remember the dimensions of the quantities involved: $I - \eta Q \in \mathbb{R}^{p\times p}$, $x_t - x^\star \in \mathbb{R}^p$ and $x_{t+1} - x^\star \in \mathbb{R}^p$.[5] Taking norms on both sides and applying the Cauchy-Schwarz inequality, we obtain:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2 &= \|(I - \eta Q) \cdot (x_t - x^\star)\|_2 \\ &\leq \|I - \eta Q\|_2 \cdot \|x_t - x^\star\|_2 ;\end{aligned}$$

i.e., the distance of the estimate x_{t+1} from $x^\star$ is bounded by the previous distance $\|x_t - x^\star\|_2$, multiplied by the quantity in colored text: $\|I - \eta Q\|_2$. In words, if $\|I - \eta Q\|_2 > 1$, then the above recursion is not very useful!: it just guarantees that the next iteration's distance is bounded above by a *larger* distance than the previous iteration; in that scenario, the algorithm could diverge and still satisfy this recursion!

Thus, it is reasonable to *demand* that $\|I - \eta Q\|_2 < 1$, which further translates into bounding the eigenvalues of the matrix $I - \eta Q$ to be less than 1. However, we know that the eigenvalues satisfy $\lambda_i \in [\mu, L]$ per our assumption. The only way we can control the spectrum of the matrix $I - \eta Q$ is by carefully selecting the step size η:

- Due to the facts that Q is symmetric, and thus diagonalizable in an orthonormal basis, the worst case analysis dictates that the largest value of the quantity $\|I - \eta Q\|_2$, for a fixed η, is that of:

$$\max_{\lambda\in[\mu,L]} |1 - \eta\lambda|.$$

- The control we can impose is by *minimizing* that quantity by carefully selecting η:

$$\min_\eta \left(\max_{\lambda\in[\mu,L]} |1 - \eta\lambda| \right).$$

- Simple geometric arguments on intersecting lines suggests that the "optimal" step size is that of $\eta = \frac{2}{L+\mu} = \frac{2}{\lambda_1+\lambda_p}$. In that case, we have:

$$\begin{aligned}\max_{\lambda\in[\mu,L]} |1 - \eta\lambda| &= \max_{\lambda\in[\mu,L]} |1 - \tfrac{2\lambda}{L+\mu}| \\ &= |1 - \tfrac{2\mu}{L+\mu}| \\ &= 1 - \tfrac{2}{\kappa+1} \in (0, 1);\end{aligned}$$

[5] This is a useful check in all your computations involving multi-variate algebra: make sure that the quantities involved add-up/multiply in a way that comply with the linear algebra rules.

here, $\kappa = \frac{\lambda_1}{\lambda_p} > 1$ is a special quantity that will play a significant role in the discussions below. We will get back to this quantity soon. The above recursion becomes:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2 &\leq \|I - \eta Q\|_2 \cdot \|x_t - x^\star\|_2 \\ &\leq \left(1 - \tfrac{2}{\kappa+1}\right) \cdot \|x_t - x^\star\|_2 \\ &\leq \cdots \leq \left(1 - \tfrac{2}{\kappa+1}\right)^{t+1} \cdot \|x_0 - x^\star\|_2 ;\end{aligned}$$

i.e., as long as the quantity $\left(1 - \frac{2}{\kappa+1}\right) < 1$, the exponentiated term $\left(1 - \frac{2}{\kappa+1}\right)^{t+1}$ converges exponentially fast to zero, leading to convergence of the quantity $\|x_{t+1} - x^\star\|_2 \overset{\text{goes fast to}}{\longrightarrow} 0$.

- The above analysis is "optimal" but requires the knowledge of both λ_1 and λ_p of Q; we will later see that this is a stronger requirement in practice for more complicated functions. Here, a classical choice for step size is that of $\eta = \frac{1}{L} = \frac{1}{\lambda_1}$, which leads to (with a similar analysis to the above):

$$\max_{\lambda \in [\mu, L]} |1 - \eta\lambda| = \max_{\lambda \in [\mu, L]} |1 - \tfrac{\lambda}{L}| = 1 - \tfrac{1}{\kappa} \in (0, 1).$$

Then, a similar analysis to the above leads to the following:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2 &\leq \|I - \eta Q\|_2 \cdot \|x_t - x^\star\|_2 \\ &\leq \left(1 - \tfrac{1}{\kappa}\right) \cdot \|x_t - x^\star\|_2 \\ &\leq \cdots \leq \left(1 - \tfrac{1}{\kappa}\right)^{t+1} \cdot \|x_0 - x^\star\|_2 .\end{aligned}$$

What have we proved thus far? First, we saw that, for some simple objectives, simple arguments originating from linear algebra lead to exciting connections with optimization theory: *i*) solutions to systems of linear equations are equivalent to finding stationary points of functions; *ii*) the spectrum of matrices related to the function at hand –like Q– characterize the type of the stationary point we look/aim for; *iii*) ratios of eigenvalues from such matrices define the convergence rate of an algorithm; *iv*) finally, simple rules can be extracted from the above on how one selects the step size.

So, overall, quadratic form minimization (under the assumption of symmetric Q with positive eigenvalues) can be efficiently solved to *global optimality* $x^\star$ using gradient descent. I.e., gradient descent can estimate a point close to $x^\star$ in a small number of iterations. But this is just a particular instance; *can we infer some properties that could lead to more general analysis and include more complicated functions f?*.

What stands out from the assumptions above is that Q has positive eigenvalues, but what is Q for this $f(\cdot)$? Given that $f(x) = \frac{1}{2} x^\top Q x - b^\top x + r$, simple calculus for Hessian calculation leads to:

$$\nabla^2 f(x) = Q.$$

I.e., the Hessian of quadratic forms are the matrices Q themselves! Thus, assuming properties on Q is equivalent to making assumptions on the Hessian of the function f. Specifically, positive eigenvalues for Q mean that the Hessian of f is positive definite, which further implies that the landscape of the function looks like ... a bowl! (*refer to plots of toy functions and Hessians in the previous chapter*). Or, in math terms, the landscape of the function is *convex*.

Convexity. A key consequence of convexity is that any local solution is global in convex optimization. To understand convexity in functions, we will cover some definitions first.

Definition 17. (Convex Function) *$f : \mathbb{R} \to \mathbb{R}$ is a univariate convex function if, for $\forall \alpha \in [0, 1]$, the following holds:*

$$f(\alpha x + (1 - \alpha) y) \leq \alpha f(x) + (1 - \alpha) f(y), \quad \forall x, y.$$

This dictates that the function value of f at any point in a given interval $[x, y]$ is lower than any secant connecting two points within that interval; see Figure 19.

Alternatively, a convex function can be defined as one that lies above any (hyper)plane tangential to f at any point. Using the gradient $\nabla f(x)$, this is interpreted as:

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle.$$

and depicted in low dimensions, as in Figure 20.

But what if f is not uniquely differentiable? I.e., what if f does not have a unique gradient at all points, but there are points where we can compute a set of gradients, *the subgradients $\partial f(x)$*? The same ideas apply in that case; see Figure 21 for the case of $f(x) = |x|$. In this example, f has a set of subgradients $\partial f(x)$ at point $x = 0$. These subgradients could take any value in the interval $[-1, 1]$; i.e., the set of all lines that touch $(0, f(0))$ with slope between -1 and 1. In general, for convex f and any subgradient $g \in \partial f(y)$ we have:

$$f(x) \geq f(y) + \langle g, x - y \rangle.$$

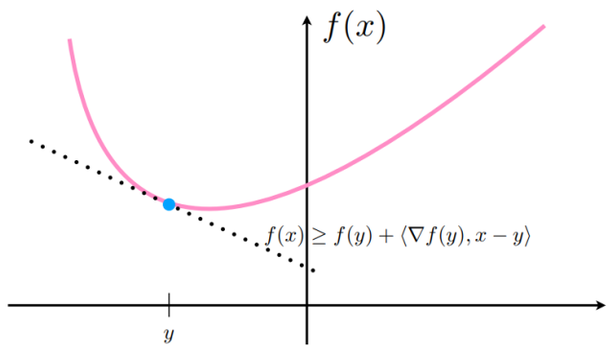

Fig. 20. Convex interpretation via gradients.

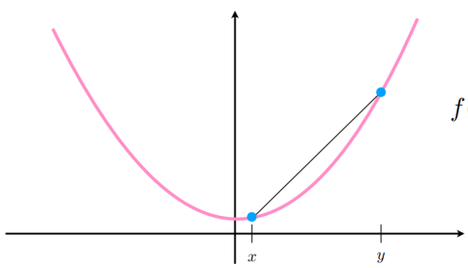

Fig. 19. $f(x) = x^2 + \beta$, for $\beta > 0$.

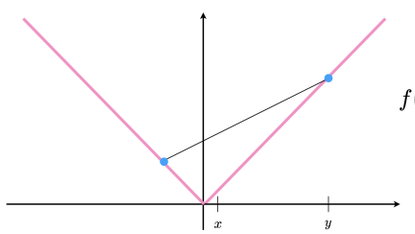

Fig. 21. $f(x) = |x|$.

The opposite (substitute $\leq$ with $\geq$) is a concave function.

An advantageous inequality for convex functions is Jensen's inequality:

Lemma 2. *For a convex function f, Jensen's inequality states:*

$$f\left(\mathbb{E}\left[x\right]\right) \leq \mathbb{E}\left[f(x)\right].$$

The geometric interpretation of Jensen's inequality that relates to convex functions is that *the function value of the average of two points is less than the average of the function values of the two points*, i.e.,

$$f\left(\tfrac{x_1+x_2}{2}\right) \leq \tfrac{1}{2}f(x_1) + \tfrac{1}{2}f(x_2).$$

What are some good examples of convex functions that we observe in practice? Examples are shown in Table 1.

Table 1. Convexity of common functions

Function	Example	Attributes
ℓ_p norms $p \geq 1$	$\|x\|_2$, $\|x\|_1$, $\|x\|_\infty$	convex
ℓ_p matrix norms $p \geq 1$	$\|X\|_2$	convex
Square root function	$\sqrt{x}$	concave
Maximum	$\max x_1, \cdots, x_n$	convex
Minimum	$\min x_1, \cdots, x_n$	concave
Sum of convex functions		convex
Logarithmic functions	log(det(X))	convex if $X \succeq 0$
Affine/linear functions	$\sum_{i=1}^{N} X_i i$	convex and concave
Eigenvalue functions	$\lambda_{\max}(X)$	convex if $X = X^\top$

Properties of convex functions. There are several alternative and potentially more practical definitions of a convex function:

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle, \quad \forall x, y$$
$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq 0, \quad \forall x, y$$
$$\nabla^2 f(x) \succeq 0, \quad \forall x.$$

A key property of convex functions is the following lemma.

Lemma 3. *Any stationary point of a convex function f is a global minimum.*

Proof: Assume that $f : \mathbb{R}^p \to \mathbb{R}$. Let $\widehat{x}$ denote a stationary point of f, where $\nabla f(\widehat{x}) = 0$. Since f is a convex function, we know that:

$$f(x) \geq f(\widehat{x}) + \langle \nabla f(\widehat{x}), x - \widehat{x} \rangle = f(\widehat{x}), \quad \forall x,$$

where the last equality is due to $\nabla f(\widehat{x}) = 0$. However, the above holds for all x, and thus for $x^\star$, which is/are the global minimum/minima. Thus, we have:

$$f(x^\star) \geq f(\widehat{x}), \quad \forall x^\star,$$

which is a contradiction. This implies that all stationary points $\widehat{x}$ are equivalent to the global minimum. □

While this fact provides hope for finding the global minimum, more is needed to guarantee the tractability and practicality of the proposed algorithms.

Does convexity help convergence rate?. We will study whether convexity improves the convergence rate of gradient descent. We remind the reader that, for a differentiable function f with gradient $\nabla f(\cdot)$, gradient descent satisfies:

$$x_{t+1} = x_t - \eta_t \nabla f(x_t), \quad t = 0, 1, \ldots$$

We will make the same baseline assumptions as before: we will assume that f has Lipschitz continuous gradients:

$$\|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L \cdot \|x_1 - x_2\|_2, \quad \forall x_1, x_2.$$

The only additional assumption we make is that f is also convex.

We will follow a different perspective—let $x^\star$ denote a global minimum.[6] Further, assume we use a constant step size $\eta_t = \eta$. Then, the following equality holds:

$$\begin{aligned}\|x_{t+1} - x^\star\|_2^2 &= \|x_t - \eta \nabla f(x_t) - x^\star\|_2^2 \\ &= \|x_t - x^\star\|_2^2 + \eta^2 \|\nabla f(x_t)\|_2^2 \\ &\quad - 2\eta \langle \nabla f(x_t),\ x_t - x^\star \rangle\end{aligned}$$

We can show that Lipschitz gradient continuity, along with convexity, leads to the following inequality:

$$\tfrac{1}{L} \cdot \|\nabla f(x_1) - \nabla f(x_2)\|_2^2 \leq \langle \nabla f(x_1) - \nabla f(x_2), x_1 - x_2 \rangle.$$

Substituting $x_1 \equiv x^\star$, $x_2 \equiv x_t$, and assuming $\nabla f(x^\star) = 0$ in the above inequality, we get:

$$\begin{aligned}\tfrac{1}{L} \cdot \|\nabla f(x_t)\|_2^2 \leq \langle -\nabla f(x_t), x^\star - x_t \rangle \quad &\Rightarrow \\ \langle \nabla f(x_t), x_t - x^\star \rangle \geq \tfrac{1}{L} \cdot \|\nabla f(x_t)\|_2^2 \quad &\Rightarrow \\ -2\eta \langle \nabla f(x_t), x_t - x^\star \rangle \leq -\tfrac{2\eta}{L} \cdot \|\nabla f(x_t)\|_2^2 &\end{aligned}$$

Combining with the above, we obtain:

$$\|x_{t+1} - x^\star\|_2^2 \leq \|x_t - x^\star\|_2^2 - \eta\left(\tfrac{2}{L} - \eta\right) \cdot \|\nabla f(x_t)\|_2^2.$$

Assuming $0 < \eta < \frac{2}{L}$, the second term on the right-hand side is negative. This implies that per iteration, we decrease the distance to optimum as in:

$$\|x_{t+1} - x^\star\|_2^2 \leq \|x_t - x^\star\|_2^2 \leq \cdots \leq \|x_0 - x^\star\|_2^2.$$

(*Question: Would such a statement hold for non-convex scenarios? Under which conditions?*)

By the analysis of the previous—not necessarily convex—case in the previous chapter, we also know that:

$$f(x_{t+1}) \leq f(x_t) - \eta\left(1 - \tfrac{L}{2}\eta\right) \cdot \|\nabla f(x_t)\|_2^2$$

By convexity, we also have:

$$\begin{aligned}f(x^\star) \geq f(x_t) + \langle \nabla f(x_t), x^\star - x_t \rangle \quad &\Rightarrow \\ f(x_t) - f(x^\star) \leq \langle \nabla f(x_t), x_t - x^\star \rangle & \\ \leq \|x_t - x^\star\|_2 \cdot \|\nabla f(x_t)\|_2 & \\ \leq \|x_0 - x^\star\|_2 \cdot \|\nabla f(x_t)\|_2. &\end{aligned}$$

The last inequality is based on the previous observation that $\|x_{t+1} - x^\star\|_2 \leq \|x_0 - x^\star\|_2$.

Then, we can combine the above into:

$$\begin{aligned}[f(x_{t+1}) - f(x^\star)] &\leq [f(x_t) - f(x^\star)] - \eta\left(1 - \tfrac{L}{2}\eta\right) \cdot \|\nabla f(x_t)\|_2^2 \\ &\leq [f(x_t) - f(x^\star)] - \eta\left(1 - \tfrac{L}{2}\eta\right) \cdot \tfrac{\left[f(x_t) - f(x^\star)\right]^2}{\|x_0 - x^\star\|_2^2}\end{aligned}$$

Define $\Delta_t := f(x_t) - f(x^\star)$. Then:

$$\Delta_{t+1} \leq \Delta_t - \tfrac{\eta\left(1 - \frac{L}{2}\eta\right)}{\|x_0 - x^\star\|_2^2} \cdot \Delta_t^2 = \Delta_t \cdot \left(1 - \tfrac{\eta\left(1 - \frac{L}{2}\eta\right)}{\|x_0 - x^\star\|_2^2} \cdot \Delta_t\right) \quad \Rightarrow$$

$$\tfrac{\Delta_{t+1}}{\Delta_t} \leq 1 - \tfrac{\eta\left(1 - \frac{L}{2}\eta\right)}{\|x_0 - x^\star\|_2^2} \cdot \Delta_t \Rightarrow$$

$$\tfrac{1}{\Delta_{t+1}} \geq \tfrac{1}{\Delta_t} + \tfrac{\eta\left(1 - \frac{L}{2}\eta\right)}{\|x_0 - x^\star\|_2^2} \cdot \tfrac{\Delta_t}{\Delta_{t+1}} \geq \tfrac{1}{\Delta_t} + \tfrac{\eta\left(1 - \frac{L}{2}\eta\right)}{\|x_0 - x^\star\|_2^2},$$

[6] Remember there might be multiple equivalent global minima, but we can assume we are converging to one of them.

for a step size $\eta = \frac{1}{L}$.

Unfolding the recursion for T iterations:

$$\frac{1}{\Delta_T} \geq \frac{1}{\Delta_0} + \frac{\eta\left(1-\frac{L}{2}\eta\right)}{\|x_0 - x^\star\|_2^2} \cdot T,$$

which leads to:

$$\begin{aligned} f(x_T) - f(x^\star) &\leq \frac{2L\left(f(x_0) - f(x^\star)\right) \cdot \|x_0 - x^\star\|_2^2}{2L\|x_0 - x^\star\|_2^2 + T \cdot \left(f(x_0) - f(x^\star)\right)} \\ &= O\left(\frac{1}{T}\right). \end{aligned}$$

The last expression is because all the other quantities are constant and depend on the initialization. Another way to interpret the above result is that if we require $f(x_T) - f(x^\star) \leq \varepsilon$, we have to perform $O\left(\frac{1}{\varepsilon}\right)$ number of iterations.

How does this compare to the result we already know? Remember that assuming only Lipschitz gradient continuity, we have:

$$\min_t \|\nabla f(x_t)\|_2 = O\left(\frac{1}{\sqrt{T}}\right),$$

and we need $O\left(1/\varepsilon^2\right)$ iterations to achieve $\min_t \|\nabla f(x_t)\|_2 \leq \varepsilon$. This reveals that convexity gains are two-fold: *i*) gradient descent over convex functions leads to convergence to global minimum/minima, not just stationary points; *ii*) gradient descent over convex functions effectively shows improved performance, compared to gradient descent over only L-smooth functions.

Beyond boilerplate convexity: strong convexity. Achieving better convergence rates can be achieved by assuming more than just convexity for f. *Strong convexity* is one such assumption that can lead to an improved result. In plain words, it implies that f should be steep enough so that gradient descent can progress (more aggressively). To see this, let us first provide its definition:

Definition 18. (Strong Convexity) *A function $f : \mathbb{R}^p \to \mathbb{R}$ is a strongly convex function if it is convex and, for $\mu > 0$, satisfies:*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|_2^2, \ \forall x, y$$

(*In the recent optimization literature, following a "machine learning" notation, L is usually substituted with β and μ with α. Here, we will follow the notation that Nesterov has used.*)

A visual illustration of strong convexity is provided in the next figure.

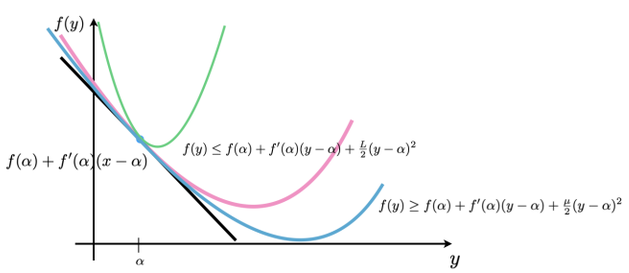

Fig. 22. Strong convexity interpretation and its relation to Lipschitz gradient continuity.

To interpret this further, while Lipschitz gradient continuity implies that, at any point of the domain of f, we can upper bound f with a quadratic (green curve), strong convexity implies that, at any point of the domain of f, we can lower bound f with a quadratic (blue curve).

A strongly convex function has a unique minimizer. Remember that a convex function has the nice property that every local minimum is a global minimum. Still, there is no guarantee that the set of global minima is a singleton.

There are several alternative and equivalent characterizations of strong convexity to know:

$$\begin{aligned} \langle \nabla f(x) - \nabla f(y), x - y \rangle &\geq \mu\|x - y\|_2^2, \\ f(y) &\leq f(x) + \langle \nabla f(x), y - x \rangle \\ &\quad + \frac{1}{2\mu}\|\nabla f(x) - \nabla f(y)\|_2^2, \\ \langle \nabla f(x) - \nabla f(y), x - y \rangle &\leq \frac{1}{\mu}\|\nabla f(x) - \nabla f(y)\|_2^2, \\ \nabla^2 f(x) &\succeq \mu \cdot I. \end{aligned}$$

(*The convergence rate proof of just a strongly convex function–not necessarily L-smooth–is left for exercise.*)

The L-smooth and μ-strongly convex functions. In convex optimization research, the two classes of convex functions that have attracted the most attention are the set of L-smooth functions (i.e., with Lipschitz continuous gradients) and the set of L-smooth AND μ-strongly convex functions.

To understand what strong convexity adds w.r.t. convergence rates, we jointly study the performance of gradient descent under these assumptions.

Similar to the proof of L-smooth functions:

$$\begin{aligned} \|x_{t+1} - x^\star\|_2^2 = \|x_t - x^\star\|_2^2 + \eta^2\|\nabla f(x_t)\|_2^2 \\ - 2\eta \langle \nabla f(x_t),\ x_t - x^\star \rangle \end{aligned}$$

A key property of L-smooth and μ-strongly convex functions is the following lemma:

Lemma 4. *Let f satisfy L-smoothness and μ-strongly convexity. Then:*

$$\begin{aligned} \langle \nabla f(x) - \nabla f(y),\ x - y \rangle \geq \frac{\mu L}{\mu + L}\|x - y\|_2^2 \\ + \frac{1}{\mu + L} \cdot \|\nabla f(x) - \nabla f(y)\|_2^2 \end{aligned}$$

(*We will see how this convex condition will "inspire" similar conditions for non-convex optimization.*)

We use the lemma above, with the substitution $x \equiv x^\star$, $y \equiv x_t$, and knowing that $\nabla f(x^\star) = 0$. This leads to:

$$-\langle \nabla f(x_t),\ x^\star - x_t \rangle \geq \frac{\mu L}{\mu + L}\|x_t - x^\star\|_2^2 + \frac{1}{\mu + L} \cdot \|\nabla f(x_t)\|_2^2.$$

Using this in the inequality above, we obtain:

$$\begin{aligned} \|x_{t+1} - x^\star\|_2^2 &\leq \|x_t - x^\star\|_2^2 + \eta^2\|\nabla f(x_t)\|_2^2 \\ &\quad - \frac{2\eta\mu L}{\mu + L}\|x_t - x^\star\|_2^2 - \frac{2\eta}{\mu + L} \cdot \|\nabla f(x_t)\|_2^2 \\ &= \left(1 - \frac{2\eta\mu L}{\mu + L}\right) \cdot \|x_t - x^\star\|_2^2 \\ &\quad + \eta \cdot \left(\eta - \frac{2}{\mu + L}\right) \cdot \|\nabla f(x_t)\|_2^2 \end{aligned}$$

Here, assuming that $\eta \leq \frac{2}{\mu + L}$, the second term on the right-hand side is ≤ 0; i.e., we can guarantee that the distance to $x^\star$ decreases per iteration since the first term has

$\left(1-\frac{2\eta\mu L}{\mu+L}\right)<1$. However, this does not say anything about the convergence rate. For that, we observe that:

$$\|x_{t+1}-x^\star\|_2^2 \le \left(1-\tfrac{2\eta\mu L}{\mu+L}\right)\cdot\|x_t-x^\star\|_2^2$$

Assume we use $\eta=\frac{2}{\mu+L}$. Then, we observe:

- $\left(1-\frac{2\eta\mu L}{\mu+L}\right)=\left(1-\frac{2\frac{2}{\mu+L}\mu L}{\mu+L}\right)=\left(1-\frac{4\mu L}{(\mu+L)^2}\right)\ge 0.$
- $\frac{2\frac{2}{\mu+L}\mu L}{\mu+L}=\frac{4\mu L}{(\mu+L)^2}=\frac{4}{\frac{\mu}{L}+2+\frac{L}{\mu}}\ge\frac{2}{\kappa+1}.$

where $\kappa:=\frac{L}{\mu}>1$ is defined as the condition number of f. Then:

$$\begin{aligned}\|x_T-x^\star\|_2^2 &\le \left(1-\tfrac{2}{\kappa+1}\right)\cdot\|x_{T-1}-x^\star\|_2^2\\ &\le \left(1-\tfrac{2}{\kappa+1}\right)^T\cdot\|x_0-x^\star\|_2^2\\ &= \left(\tfrac{\kappa-1}{\kappa+1}\right)^T\cdot\|x_0-x^\star\|_2^2\\ &= O\left(c^T\right)\cdot\|x_0-x^\star\|_2^2\end{aligned}$$

for $c<1$ constant. This is what we call *linear convergence rate*.

To compare the number of iterations required to get to an ε-close solution, we get:

$$\|x_T-x^\star\|_2^2\le\varepsilon \overset{\text{Requires}}{\longrightarrow} \left(\tfrac{\kappa-1}{\kappa+1}\right)^T\cdot\|x_0-x^\star\|_2^2\le\varepsilon$$
$$T\ge\frac{\log\left(\|x_0-x^\star\|_2^2/\varepsilon\right)}{\log\frac{\kappa+1}{\kappa-1}}.$$

Compared to just L-smooth convex functions, we have:

$$O\left(\tfrac{1}{\varepsilon}\right)\quad\text{vs}\quad O\left(\log\tfrac{1}{\varepsilon}\right).$$

Thus, if we require a solution that is $\varepsilon=10^{-3}$-close in some sense, for L-smooth functions, we require $O(1000)$ iterations, while for strongly convex functions, we require $O(3)$ iterations (hiding though a lot of constants). Moreover, observe that the premise in the strongly convex case is stronger: we are guaranteed to converge to the unique global solution, while L-smoothness convex itself cannot guarantee anything about which global solution we converge to.

Please revisit the figures in previous chapters for an illustration and comparison between different convergence rates.

What should our expectations be: Lower bounds. Let us summarize some of our results, especially under the convexity assumption. We know that:

- For L-smooth convex functions, we have:

$$f(x_T)-f(x^\star)\le\frac{2L\left(f(x_0)-f(x^\star)\right)\cdot\|x_0-x^\star\|_2^2}{2L\|x_0-x^\star\|_2^2+T\cdot\left(f(x_0)-f(x^\star)\right)}.$$

- When we also have strong convexity:

$$\|x_T-x^\star\|_2^2\le\left(\tfrac{\kappa-1}{\kappa+1}\right)^T\cdot\|x_0-x^\star\|_2^2.$$

But is this the best we can achieve when dealing only with L-smooth convex functions? E.g., is there another analysis that leads to $f(x_T)-f(x^\star)\le c^T$, for some $c<1$, when only L-smoothness holds? Can we achieve a better convergence rate under L-smooth and μ-strong convexity?

The above leads to the discussion on *lower bounds*: i.e., making the same assumptions—and no more—can we construct functions f that, under these assumptions, we cannot achieve something better than the above?[7]

The following summarizes lower bounds on the types of objective functions we have previously discussed.

- For objective functions with Lipschitz continuous gradients, with constant L, we can prove that there are f instances such that we cannot achieve something better than:

$$f(x_T)-f(x^\star)\ge\frac{3L\|x_0-x^\star\|_2^2}{32(T+1)^2}=O\left(\tfrac{1}{T^2}\right).$$

  Under this assumption, and only using gradients, we cannot achieve better than the above.
- For objective functions with both Lipschitz continuous gradients and the strong convexity assumption satisfied, there are f instances with a convergence rate lower bounded by:

$$\|x_T-x^\star\|_2^2\ge\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2T}\|x_0-x^*\|_2^2,$$

  where $\kappa=L/\mu>1$. Here we observe that, while we have achieved the same convergence rate with respect to the exponent—i.e.., in both cases, we have c^T, for $c<1$—in the lower bound case, we see $\sqrt{\kappa}$ instead of κ.

But how do we obtain such lower bounds? By constructing special functions f that satisfy our assumptions and provably show such lower bounds behavior in theory.[8]

Later in the course, we will see how to achieve these lower bounds under the same assumptions and rely only on a first-order oracle.

Other powerful global assumptions. Convexity is a strong assumption that every local minimum is equivalent to a global minimum. In math, along with L-smoothness and strong convexity, we use the basic condition:

$$f(x)\ge f(y)+\langle\nabla f(y),x-y\rangle,$$

to obtain the results above. *But are there any other assumptions we can use to prove similar convergence rates?*

In this subsection, we will focus on the notion of *Polyak-Lojasiewicz (PL) inequality*, use it in proof techniques, and conclude with other global assumptions similar to PL.

The definition of PL is as follows:

Definition 19. *A function f satisfies the PL inequality if the following holds for some $\xi>0$:*

$$\tfrac{1}{2}\|\nabla f(x)\|_2^2\ge\xi\cdot\left(f(x)-f(x^\star)\right),\quad\forall x.$$

(*Any thoughts on what this inequality implies, concerning stationary points?*)

Let us use this new definition to prove convergence. We will assume L-smoothness of f (*this does not imply anything about convexity*). Using step size $\eta=\frac{1}{L}$ in gradient descent leads to:

$$f(x_{t+1})-f(x_t)\le-\tfrac{1}{2L}\cdot\|\nabla f(x_t)\|_2^2,\forall t.$$

[7] This also includes the assumption that we will only use first-order oracles; if we had the option to use more information– -say Hessians– -then we could achieve more. We will defer this discussion to the chapters that follow.

[8] However, is this a pessimistic way of thinking convergence rates? For the careful reader, this is similar to characterizing a problem NP-hard by the time we find an instance that is NP-hard. Does this hold, though, for the most practical f cases? Food for thought.

By PL, we know that:

$$-\tfrac{1}{2}\|\nabla f(x_t)\|_2^2 \leq -\xi \cdot (f(x_t) - f(x^\star)) .$$

Then:

$$\begin{aligned} &f(x_{t+1}) - f(x_t) \leq -\tfrac{\xi}{L} \cdot (f(x_t) - f(x^\star)) \quad \Rightarrow \\ &f(x_{t+1}) - f(x^\star) \leq f(x_t) - f(x^\star) - \tfrac{\xi}{L} \cdot (f(x_t) - f(x^\star)) \quad \Rightarrow \\ &f(x_{t+1}) - f(x^\star) \leq \left(1 - \tfrac{\xi}{L}\right) \cdot (f(x_t) - f(x^\star)) . \end{aligned}$$

Unfolding this recursion:

$$f(x_T) - f(x^\star) \leq \left(1 - \tfrac{\xi}{L}\right)^T \cdot (f(x_0) - f(x^\star)) .$$

Under the assumption that $L \geq \xi$, this leads to a linear convergence rate.

Some comments:

- We proved linear convergence to the global optimum without assuming strong convexity. This dictates that one might make different assumptions that lead to favorable behavior.
- Further, PL inequality does not imply convexity; i.e., we proved convergence with linear rate to the global optimum, even if the objective is not convex.
- PL assumption does not imply uniqueness of the global optimum; there might be several $x^\star$ (*one of the reasons we do not have convergence guarantees in* $\|x_t - x^\star\|_2$ *terms*).

What does a function that satisfies PL inequality look like? Here is an example we have seen in the previous chapter.

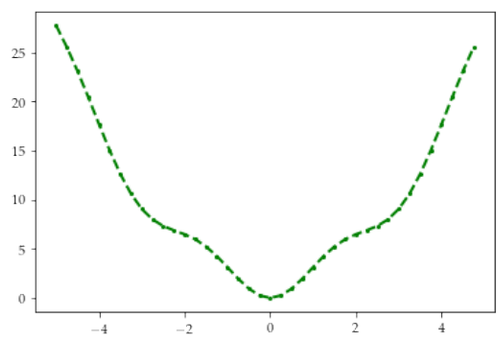

Fig. 23. $f(x) = x^2 + 3\sin^2(x)$

Some other conditions that have been used in convergence proofs, but we will not focus on this chapter, are:

Definition 20. *A function f satisfies the weak strong convexity (WSC) condition if the following holds for some $\mu > 0$:*

$$f(x^\star) \geq f(x) + \langle \nabla f(x), x^\star - x \rangle + \tfrac{\mu}{2}\|x - x^\star\|_2^2, \quad \forall x.$$

Observe that this inequality holds for only $x^\star$ on the left-hand side; this justifies the term "weak" in its name.

Definition 21. *A function f satisfies the restricted secant inequality (RSI) if the following holds for some $\mu > 0$:*

$$\langle \nabla f(x), x - x^\star \rangle \geq \mu\|x - x^\star\|_2^2, \quad \forall x.$$

If the function f is also convex, this is also called restricted strong convexity.

Definition 22. *A function f satisfies the error bound condition (EB) if the following holds for some $\mu > 0$:*

$$\|\nabla f(x)\|_2 \geq \mu\|x - x^\star\|_2, \quad \forall x.$$

Definition 23. *A function f satisfies the quadratic growth (QG) condition if the following holds for some $\mu > 0$:*

$$f(x) - f(x^\star) \geq \tfrac{\mu}{2}\|x - x^\star\|_2, \quad \forall x.$$

In the above definitions, μ does not dictate the same value for all definitions; we use the same letter for clarity.

Finally, there is a hierarchy of these conditions.

$$\texttt{(WSC)} \to \texttt{(RSI)} \to \texttt{(EB)} \equiv \texttt{(PL)} \to \texttt{(QG)}$$

The above indicates "implications": e.g., if we assume WSC holds for a function, then the rest of the conditions are also satisfied for some constants μ [36].

When we know neither L nor μ. In this subsection, we will present one (of many) adaptive step size schedules that are theoretically justified. In the deep learning literature, various step size schedules are pretty successful (and are used dominantly in practical scenarios), but remain heuristics which could make them questionable if one "jumps" from one application to another, or even from one dataset to another for the same application. Traditionally, selecting a good step size that does not rely heavily on knowing some "hard-to-know or hard-to-approximate" constants is an active research area with deep roots in the past.

Here, we will focus on the case of the *Polyak step size*,[9] which comes with convergence guarantees for convex functions but does not rely on unknown constants (*there is a caveat here as we will see later on, but that caveat is milder than assuming we know L or μ*). Polyak derived the Polyak step size in 1987 [40]; for generality purposes, we will consider the case where the f function is just convex, even non-smooth. In this scenario, we will focus on the generic (sub)gradient descent algorithm on convex function f:

$$\min_x f(x)$$

that follows the recursion:

$$x_{t+1} = x_t - \eta_t g_t,$$

where g_t represents one of the subgradients of f at point x_t. Let us also assume that $\|g_t\|_2 < G$ for some constant G. A similar analysis to the above theorems leads to the following:

$$\|x_{t+1} - x^\star\|_2^2 = \|x_t - x^\star\|_2^2 + \eta_t^2\|g_t\|_2^2 - 2\eta_t \langle g_t,\ x_t - x^\star \rangle .$$

We know that by convexity, the following inequality holds:

$$f(x^\star) \geq f(x_t) + \langle g_t,\ x_t - x^\star \rangle$$

and the above expression is bounded as:

$$\|x_{t+1} - x^\star\|_2^2 \leq \|x_t - x^\star\|_2^2 + \eta_t^2\|g_t\|_2^2 - 2\eta_t \left(f(x_t) - f(x^\star)\right) .$$

To choose the step size, one option is to choose η_t such that the right-hand side is minimized, i.e.,

$$\eta_t = \arg\min_\eta \left\{ \|x_t - x^\star\|_2^2 + \eta_t^2\|g_t\|_2^2 - 2\eta_t \left(f(x_t) - f(x^\star)\right) \right\} ,$$

which has a closed-form solution:

$$\eta_t = \tfrac{f(x_t) - f(x^\star)}{\|g_t\|_2^2}.$$

[9] Other techniques for finding an acceptable step size/learning rate/step length include Wolfe conditions [37], Armijo conditions [38], Curvature conditions, Goldstein Conditions [39], Backtracking line search [39]

This is known as the *Polyak step size*. Substituting this step size in the expression above, we obtain:

$$\|x_{t+1} - x^\star\|_2^2 \leq \|x_t - x^\star\|_2^2 - \frac{\left(f(x_t) - f(x^\star)\right)^2}{\|g_t\|_2^2}.$$

I.e., $\|x_{t+1} - x^\star\|_2^2$ monotonically decreases per iteration. By telescoping the above expression over t iterations, we obtain:

$$\|x_{t+1} - x^\star\|_2^2 \leq \|x_0 - x^\star\|_2^2 - \frac{1}{G^2}\sum_{i=0}^{t}\left(f(x_i) - f(x^\star)\right)^2,$$

where we also utilized the assumption $\|g_t\|_2 < G$. Rearranging the above terms and removing any unnecessary terms, we obtain:

$$\min_{i\in[t]} f(x_i) - f(x^\star) \leq \frac{G\cdot\|x_0 - x^\star\|_2}{\sqrt{t+1}} = O(\tfrac{1}{\sqrt{t}}).$$

What is the caveat? Polyak's step size can be used only when the optimal value $f(x^\star)$ is known. Yet, there are references in the literature [41] that demonstrate that $f(x^\star) = 0$ for several applications (for example, finding a point in the intersection of convex sets, positive semidefinite matrix completion and solving convex inequalities). Moreover, in overparameterized neural networks, this is a common assumption that holds also in practice, but it is out of the scope of this chapter.

Constrained convex optimization and convex sets. In Chapter 1, we mentioned that the focus of this class will be a subset of problems of the form:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$

where $\mathcal{C}$ is the constraint set on x. The nature of $\mathcal{C}$ depends on the application; there are applications where $\mathcal{C}$ is simple enough and does not affect much how gradient descent behaves, and there are applications where $\mathcal{C}$ is not of a straightforward form (e.g., think of combinatorial constraints). One such example is the sparsity constraint, where we are looking for a sparse vector that minimizes f (i.e., there might be dense vectors that minimize f even further, but we are interested in sparse solutions).

Similarly to functions, we must define the difference between *convex and non-convex sets*. To understand and appreciate the difficulty of including non-convex constraints, we need to know how simple, convex constraints affect the performance of gradient descent.

When is our problem convex or non-convex? First, it is important to understand what convex optimization can solve and what it can not. When *both the objective and the constraints are convex*, then the problem (in most cases) can be solved by standard convex optimization tools (including gradient descent as a solver). When *either of the objective or the constraints are non-convex, or neither of them are convex*, then the problem is non-convex.

To provide a pictorial explanation, look at the following toy example curve.

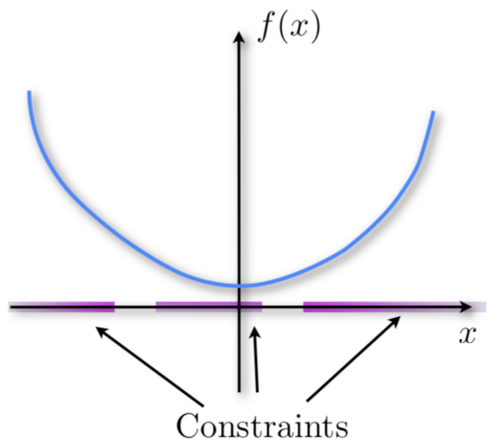

Fig. 24. Constrained optimization example, where the constraint set is non-convex. The purple parts do not belong in the feasibility set.

If we had no constraints, the function would be smooth and convex; thus, gradient descent would work as expected. However, including the purple parts on the feasibility set, the optimal could be no longer the bottom of the "bowl". Also, solving the problem first without the constraints and then applying the constraints most often does not lead to a good solution.

It is natural to study constrained convex optimization first. The following are additional definitions related to convexity that will become important later in the course.

Definition 24. (Convex Set) *The set $\mathcal{C} \subset \mathbb{R}^p$ is a convex set if $\forall x_1, x_2 \in \mathcal{C}$, it holds that*

$$\forall \alpha \in [0,1] : \alpha x_1 + (1-\alpha)x_2 \in \mathcal{C}.$$

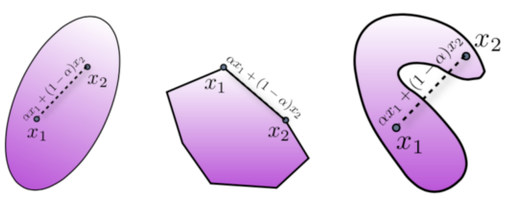

Fig. 25. Some convex set examples.

Definition 25. (Convex Hull) *The convex hull of a set of points in $\mathcal{Q}$ is the intersection of all convex sets containing $\mathcal{Q}$. For n points $\mathcal{Q} := \{x_1, \ldots, x_n\}$, the convex hull is*

$$\text{conv}(\mathcal{Q}) = \left\{ \sum_{j=1}^{n} \alpha_j x_j \ : \alpha_j \geq 0, \ \forall j, \ \sum_{j=1}^{n} \alpha_j = 1 \right\}.$$

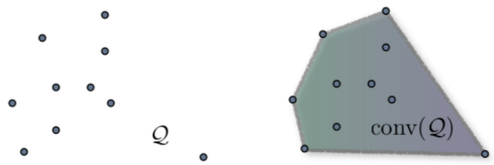

Fig. 26. Convex hull of a set of points.

Some notable convex sets:

- Linear spaces and halfspaces: e.g., $\{x \in \mathbb{R}^p \mid Ax = 0\}$ and $\{x \in \mathbb{R}^p \mid \langle z, x\rangle \geq 0\}$.
- Affine transformations of convex sets: e.g., if $\mathcal{C}$ is a convex set, then so it is the set $\{Ax + b \mid x \in \mathcal{C}\}$.
- Intersections of convex sets.
- Special cases that are worth to be mentioned: Norm inequality constraints define convex sets (e.g., $\|x\|_2 \leq 1$, $\|x\|_1 \leq \lambda$, $\|X\|_F \leq c$, $\|y - Ax\|_2 \leq \varepsilon$—*however, the following set* $\|x\|_2 = 1$ *is not convex, why?*); linear constraints define convex sets, such as $Ax \leq b$; linear matrix inequalities define convex sets— particular case the PSD constraint, $A \succeq 0$.

Projections onto convex sets. The definition of a projection onto a set is as the following optimization problem:

$$\Pi_{\mathcal{C}}(x) = \underset{z \in \mathcal{C}}{\operatorname{argmin}}\, \ell(x, z).$$

Here, $\mathcal{C}$ defines the set on which we want to project, x is a given point, and $\ell(x, z)$ defines a notion of distance between x and a point in $\mathcal{C}$, which we want to minimize. Classical examples for $\ell(x, z)$ are norms such as $\ell(x, z) = \|x - z\|_2^2$, which will be the focus here. Thus, a verbal description of

$$\Pi_{\mathcal{C}}(x) = \underset{z \in \mathcal{C}}{\operatorname{argmin}} \|x - z\|_2^2$$

is *"Given a point x and a set $\mathcal{C}$, find a point in $\mathcal{C}$ that is closer to x with respect to the Euclidean distance"*. When the set $\mathcal{C}$ is convex, this defines the Euclidean projection onto the convex set $\mathcal{C}$.

An illustration of the above is shown below.

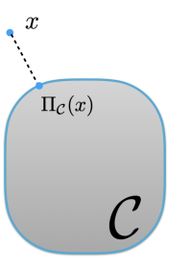

Fig. 27. Projection onto convex set $\mathcal{C}$.

Some useful properties for projections onto convex sets are:

- $\|x - \Pi_{\mathcal{C}}(x)\|_2^2 \leq \|x - y\|_2^2, \ \forall y \in \mathcal{C}, \ \forall x$, with the following illustration:

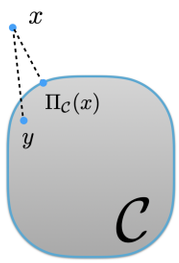

Fig. 28. The Euclidean distance from x to its projection onto $\mathcal{C}$ is the smallest among the points in $\mathcal{C}$.

This is the definition of the projection as the minimum distance.

- $\langle \Pi_{\mathcal{C}}(x) - y, \ \Pi_{\mathcal{C}}(x) - x\rangle \leq 0, \ \forall y \in \mathcal{C}, \ \forall x$, with the following illustration:

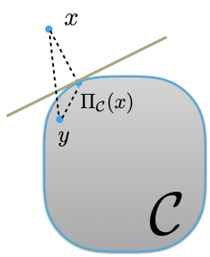

Fig. 29. The angle between the rays $\Pi_{\mathcal{C}}(x) - y$ and $\Pi_{\mathcal{C}}(x) - x$ are more than 90^o.

The interpretation is given in the caption above. This property does not hold for non-convex functions; a counterexample is given in the following figure.

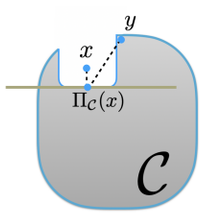

Fig. 30. The above property does not necessarily hold for non-convex sets.

- $\|\Pi_{\mathcal{C}}(x) - \Pi_{\mathcal{C}}(y)\|_2 \leq \|x - y\|_2, \ \forall x, y.$

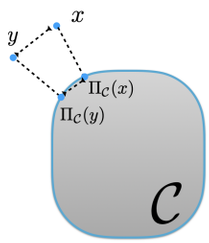

Fig. 31. The distance between any two points is greater than the distance of their projections onto a convex set.

Projected gradient descent. Given the notion of projections, we can define the projected version of gradient descent:

$$x_{t+1} = \Pi_{\mathcal{C}}\left(x_t - \eta_t \nabla f(x_t)\right), \quad t = 0, 1, \dots,$$

which can be alternatively seen as a two-step procedure:

$$\begin{aligned} \widetilde{x}_{t+1} &= x_t - \eta_t \nabla f(x_t), \quad t = 0, 1, \dots, \\ x_{t+1} &= \Pi_{\mathcal{C}}\left(\widetilde{x}_{t+1}\right), \quad t = 0, 1, \dots, \end{aligned}$$

But do we lose anything by including the projection step? Can we preserve the same convergence guarantees?

Claim 5. *For a function $f : \mathbb{R}^p \to \mathbb{R}$ that is L-smooth and μ-strongly convex, projected gradient descent converges according to:*

$$\|x_{t+1} - x^\star\|_2^2 \leq \left(1 - \tfrac{2\eta\mu L}{\mu + L}\right) \|x_t - x^\star\|_2^2.$$

Proof: By definition, $x_{t+1} = \Pi_C(x_t - \eta \nabla f(x_t))$. So,

$$\begin{aligned} \|x_{t+1} - x^\star\|_2^2 &= \|\Pi_{\mathcal{C}}\left(x_t - \eta \nabla f(x_t)\right) - x^\star\|_2^2 \\ &= \|\Pi_{\mathcal{C}}\left(x_t - \eta \nabla f(x_t)\right) - \Pi_{\mathcal{C}}(x^\star)\|_2^2 \\ &\leq \|x_t - \eta \nabla f(x_t) - x^\star\|_2^2 \\ &\leq \dots \quad \text{(Similar analysis to GD)} \\ &\leq \left(1 - \tfrac{2\eta\mu L}{\mu + L}\right) \|x_t - x^\star\|_2^2. \end{aligned}$$

□

Convergence of projected gradient descent for L-smooth functions. A similar analysis holds for the case of just L-smooth functions. However, we need some care to handle the analysis in f values. Remember that for just L-smooth functions, there might be multiple global solutions $x^\star$ that minimize the objective, and thus the notion of a distance $\|x_{t+1} - x^\star\|_2$ does not make sense in a recursion.

We know from the analysis of unconstrained optimization that, for L-smooth functions and for step size $\eta = \frac{1}{L}$ in gradient descent:

$$f(x_{t+1}) - f(x_t) \leq -\tfrac{1}{2L} \|\nabla f(x_t)\|_2^2.$$

This is based on the application of L-smoothness, where:

$$\begin{aligned} f(x_{t+1}) &\leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \tfrac{L}{2} \|x_{t+1} - x_t\|_2^2 \\ &\overset{(i)}{=} f(x_t) + \langle \nabla f(x_t), x_t - \eta_t \nabla f(x_t) - x_t \rangle \\ &\qquad + \tfrac{L}{2} \|x_t - \eta_t \nabla f(x_t) - x_t\|_2^2 \\ &= \cdots \end{aligned}$$

Though, in a constrained case such as,

$$\begin{aligned} \min_{x \in \mathbb{R}^p} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$

we have:

$$x_{t+1} = \Pi_{\mathcal{C}}\left(x_t - \eta \nabla f(x_t)\right),$$

which complicates things, so that equation (i) does not hold.

We will need the notion of *gradient mapping* to overcome this difficulty. Without getting into many details (*gradient mapping is not going to be used for the rest of the course*), we can prove the following result:

Lemma 5. *Let $\mathcal{C} \subset \mathbb{R}^p$ be a convex set, and let $x, y \in \mathcal{C}$. Let $f : \mathcal{C} \to \mathbb{R}$ be a convex function that we want to minimize and satisfies L-smoothness. Define: $x^+ = \Pi_{\mathcal{C}}\left(x - \frac{1}{L}\nabla f(x)\right)$. Define also the function $g_{\mathcal{C}}(x) = L \cdot (x - x^+)$. Then, the following inequality holds:*

$$f(x^+) - f(y) \leq \langle g_{\mathcal{C}}(x), x - y \rangle - \tfrac{1}{2L} \|g_{\mathcal{C}}(x)\|_2^2.$$

Proof: Since $\mathcal{C}$ is a convex set, by the projection properties, we know that:

$$\begin{aligned} &\left\langle x^+ - \left(x - \tfrac{1}{L}\nabla f(x)\right),\ x^+ - y \right\rangle \leq 0 \Rightarrow \\ &\left\langle x^+ - x,\ x^+ - y \right\rangle + \tfrac{1}{L} \left\langle \nabla f(x),\ x^+ - y \right\rangle \leq 0 \Rightarrow \\ &\qquad \tfrac{1}{L} \left\langle \nabla f(x),\ x^+ - y \right\rangle \leq \left\langle x - x^+,\ x^+ - y \right\rangle \Rightarrow \\ &\qquad\quad \left\langle \nabla f(x),\ x^+ - y \right\rangle \leq \left\langle g_{\mathcal{C}}(x),\ x^+ - y \right\rangle \end{aligned}$$

Then, we observe the following series of (in)equalities:

$$\begin{aligned} f(x^+) - f(y) &= f(x^+) - f(x) + f(x) - f(y) \\ &\leq \left\langle \nabla f(x),\ x^+ - x \right\rangle + \tfrac{L}{2} \|x^+ - x\|_2^2 + \langle \nabla f(x),\ x - y \rangle \\ &= \left\langle \nabla f(x),\ x^+ - y \right\rangle + \tfrac{1}{2L} \|g_{\mathcal{C}}(x)\|_2^2 \\ &\leq \left\langle g_{\mathcal{C}}(x),\ x^+ - y \right\rangle + \tfrac{1}{2L} \|g_{\mathcal{C}}(x)\|_2^2 \\ &= \langle g_{\mathcal{C}}(x),\ x - y \rangle - \tfrac{1}{2L} \|g_{\mathcal{C}}(x)\|_2^2 \end{aligned}$$

where the first inequality is due to L-smoothness and convexity.

□

Given the above and the recursion of projected gradient descent:

$$x_{t+1} = \Pi_{\mathcal{C}}\left(x_t - \eta \nabla f(x_t)\right),$$

by the lemma above, we can compute the following:

$$f(x_{t+1}) - f(x_t) \leq -\tfrac{1}{2L} \|g_{\mathcal{C}}(x_t)\|_2^2,$$

and

$$f(x_{t+1}) - f(x^\star) \leq \|g_{\mathcal{C}}(x_t)\|_2 \cdot \|x_t - x^\star\|_2$$

Using similar analysis with the unconstrained case, we can prove:

$$\Delta_T \leq \frac{3L\|x_1 - x^\star\|_2^2 + (f(x_1) - f(x^\star))}{T},$$

where showing $\|x_{t+1} - x^\star\|_2 \leq \|x_t - x^\star\|_2$ stems from the use of the above lemma.

Thus, overall, projections do not change the convergence rate. However, it changes the per iteration complexity: e.g., consider a projection procedure as challenging to complete as the original problem.

Opinion: Convex optimization is a technology. Convex optimization has become one of the most well-studied and well-understood areas of optimization; another such area is that of linear programming. To this point, there are several off-the-shelf solvers that are available online

- `CVXOPT` - `https://cvxopt.org`
- `CVXPy` - `http://www.cvxpy.org/`
- `CVX` - `http://cvxr.com/cvx/`
- `JuliaOpt` - `https://www.juliaopt.org/`
- TensorFlow - `https://www.tensorflow.org/`
- PyTorch - `https://pytorch.org/`

Why do we still care about convex optimization?

- Several practical problems are convex.
- Many practical problems can be approximated by convex ones
- If one does not understand convex optimization, why even try understanding non-convex optimization? :)

———— ∞ ————

As an exciting interlude, we will continue with the discussion that started at the beginning of this chapter. Consider the minimization problem of the function:

$$f(x) = \tfrac{1}{2}x^\top Qx - b^\top x + r,$$

where $x \in \mathbb{R}^p$, $Q \in \mathbb{R}^{p\times p}$ is a symmetric matrix, $b \in \mathbb{R}^p$ is a vector and r is a scalar. I.e.,

$$\min_{x\in\mathbb{R}^p} f(x).$$

Here, we follow the discussion in this chapter, where $\mu \cdot I \preceq Q \preceq L \cdot I$. Further, we know that $\nabla f(x) = Qx - b = Q(x - x^\star)$, assuming that $x^\star$ solves the problem and thus $Qx^\star = b$.

By definition of gradient descent, we have the following recursion:

$$x_{t+1} - x^\star = (I - \eta Q) \cdot (x_t - x^\star).$$

This further implies that the quantities $x_t - x^\star$ can be exactly computed by recursive applying the above rule, all the way to the initial conditions x_0, to obtain:

$$x_t - x^\star = P_t(Q) \cdot (x_0 - x^\star),$$

where

$$P_t(Q) = (I - \eta Q)^t.$$

Here, $P_t(Q)$ is a *matrix polynomial*, whose value depends on the behavior of an eigenvalue-based polynomial; the degree of this polynomial depends on the number of times we repeat this process, i.e., the number of iterations of the gradient descent method, t.

In the discussion at the beginning of this chapter, we assumed that $\|I - \eta Q\|_2 < 1$ to apply the Cauchy-Schwarz rule and obtain that result recursively. Here, we would like to highlight a different perspective of gradient-based methods, that of *minimax polynomials* and how these polynomials could define the type of gradient-based method we use (or would like to use). To do so, we take one step back and define as first-order methods (for quadratic optimization) all the methods that generate solutions after t iterations, as follows:

$$x_{t+1} \in x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_t)\}.$$

I.e., for any iteration t, the estimate we obtain is a linear combination of the initial point x_0 and the gradient vectors computed $\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_t)\}$, up to this point. To see that gradient descent is an algorithm that belongs to this class, consider the following (with constant step size):

$$\begin{aligned} x_{t+1} &= x_t - \eta\nabla f(x_t) = x_{t-1} - \eta\nabla f(x_{t-1}) - \eta\nabla f(x_t) \\ &= \cdots = x_0 - \eta\sum_{i=0}^{t}\nabla f(x_i) \\ &\in x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_t)\}. \end{aligned}$$

Given the above, we are now stating an interesting theorem (the proof is skipped, but it is not difficult to show this using mathematical induction).

Theorem 2. *Let starting point $x_0 \in \mathbb{R}^p$; consider the quadratic function, as defined above. Then, the sequence of points:*

$$x_{t+1} \in x_0 + span\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_t)\}$$

also satisfy the expression:

$$x_t - x^\star = P_t(Q)(x_0 - x^\star),$$

for some sequence of polynomials $\{P_t\}_{t=0,1,\ldots}$, with P_t of degree at most t, and initial conditions $P_t(0) = 1$, and vice-versa ("if and only if" condition).

But why is this important? If you think about it, this theorem does not specify an algorithm; it just states that this should hold for any gradient-based algorithm that "accumulates" gradient estimates. However, all these methods can be nicely characterized (in terms of convergence and convergence rates) by finding the associated matrix polynomial P_t and deriving the worst-case analysis on that polynomial. I.e., given the collection of symmetric matrices Q, denoted as $\mathcal{Q}$ and given that we know that $\|x_t - x^\star\| \leq \|P_t(Q)\|_2 \cdot \|x_0 - x^\star\|_2$, one could try to characterize what is the best polynomial P_t (as in minimizer of the quantity $\|P_t(Q)\|_2$, with respect to P_t) over the worst possible problem instance wrt Q (as in maximizer of the quantity $\|P_t(Q)\|_2$, with respect to Q). In other words, one could search over all possible P_t's such that:

$$P_t^\star = \arg\min_{P:P(0)=1}\max_{Q\in\mathcal{Q}}\|P(Q)\|_2.$$

As we will see in future chapters, one can find new algorithms that are optimal (= faster) than gradient descent by only looking into what the "limits" of matrix polynomials in terms of minimizing the worst-case scenario are; then, reverse-engineering these polynomials, we obtain specific algorithmic constructions that lead to variants of gradient descent with provably better performance (spoiler alert: accelerated gradient descent methods).

1. J. Nocedal and S. Wright. Numerical optimization. Springer Science & Business Media, 2006.
2. Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.
3. S. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004.
4. D. Bertsekas. Convex optimization algorithms. Athena Scientific Belmont, 2015.
5. Sébastien Bubeck. Convex optimization: Algorithms and complexity. Foundations and Trends® in Machine Learning, 8(3-4):231–357, 2015.
6. S. Weisberg. Applied linear regression, volume 528. John Wiley & Sons, 2005.
7. T. Hastie, R. Tibshirani, and M. Wainwright. Statistical learning with sparsity: the lasso and generalizations. CRC press, 2015.
8. J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning, volume 1. Springer series in statistics New York, 2001.
9. M. Paris and J. Rehacek. Quantum state estimation, volume 649. Springer Science & Business Media, 2004.
10. M. Daskin. A maximum expected covering location model: formulation, properties and heuristic solution. Transportation science, 17(1):48–70, 1983.
11. I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. MIT press, 2016.
12. L. Trefethen and D. Bau III. Numerical linear algebra, volume 50. Siam, 1997.
13. G. Strang. Introduction to linear algebra, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.
14. G. Golub. Cmatrix computations. The Johns Hopkins, 1996.
15. Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In Neural networks: Tricks of the trade, pages 9–50. Springer, 2002.
16. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015.
17. Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31, 2018.
18. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
19. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
20. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
21. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
22. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
23. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1243–1252. JMLR. org, 2017.
24. Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association, 2014.
25. Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun. Very deep multilingual convolutional neural networks for LVCSR. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4955–4959. IEEE, 2016.
26. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. page arXiv:1706.03762, 2017.
27. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. page arXiv:1810.04805, 2018.
28. Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In AAAI, pages 13041–13049, 2020.
29. Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.
30. Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. arXiv preprint arXiv:1909.08053, 2019.
31. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683, 2019.
32. Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of DALL-E 2. arXiv preprint arXiv:2204.13807, 2022.
33. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873):583–589, 2021.
34. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
35. Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. arXiv preprint arXiv:2004.08900, 2020.
36. H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 795–811. Springer, 2016.
37. Philip Wolfe. Convergence conditions for ascent methods. SIAM review, 11(2):226–235, 1969.
38. Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.
39. Stephen Wright and Jorge Nocedal. Numerical optimization. Springer Science, 35(67-68):7, 1999.
40. B. Polyak. Introduction to optimization. Inc., Publications Division, New York, 1, 1987.
41. Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004:2004–2005, 2003.
42. Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. Naval research logistics quarterly, 3(1-2):95–110, 1956.
43. M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th international conference on machine learning, number CONF, pages 427–435, 2013.
44. J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In Proceedings of the 25th international conference on Machine learning, pages 272–279, 2008.
45. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, (8):30–37, 2009.
46. A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In Advances in neural information processing systems, pages 1257–1264, 2008.
47. T. Booth and J. Gubernatis. Improved criticality convergence via a modified Monte Carlo power iteration method. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
48. S. Zavriev and F. Kostyuk. Heavy-ball method in nonconvex optimization problems. Computational Mathematics and Modeling, 4(4):336–341, 1993.
49. E. Ghadimi, H. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In 2015 European control conference (ECC), pages 310–315. IEEE, 2015.
50. Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$. In Soviet Mathematics Doklady, volume 27, pages 372–376, 1983.
51. B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics, 15(3):715–732, 2015.
52. O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. Mathematical Programming, 146(1-2):37–75, 2014.
53. L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. Siam Review, 60(2):223–311, 2018.
54. S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. SIAM review, 43(1):129–159, 2001.
55. R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.
56. P. Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. Computational Statistics & Data Analysis, 115:186–198, 2017.
57. S. Becker, J. Bobin, and E. Candès. NESTA: A fast and accurate first-order method for sparse recovery. SIAM Journal on Imaging Sciences, 4(1):1–39, 2011.
58. T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.
59. D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and computational harmonic analysis, 26(3):301–321, 2009.
60. S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis, 49(6):2543–2563, 2011.
61. J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. SIAM Journal on Scientific Computing, 35(5):S104–S125, 2013.
62. K. Wei. Fast iterative hard thresholding for compressed sensing. IEEE Signal processing letters, 22(5):593–597, 2014.
63. Rajiv Khanna and Anastasios Kyrillidis. Iht dies hard: Provable accelerated iterative hard thresholding. In International Conference on Artificial Intelligence and Statistics, pages 188–198. PMLR, 2018.
64. Jeffrey D Blanchard and Jared Tanner. GPU accelerated greedy algorithms for compressed sensing. Mathematical Programming Computation, 5(3):267–304, 2013.
65. A. Kyrillidis, G. Puy, and V. Cevher. Hard thresholding with norm constraints. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3645–3648. Ieee, 2012.
66. A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on, pages 353–356. IEEE, 2011.

67. X. Zhang, Y. Yu, L. Wang, and Q. Gu. Learning one-hidden-layer ReLU networks via gradient descent. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1524–1534, 2019.
68. Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory, 52(2):489–509, 2006.
69. Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. Optimization Methods & Software, 23(4):501–520, 2008.
70. Joseph B Altepeter, Daniel FV James, and Paul G Kwiat. 4 qubit quantum state tomography. In Quantum state estimation, pages 113–145. Springer, 2004.
71. Jens Eisert, Dominik Hangleiter, Nathan Walk, Ingo Roth, Damian Markham, Rhea Parekh, Ulysse Chabaud, and Elham Kashefi. Quantum certification and benchmarking. arXiv preprint arXiv:1910.06343, 2019.
72. Masoud Mohseni, AT Rezakhani, and DA Lidar. Quantum-process tomography: Resource analysis of different strategies. Physical Review A, 77(3):032322, 2008.
73. D. Gross, Y.-K. Liu, S. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. Physical review letters, 105(15):150401, 2010.
74. Y.-K. Liu. Universal low-rank matrix recovery from Pauli measurements. In Advances in Neural Information Processing Systems, pages 1638–1646, 2011.
75. K Vogel and H Risken. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. Physical Review A, 40(5):2847, 1989.
76. Miroslav Ježek, Jaromír Fiurášek, and Zdeněk Hradil. Quantum inference of states and processes. Physical Review A, 68(1):012305, 2003.
77. Konrad Banaszek, Marcus Cramer, and David Gross. Focus on quantum tomography. New Journal of Physics, 15(12):125020, 2013.
78. A. Kalev, R. Kosut, and I. Deutsch. Quantum tomography protocols with positivity are compressed sensing protocols. Nature partner journals (npj) Quantum Information, 1:15018, 2015.
79. Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. Nat. Phys., 14:447–450, May 2018.
80. Matthew JS Beach, Isaac De Vlugt, Anna Golubeva, Patrick Huembeli, Bohdan Kulchytskyy, Xiuzhe Luo, Roger G Melko, Ejaaz Merali, and Giacomo Torlai. Qucumber: wavefunction reconstruction with neural networks. SciPost Physics, 7(1):009, 2019.
81. Giacomo Torlai and Roger Melko. Machine-learning quantum states in the NISQ era. Annual Review of Condensed Matter Physics, 11, 2019.
82. M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu. Efficient quantum state tomography. Nat. Comm., 1:149, 2010.
83. BP Lanyon, C Maier, Milan Holzäpfel, Tillmann Baumgratz, C Hempel, P Jurcevic, Ish Dhand, AS Buyskikh, AJ Daley, Marcus Cramer, et al. Efficient tomography of a quantum many-body system. Nature Physics, 13(12):1158–1162, 2017.
84. D. Gonçalves, M. Gomes-Ruggiero, and C. Lavor. A projected gradient method for optimization over density matrices. Optimization Methods and Software, 31(2):328–341, 2016.
85. E. Bolduc, G. Knee, E. Gauger, and J. Leach. Projected gradient descent algorithms for quantum state tomography. npj Quantum Information, 3(1):44, 2017.
86. Jiangwei Shang, Zhengyun Zhang, and Hui Khoon Ng. Superfast maximum-likelihood reconstruction for quantum tomography. Phys. Rev. A, 95:062336, Jun 2017.
87. Zhilin Hu, Kezhi Li, Shuang Cong, and Yaru Tang. Reconstructing pure 14-qubit quantum states in three hours using compressive sensing. IFAC-PapersOnLine, 52(11):188 – 193, 2019. 5th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2019.
88. Zhibo Hou, Han-Sen Zhong, Ye Tian, Daoyi Dong, Bo Qi, Li Li, Yuanlong Wang, Franco Nori, Guo-Yong Xiang, Chuan-Feng Li, et al. Full reconstruction of a 14-qubit state within four hours. New Journal of Physics, 18(8):083036, 2016.
89. C. Riofrío, D. Gross, S.T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert. Experimental quantum compressed sensing for a seven-qubit system. Nature Communications, 8, 2017.
90. Martin Kliesch, Richard Kueng, Jens Eisert, and David Gross. Guaranteed recovery of quantum processes from few measurements. Quantum, 3:171, 2019.
91. S. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. Quantum tomography via compressed sensing: Error bounds, sample complexity and efficient estimators. New Journal of Physics, 14(9):095022, 2012.
92. A. Kyrillidis, A. Kalev, D. Park, S. Bhojanapalli, C. Caramanis, and S. Sanghavi. Provable quantum state tomography via non-convex methods. npj Quantum Information, 4(36), 2018.
93. B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review, 52(3):471–501, 2010.
94. N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In Advances in neural information processing systems, pages 1329–1336, 2004.
95. J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd international conference on Machine learning, pages 713–719. ACM, 2005.
96. D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In Proceedings of the 23rd international conference on Machine learning, pages 249–256. ACM, 2006.
97. J. Bennett and S. Lanning. The Netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.
98. M. Jaggi and M. Sulovsk. A simple algorithm for nuclear norm regularized problems. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 471–478, 2010.
99. R. Keshavan. Efficient algorithms for collaborative filtering. PhD thesis, Stanford University, 2012.
100. R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In Proceedings of the 22nd international conference on World Wide Web, pages 13–24. International World Wide Web Conferences Steering Committee, 2013.
101. K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In Advances in Neural Information Processing Systems, pages 730–738, 2015.
102. G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(3):394–410, 2007.
103. A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In Computer Vision–ECCV 2008, pages 316–329. Springer, 2008.
104. C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1643–1650. IEEE, 2009.
105. J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In IJCAI, volume 11, pages 2764–2770, 2011.
106. Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. In AISTATS, 2003.
107. K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. Dhillon, and A. Tewari. Prediction and clustering in signed networks: A local to global perspective. The Journal of Machine Learning Research, 15(1):1177–1213, 2014.
108. C. Johnson. Logistic matrix factorization for implicit feedback data. Advances in Neural Information Processing Systems, 27, 2014.
109. Koen Verstrepen. Collaborative Filtering with Binary, Positive-only Data. PhD thesis, University of Antwerpen, 2015.
110. N. Gupta and S. Singh. Collectively embedding multi-relational data for predicting user preferences. arXiv preprint arXiv:1504.06165, 2015.
111. Y. Liu, M. Wu, C. Miao, P. Zhao, and X.-L. Li. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. PLoS Computational Biology, 12(2):e1004760, 2016.
112. S. Aaronson. The learnability of quantum states. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 463, pages 3089–3114. The Royal Society, 2007.
113. E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. SIAM Review, 57(2):225–251, 2015.
114. I. Waldspurger, A. d'Aspremont, and S. Mallat. Phase recovery, MaxCut and complex semidefinite programming. Mathematical Programming, 149(1-2):47–81, 2015.
115. P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. IEEE transactions on automation science and engineering, 3(4):360, 2006.
116. K. Weinberger, F. Sha, Q. Zhu, and L. Saul. Graph Laplacian regularization for large-scale semidefinite programming. In Advances in Neural Information Processing Systems, pages 1489–1496, 2007.
117. F. Lu, S. Keles, S. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. Proceedings of the National Academy of Sciences of the United States of America, 102(35):12332–12337, 2005.
118. H. Andrews and C. Patterson III. Singular value decomposition (SVD) image coding. Communications, IEEE Transactions on, 24(4):425–432, 1976.
119. M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In American Control Conference, 2004. Proceedings of the 2004, volume 4, pages 3273–3278. IEEE, 2004.
120. E. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717–772, 2009.
121. P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In Advances in Neural Information Processing Systems, pages 937–945, 2010.
122. S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In 10th International Conference on Sampling Theory and Applications (Sampta), 2013.
123. L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, pages 704–711. IEEE, 2010.
124. K. Lee and Y. Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. Information Theory, IEEE Transactions on, 56(9):4402–4416, 2010.
125. A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. Journal of mathematical imaging and vision, 48(2):235–265, 2014.
126. Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055, 2010.
127. S. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. Mathematical Programming Computation, 3(3):165–218, 2011.
128. J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.
129. Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent matrix completion. In Proceedings of The 31st International Conference on Machine Learning, pages 674–682, 2014.

130. A. Yurtsever, Q. Tran-Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In Advances in Neural Information Processing Systems 28, pages 3132–3140. 2015.
131. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386–408, 1958.
132. Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley - benchmarking deep learning optimizers. CoRR, abs/2007.01547, 2020.
133. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12(null):2121–2159, jul 2011.
134. Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc' aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc Le, and Andrew Ng. Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.
135. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.