1 Generalizing Quadratics – Smooth and PŁ Optimization

At this point, we have seen how gradient descent behaves on quadratic functions when we have a good handle on how close it looks like to the identity. But, a useful question to ask is – did we deeply use the fact that the function was a quadratic? Can we get a similar result by assuming something weaker?

For instance, based on the above, it seems plausible that a function that always looks close enough to a quadratic at each iterate will make GD behave similarly to the exactly quadratic case. In this section, we will see that this is indeed the case.

Let us first formalize what we mean by a function looking close enough to a quadratic at each iterate.

Definition 1 (Nice function (we couldn't think of a better name...))**.** *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ denote some sequence of points in $\mathbb{R}^d$. We say that f is a "nice function" with respect to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ and a reference $\boldsymbol{x}^\star$ if $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}_i)$ for all $i \in \{1, \ldots, T\}$, and if for all $i \in \{1, \ldots, T\}$, we have*

$$f(\boldsymbol{x}_{i+1}) \leq f(\boldsymbol{x}_i) + \langle \nabla f(\boldsymbol{x}_i), \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \rangle + \frac{\beta}{2} \cdot \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|_2^2 \qquad \textit{quadratic upper bound}$$

$$f(\boldsymbol{x}_i) \leq f(\boldsymbol{x}^\star) + \frac{1}{2\alpha} \|\nabla f(\boldsymbol{x})\|_2^2 \qquad \textit{suboptimality vs gradient}$$

The second condition is a bit less natural, but we will see how this relates to things we have seen previously.

As a sanity check, we will see that quadratics are nice functions for any sequence of points in $\mathbb{R}^d$. However, Definition 1 is a lot more general, as we might only care about f behaving well on a particular sequence of points, or a particular subspace, etc.

For example, when f is a "nice function" with respect to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ generated by running gradient descent, we have:

Theorem 1. *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ be generated by running gradient descent on f with fixed step size η. If f is a nice function with respect to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ with reference $\boldsymbol{x}^\star$, then we have for all i that*

$$f(\boldsymbol{x}_{i+1}) - f(\boldsymbol{x}^\star) \leq (f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star)) \left(1 - 2\alpha \left(\eta - \frac{\beta\eta^2}{2}\right)\right).$$

In particular, if we choose $\eta = 1/\beta$, then we improve our function distance from the reference $\boldsymbol{x}^\star$ significantly in each round.

$$f(\boldsymbol{x}_{i+1}) - f(\boldsymbol{x}^\star) \leq (f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star)) \left(1 - \frac{\alpha}{\beta}\right).$$

Proof of Theorem 1. Each difference has a convenient form – we write this below.

$$\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = -\eta \nabla f(\boldsymbol{x}_i)$$

Thus, we get bounds on $f(\boldsymbol{x}_{i+1})$, which follow from plugging into the upper bound part of Definition 1.

$$f(\boldsymbol{x}_{i+1}) \leq f(\boldsymbol{x}_i) - \eta \|\nabla f(\boldsymbol{x}_i)\|_2^2 + \frac{\beta\eta^2}{2} \|\nabla f(\boldsymbol{x}_i)\|_2^2 = f(\boldsymbol{x}_i) - \|\nabla f(\boldsymbol{x}_i)\|_2^2 \left(\eta - \frac{\beta\eta^2}{2}\right) \qquad (1)$$

Next, we use

$$\|\nabla f(\boldsymbol{x}_i)\|_2^2 \geq 2\alpha \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star) \right). \tag{2}$$

Let $\delta_i := f(\boldsymbol{x}_i) - f(\boldsymbol{x}^\star)$, subtract $f(\boldsymbol{x}^\star)$ from both sides of (1), plug in (2), and we get

$$\delta_{i+1} \leq \delta_i - 2\alpha\delta_i \left(\eta - \frac{\beta\eta^2}{2} \right) = \delta_i \left(1 - 2\alpha \left(\eta - \frac{\beta\eta^2}{2} \right) \right).$$

Now, let us optimize over the step size η. Clearly, it is enough to maximize the below function over η.

$$2\alpha\eta - \alpha\beta\eta^2$$

It is easy to check that the best value of $\eta = (2\alpha)/(2\alpha\beta) = 1/\beta$. Plugging this all the way through gives us

$$\delta_{i+1} \leq \delta_i \left(1 - \frac{\alpha}{\beta} \right),$$

thereby completing the proof of Theorem 1. □

From this, a quick calculation reveals that to get ε-close in function value to the reference $\boldsymbol{x}^\star$, it is enough to run about $\delta_0 \cdot \beta/\alpha \cdot \ln(1/\varepsilon)$ iterations of gradient descent if f was "nice" with respect to the iterates of gradient descent. Note that we did not explicitly assume that f is convex in the above analysis.

One problem with this is that it does not immediately seem that useful – we are left with showing that the iterates of gradient descent satisfy Definition 1. This might depend on several things, including our initialization, what the gradients look like at various parts of f, etc. For starters, for simplicity, there are global conditions we can impose on f that will imply these for free.

We first handle the quadratic upper bound assumption. We give a global condition that will ensure that all the iterates of gradient satisfy it, regardless of how we initialize.

Definition 2 (Smooth function)**.** *We say f is β-smooth if:*

$$\textit{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d: \quad |f(\boldsymbol{x}) - f(\boldsymbol{y}) - \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle| \leq \frac{\beta}{2} \cdot \|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

In fact, Definition 2 implies the quadratic upper bound condition imposed by Definition 1 for *any* sequence of points, not necessarily just those generated by gradient descent.

We now discuss the suboptimality vs gradient condition. Definition 3 presents a natural condition under which we get what we need for free.

Definition 3 (Polyak-Łojasiewicz function)**.** *We say that f is α-PŁ if for all global minimizers $\boldsymbol{x}^\star$, we have:*

$$\textit{for all } \boldsymbol{x} \in \mathbb{R}^d: \quad f(\boldsymbol{x}) \leq f(\boldsymbol{x}^\star) + \frac{1}{2\alpha} \|\nabla f(\boldsymbol{x})\|_2^2.$$

2 Smooth Convex Optimization

Next, we will see a quick reduction to a more general setting – one where we do not necessarily have all of Definition 1.

Theorem 2. *Suppose that f is convex and β-smooth (Definition 2). Additionally, suppose we are given $\boldsymbol{x}_0$ such that for all $\boldsymbol{x}$ with $f(\boldsymbol{x}) \le f(\boldsymbol{x}_0)$, we have $\|\boldsymbol{x}_0 - \boldsymbol{x}\|_2 \le B$ (this is called bounded sub-level set).*

Then, after T steps of gradient descent, for some universal constant C, we have

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \le \left(1 - C\left(\frac{\varepsilon/B^2}{\varepsilon/B^2 + \beta}\right)\right)^T (f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)).$$

In other words, to reach an ε-suboptimal point, it is sufficient to run gradient descent for

$$T = O\left(\frac{B^2}{\varepsilon} \cdot \log\left(\frac{f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)}{\varepsilon}\right)\right)$$

iterations.

Proof of Theorem 2. The proof we present is quite easy but is not optimal in terms of $\log(1/\varepsilon)$ factors. This does not really matter here though.

Consider the auxiliary function

$$g(\boldsymbol{x}) := f(\boldsymbol{x}) + \frac{\varepsilon}{2B^2}\|\boldsymbol{x}_0 - \boldsymbol{x}\|_2^2.$$

Let $\boldsymbol{y}^\star$ be the minimizer for g. Assume that $f(\boldsymbol{y}^\star) \le f(\boldsymbol{x}_0)$ so that we have $\|\boldsymbol{x}_0 - \boldsymbol{y}^\star\|_2 \le B$ (if not, then we are done immediately – why?). Also, consider some approximate minimizer for g. We will try to show that $f(\boldsymbol{y})$ is close to $f(\boldsymbol{x}^\star)$.

For any $\boldsymbol{y}$ that is $\varepsilon/2$-suboptimal with respect to g, we must have

$$f(\boldsymbol{y}) \le g(\boldsymbol{y}) \le \left(f(\boldsymbol{y}^\star) + \frac{\varepsilon}{2B^2}\|\boldsymbol{x}_0 - \boldsymbol{y}^\star\|_2^2\right) + \frac{\varepsilon}{2} \le f(\boldsymbol{y}^\star) + \frac{\varepsilon}{2B^2} \cdot B^2 + \frac{\varepsilon}{2} \le f(\boldsymbol{y}^\star) + \varepsilon,$$

and

$$f(\boldsymbol{y}^\star) \le f(\boldsymbol{y}^\star) + \frac{\varepsilon}{2B^2}\|\boldsymbol{x}_0 - \boldsymbol{y}^\star\|_2^2 \le f(\boldsymbol{x}^\star) + \frac{\varepsilon}{2B^2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2 \le f(\boldsymbol{x}^\star) + \frac{\varepsilon}{2}.$$

This means that

$$f(\boldsymbol{y}) \le f(\boldsymbol{x}^\star) + \frac{3\varepsilon}{2},$$

so rescaling ε means that we can output $\boldsymbol{y}$ obtained from minimizing g as our approximate minimizer for f.

Finally, we have to count the number of iterations we need to approximately minimize g. One can check that g is α-PŁ (Definition 3) with $\alpha =$ and is $\beta + \alpha$-smooth (Definition 2). By Theorem 1, we get (the constants here are probably wrong, but it does not really matter)

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \le \left(1 - \frac{\varepsilon/B^2}{\varepsilon/B^2 + \beta}\right)^T (f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)),$$

so solving for T gives

$$T = O\left(\frac{B^2}{\varepsilon} \cdot \log\left(\frac{f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)}{\varepsilon}\right)\right).$$

This completes the proof of Theorem 2. □

3 Lipschitz and Bounded Convex Optimization

At this point, we have enough tools to relax even the smoothness assumption and still get some understanding of the convergence rate of gradient descent.

First, remember that we did not strictly need our function to be differentiable in order for it to be convex (e.g. remember $|x|$). However, what we did need is that every tangent plane lies under the function. This gives us the following alternative definition for a convex function.

Definition 4 (Subgradient definition of convex function)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$. A function $f\colon \mathcal{X} \to \mathbb{R}$ is convex if for every $\boldsymbol{x} \in \mathcal{X}$, there exists a normal vector $\nabla f(\boldsymbol{x})$ such that for all $\boldsymbol{y} \in \mathcal{X}$ we have*

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

The set of all valid choices of $\nabla f(\boldsymbol{x})$ is called the subgradient of f at $\boldsymbol{x}$ and is written as $\partial f(\boldsymbol{x})$.

Many useful convex functions in practice are not differentiable. But since their subgradients exist, we would like to use this to perform optimization. Below we give some warm-up examples to get used to this way of thinking (they will also be useful in a moment for the programming problems). They are probably easiest solved by drawing a picture in 2 or 3 dimensions and extending that intuition.

Problem 1. *Write down the list of all valid subgradients for $|x|$. You will probably have to use a piecewise function.*

Problem 2. *Extend the above to give the list of all valid subgradients for $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{d} |\boldsymbol{x}_i|$.*

Problem 3. *Last time, we discussed some intuition about what happens at the minimum of a convex function – essentially, 0 was a valid tangent plane.*

Use this to show that based on Definition 4, if $\nabla f(\boldsymbol{x}) = 0$ is a valid subgradient, then $f(\boldsymbol{x}) = f(\boldsymbol{x}^\star)$.

Problem 4. *Consider the function $f(\boldsymbol{x}) = \max_{1 \leq i \leq n} \{f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\}$ where all the functions $f_i(\boldsymbol{x})$ are convex. Prove that f as defined is convex. What is the subgradient of f in terms of the subgradients of f_i?*

One example of a problem we would like to solve is called LASSO. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be such that $n \ll d$. Let the rows of $\mathbf{A}$ be denoted as $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ (so that $\boldsymbol{a}_i \in \mathbb{R}^d$). Let $\boldsymbol{b} \in \mathbb{R}^n$. We are given several measurements of an unknown weight vector $\boldsymbol{w}$ of the form $\langle \boldsymbol{a}_i, \boldsymbol{w} \rangle = \boldsymbol{b}_i$. We want to find $\boldsymbol{w}$ that explains the data.

However, since $n \ll d$, there are many different $\boldsymbol{w}$ that will exactly fit the data. How do we know which one to return?

In practice, an attractive assumption is that most of the features of $\boldsymbol{w}$ do not actually matter. If we knew which features mattered, we could restrict our attention to those and then solve a lower dimensional linear system. This may return a unique solution.

The problem with this is that we do not know which features to select. So – can we automate the feature selection and system solving simultaneously?

One popular approach in practice is to solve the following problem, called LASSO (least absolute shrinkage and selection operator – don't ask us why it's called that)

$$f_\lambda(\boldsymbol{x}) := \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \,. \tag{3}$$

The objective (3) is parameterized by λ. Intuitively, as λ increases, $\|\boldsymbol{x}^\star\|_1$ decreases. We can think of $\|\boldsymbol{x}^\star\|_1$ as a proxy for the number of nonzero entries of $\boldsymbol{x}^\star$, as evidenced by the following exercise.

Problem 5. *Let*

$$S_k := \left\{\boldsymbol{x} \in \{0,1\}^d : \boldsymbol{x} \textit{ has at most } k \textit{ nonzero entries}\right\}.$$

Show that

$$\mathsf{conv}(S_k) = \left\{\boldsymbol{x} \in \mathbb{R}^d_{\geq 0} \colon \sum_{i=1}^{n} \boldsymbol{x}_i \leq k\right\}. \tag{4}$$

To implement subgradient descent, we need to be able to return subgradients of $f_\lambda(\boldsymbol{x})$.

Problem 6 (Needed for programming part)**.** *Write down a valid subgradient of f_λ. Make subgradients 0 whenever it is valid to do so.*

Although we will not go over it in the class, one can show the following guarantee on subgradient descent.

References