1 Overview

Throughout the course we developed an appreciation for the power of convex optimization. This lecture discusses robustness to errors or equivalently optimization of approximately convex functions.

2 Convex Optimization under Error

Consider the problem of minimizing a convex function. In lecture 9 we introduced the *gradient descent* algorithm and proved that by using approximately

$$\kappa = \frac{\log\left(\frac{f(\mathbf{x}^{(0)})-f(\mathbf{x}^\star)}{\epsilon}\right)}{\log(1-\frac{m}{M})}$$

iterations we get a solution $\mathbf{x}$ s.t. $|f(\mathbf{x}) - f(\mathbf{x}^\star)| \leq \epsilon$ where $\mathbf{x}^\star$ is the optimal solution. The problem of convex optimization is therefore solvable in the sense that there are algorithms which make a reasonable number (polynomially-many) of iterations and obtain an arbitrarily close solution to optimal[1]. But suppose that instead of the true value of the function, we have access to some erroneous version of it. Would it still be possible to optimize the function efficiently?

The value query model. The computational model we will consider is the *value query* model, also known as *zeroth order*. In this model we are given an oracle to a function $f : \mathbb{R}^n \to \mathbb{R}$ we wish to optimize s.t. an algorithm can query the oracle at any point $\mathbf{x}$ and the oracle returns the value $f(\mathbf{x})$. In particular, we are not given direct access to the gradient. One option is to approximate the gradient to high degree of accuracy and run gradient descent (see problem set), but there are also other algorithms like *simulated annealing* that are efficient and do rely on gradient evaluations.

Erroneous oracles. We can consider two types of erroneous oracles, as defined below.

[1] We must always rely on approximation as the optimal solution may be an irrational number.

Definition. *For a given function $f : [0,1]^n \to [0,1]$ we say that:*

- $\widetilde{f} : [0,1]^n \to [0,1]$ *is an* ***absolute*** ϵ*-erroneous oracle if* $\forall \mathbf{x} \in [0,1]^n$ *we have that:*

$$f(\mathbf{x}) - \epsilon \leq \widetilde{f}(\mathbf{x}) \leq f(\mathbf{x}) + \epsilon;$$

- $\widetilde{f} : [0,1]^n \to \mathbb{R}$ *is a* ***relative*** ϵ*-erroneous oracle if* $\forall \mathbf{x} \in [0,1]^n$ *we have that:*

$$(1-\epsilon) f(\mathbf{x}) \leq \widetilde{f}(\mathbf{x}) \leq (1+\epsilon) f(x).$$

Note that we intentionally do not make distributional assumptions about the errors. This is in contrast to *noise*, where the errors are assumed to be random and independently generated from some distribution. In such cases, under reasonable conditions on the distribution, one can obtain arbitrarily good approximations of the true function value by averaging polynomially many points in some ϵ-ball around the point of interest. While distributional i.i.d. assumptions are often reasonable models, it is a strong assumption. From a practical perspective, there are cases in which noise can be correlated, or where the data we use to estimate the function is corrupted in some arbitrary way. Furthermore, since we often optimize over functions that we learn from data, the process of fitting a model to a function may also introduce some bias that does not necessarily vanish, or vanishes at some slow rate. But more generally, it seems like we should morally know the consequences that modest inaccuracies may have, or alternatively the limitation of *approximately convex optimization*.

Approximately convex functions. Notice that an absolute erroneous oracle is an *approximately convex function*. That is, an ϵ-erroneous oracle respects an approximate version of convexity in that for every $\mathbf{x}, \mathbf{y} \in [0,1]^n$ we have that:

$$\begin{aligned}
\widetilde{f}(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) &\leq f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) + \epsilon \\
&\leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}) + \epsilon \\
&\leq \lambda \left(\widetilde{f}(\mathbf{x}) + \epsilon \right) + (1-\lambda) \left(\widetilde{f}(\mathbf{y}) + \epsilon \right) + \epsilon \\
&= \lambda \widetilde{f}(\mathbf{x}) + (1-\lambda) \widetilde{f}(\mathbf{y}) + 3\epsilon
\end{aligned}$$

So, in this model, instead of seeking to optimize a convex function, we seek to optimize a function that is only 3ϵ away from being convex. If $\epsilon > 0$ is small (e.g. ϵ converges to zero as n grows large), can we design algorithms that approximately minimize the convex function?

Approximation algorithms The quality of an algorithm is typically measured through two parameters: running time and approximation ratio. We seek algorithms whose running time is polynomial in n and the desired degree of accuracy, and at the same time obtain a good approximation ratio (i.e. have good accuracy).

Definition. *For a minimization problem, given a nonnegative objective function f and a polytope $\mathcal{P}$ we will say that:*

- *An algorithm provides a multiplicative α-approximation ($\alpha > 1$) if it finds a point $\bar{\mathbf{x}} \in \mathcal{P}$ s.t. $f(\bar{\mathbf{x}}) \leq \alpha \min_{\mathbf{x}\in\mathcal{P}} f(\mathbf{x})$. For a maximization problem, an algorithm provides an α-approximation ($\alpha < 1$) if it finds a point $\bar{\mathbf{x}}$ s.t. $f(\bar{\mathbf{x}}) \geq \alpha \max_{\mathbf{x}\in\mathcal{P}} f(\mathbf{x})$.*
- *For absolute erroneous oracles, given an objective function f and a polytope $\mathcal{P}$ we will aim to find a point $\bar{\mathbf{x}} \in \mathcal{P}$ which is within an additive error of δ from the optimum, with δ as small as possible. That is, for a $\delta > 0$ we aim to find a point $\bar{\mathbf{x}}$ s.t. $|f(\bar{\mathbf{x}}) - \min_{\mathbf{x}} f(\mathbf{x})| < \delta$ in the case of minimization.*

Benign cases. In the special case of a linear function $f(\mathbf{x}) = \mathbf{c}^\intercal\mathbf{x}$, for some $\mathbf{c} \in \mathbb{R}^n$, a relative ϵ-error has little effect on the optimization. By querying $f(\mathbf{e}_i)$, for every $i \in [n]$ we can extract $\widetilde{c}_i \in [(1-\epsilon)c_i, (1+\epsilon)c_i]$ and then optimize over $f'(\mathbf{x}) = \widetilde{\mathbf{c}}^\intercal\mathbf{x}$. This results in a $(1 \pm \epsilon)$-multiplicative approximation. Alternatively, if the erroneous oracle $\widetilde{f}$ happens to be a convex function, optimizing $\widetilde{f}(\mathbf{x})$ directly retains desirable optimization guarantees, up to either additive and multiplicative errors. We are therefore interested in scenarios where the error does not necessarily have nice properties.

Example: Gradient descent fails with error. For a simple example, consider the function illustrated in Figure 1. The figure illustrates a convex function (depicted in blue) and an erroneous version of it (dotted red), s.t. on every point, the oracle is at most some additive $\epsilon > 0$ away from the true function value (the ϵ margins of the function are depicted in grey). If we assume that a gradient descent algorithm is given access to the erroneous version (dotted red) instead of the true function (blue), the algorithm will be trapped in a local minimum that can be arbitrarily far from the true minimum. But the fact that a naive gradient descent algorithm fails does not necessarily mean that there isn't an algorithm that can overcome small errors. This narrates the main question in this paper.

Is convex optimization robust to error?

The main result in this lecture is a spoiler. We will show stark information-theoretic lower bounds for both relative and absolute ϵ-erroneous oracles, for any constant and even sub-constant $\epsilon > 0$. To do so, we will use a *probabilistic construction*. We first give a review of basic facts from probability theory, and then prove the impossibility result.

3 Probability Review

Probabilistic constructions. A probabilistic construction is general term for cases in which instead of giving an explicit example we use describe a distribution over examples. In our case we wish to show an impossibility result for convex optimization under error. More specifically, we want to show that for any algorithm, there exists an instance (a function) s.t. given an erroneous oracle, the algorithm will need to make a large (exponential) number of queries to obtain some reasonable guarantee. To show this, instead of explicitly giving a "bad" function for every algorithm, we will

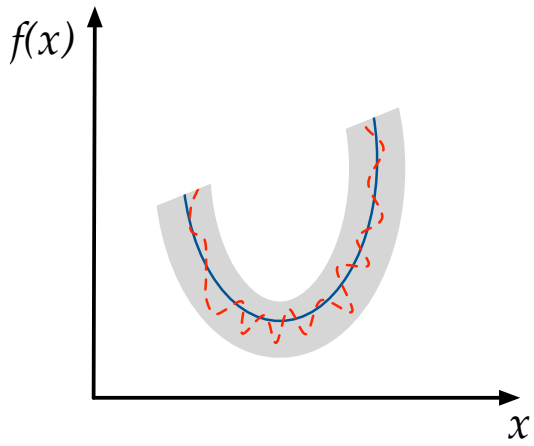

Figure 1: An illustration of an erroneous oracle to a convex function that fools a gradient descent algorithm.

describe a distribution over functions and use this to argue that there exists a function that is "bad" for every algorithm. This statement is a bit abstract at this stage, but in the next section we will see exactly how to use this principle in a meaningful way.

Union bound. Recall that the union bound says that for any finite or countable set of events, the probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events. Formally, for a countable set of events A_1, A_2, A_3 we have that:

$$\mathbb{P}[\cup_i A_i] \leq \sum_i \mathbb{P}[A_i]$$

Chernoff bounds. An important tool we will use is the Chernoff bounds. We note that while typically stated for independent random variables $X_1, \ldots, X_m$, Chernoff bounds also hold for *negatively associated* random variables.

Definition. *Random variables $X_1, \ldots, X_n$ are negatively associated, if for every $I \subseteq [n]$ and every non-decreasing $f : \mathbb{R}^I \to \mathbb{R}, g : \mathbb{R}^{\bar{I}} \to \mathbb{R}$,*

$$\mathbb{E}[f(X_i, i \in I)g(X_j, j \in \bar{I})] \leq \mathbb{E}[f(X_i, i \in I)]\,\mathbb{E}[g(X_j, j \in \bar{I})].$$

Claim 1. *Let $X_1, \ldots, X_n$ be negatively associated random variables that take values in $[0, 1]$ and $\mu = \mathbb{E}[\sum_{i=1}^n X_i]$. Then, for any $\delta \in [0, 1]$ we have that:*

$$\mathbb{P}[\sum_{i=1}^n X_i > (1+\delta)\mu] \leq e^{-\delta^2\mu/3},$$

$$\mathbb{P}[\sum_{i=1}^n X_i < (1-\delta)\mu] \leq e^{-\delta^2\mu/2}.$$

We apply this to random variables that are formed by selecting a random subset of a fixed size. In particular, we use the following.

Claim 2. *Let $x_1, \ldots, x_n \geq 0$ be fixed. For $1 \leq k \leq n$, let R be a uniformly random subset of k elements out of $[n]$. Let $X_i = x_i$ if $i \in R$ and $X_i = 0$ otherwise. Then $X_1, \ldots, X_n$ are negatively associated.*

4 An Impossibility Result for Convex Optimization under Error

We will consider convex minimization over $[0,1]^n$ so that we can calibrate the error level ϵ. In this setting, we show that errors as small as $n^{-(1-\delta)/2}$ prevent us from optimizing within a constant additive error of nearly $1/2$. Notice that a dummy algorithm that regardless of the function being optimized always returns $1/2$ will always be $\pm 1/2$ from optimal.

Theorem 3. *Let $\delta > 0$ be a constant. There are instances of a convex function $f : [0,1]^n \to [0,1]$ accessible through an absolute $n^{-(1-\delta)/2}$-erroneous oracle, such that a (possibly randomized) algorithm that makes $e^{O(n^\delta)}$ queries cannot find a solution of value better than within additive $1/2 - o(1)$ of the optimum with probability more than $e^{-\Omega(n^\delta)}$.*

Proof. Let $\epsilon = n^{-(1-\delta)/2}$; we can assume that $\epsilon < \frac{1}{2}$, otherwise n is constant and the statement is trivial. We will construct an ϵ-erroneous oracle (both in the relative and absolute sense) for a convex function $f : [0,1]^n \to [0,1]$. Consider a partition of $[n]$ into two subsets A, B of size $|A| = |B| = n/2$ (which will be eventually chosen randomly). We define the following function:

- $f(\mathbf{x}) = \frac{1}{2} + \frac{1}{n}(\sum_{i \in A} x_i - \sum_{j \in B} x_j)$.

This is a convex (in fact linear) function. Next, we define the following modification of f, which could be the function returned by an ϵ-erroneous oracle.

- If $|\sum_{i \in A} x_i - \sum_{j \in B} x_j| > \frac{1}{2}\epsilon n$, then $\tilde{f}(\mathbf{x}) = f(\mathbf{x}) = \frac{1}{2} + \frac{1}{n}(\sum_{i \in A} x_i - \sum_{j \in B} x_j)$.
- If $|\sum_{i \in A} x_i - \sum_{j \in B} x_j| \leq \frac{1}{2}\epsilon n$, then $\tilde{f}(\mathbf{x}) = \frac{1}{2}$.

Note that $f(\mathbf{x})$ and $\tilde{f}(\mathbf{x})$ differ only in the region where $|\sum_{i \in A} x_i - \sum_{j \in B} x_j| \leq \frac{1}{2}\epsilon n$. In particular, the value of $f(\mathbf{x})$ in this region is within $[\frac{1-\epsilon}{2}, \frac{1+\epsilon}{2}]$, while $\tilde{f}(\mathbf{x}) = \frac{1}{2}$, so an ϵ-erroneous oracle for $f(\mathbf{x})$ (both in the relative and absolute sense) could very well return $\tilde{f}(\mathbf{x})$ instead.

Now assume that (A, B) is a random partition, unknown to the algorithm. We argue that with high probability, a fixed query $\mathbf{x}$ issued by the algorithm will have the property that $|\sum_{i \in A} x_i - \sum_{j \in B} x_j| \leq \frac{1}{2}\epsilon n$. More precisely, since (A, B) is chosen at random subject to $|A| = |B| = n/2$, we have that $\sum_{i \in A} x_i$ is a sum of negatively associated random variables in $[0,1]$ (by Claim 2). The expectation of this quantity is $\mu = \mathbb{E}[\sum_{i \in A} x_i] = \frac{1}{2}\sum_{i=1}^n x_i \leq \frac{1}{2}n$. By Claim 1, we have

$$\mathbb{P}[\sum_{i \in A} x_i > \mu + \frac{1}{4}\epsilon n] = \mathbb{P}[\sum_{i \in A} x_i > (1 + \frac{n}{4\mu}\epsilon)\mu] < e^{-(n\epsilon/(4\mu))^2 \mu/3} \leq e^{-\epsilon^2 n/24}.$$

Since $\frac{1}{2}\sum_{i\in A} x_i + \frac{1}{2}\sum_{i\in B} x_i = \frac{1}{2}\sum_{i=1}^{n} x_i = \mu$, we get

$$\mathbb{P}[\sum_{i\in A} x_i - \sum_{i\in B} x_i > \frac{1}{2}\epsilon n] = \mathbb{P}[\sum_{i\in A} x_i - \mu > \frac{1}{4}\epsilon n] < e^{-\epsilon^2 n/24}.$$

By symmetry,

$$\mathbb{P}[|\sum_{i\in B} x_i - \sum_{j\in A} x_j| > \frac{1}{2}\epsilon n] < 2e^{-\epsilon^2 n/24}.$$

We emphasize that this holds for a *fixed query* $\mathbf{x}$.

Recall that we assumed the algorithm to be deterministic. Hence, as long as its queries satisfy the property above, the answers will be $\tilde{f}(\mathbf{x}) = 1/2$, and the algorithm will follow the same path of computation, no matter what the choice of (A, B) is. (Effectively we will not learn anything about A and B.) Considering the sequence of queries on this computation path, if the number of queries is t then with probability at least $1 - 2te^{-\epsilon^2 n/24}$ the queries will indeed fall in the region where $\tilde{f}(\mathbf{x}) = 1/2$ and the algorithm will follow this path. If $t \leq e^{\epsilon^2 n/48}$, this happens with probability at least $1 - 2e^{-\epsilon^2 n/48}$. In this case, all the points queried by the algorithm as well as the returned solution $\mathbf{x}_{out}$ (by the same argument) satisfies $\tilde{f}(\mathbf{x}_{out}) = 1/2$, and hence $f(\mathbf{x}_{out}) \geq \frac{1-\epsilon}{2}$. In contrast, the actual optimum is $f(1_B) = 0$. Recall that $\epsilon = n^{-(1-\delta)/2}$; hence, $f(\mathbf{x}_{out}) \geq \frac{1}{2}(1 - n^{-(1-\delta)/2})$ and the bounds on the number of queries and probability of success are as in the statement of the theorem.

Finally, consider a randomized algorithm. Denote by $(R_1, R_2, \ldots, ...)$ the random variables used by the algorithm in its decisions. We can condition on a fixed choice of $(R_1 = r_1, R_2 = r_2, \ldots)$ which makes the algorithm deterministic. By our proof, the algorithm conditioned on this choice cannot succeed with probability more than $e^{-\Omega(n^\delta)}$. Since this is true for each particular choice of $(r_1, r_2, \ldots)$, by averaging it is also true for a random choice of $(R_1, R_2, \ldots)$. Hence, we obtain the same result for randomized algorithms as well. □

5 Discussion and Further Reading

This lecture is based on a recent paper [1]. In addition to the result we presented here, one can show similar lower bounds for for maximizing a concave function $f : [0, 1]^n \to [0, 1]$ over a polytope, maximizing a concave function $f : [0, 1]^n \to [0, 1]$ over $[0, 1]^n$, and maximizing a concave function over a partition polytope.

References

[1] Yaron Singer and Jan Vondrak. Information-theoretic lower bounds for convex optimization with erroneous oracles. In *Advances in Neural Information Processing Systems 28*, pages 3204–3212. 2015.