CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

Feburary 3, 2015 Notes: Ashia Wilson and Benjamin Recht

Lecture 5: Lower Bounds

Our previous lectures have looked at the iteration complexity of first order methods. We saw that we could efficiently find approximate minimizers of convex functions in polynomial time, and provided guarantees on the rate of convergence in terms of parameters of the function to be minimized. These bounds held *for any* function in the class. So, for example, we discussed how the gradient method could be used to optimize a convex function satisfying

$$mI \preceq \nabla^2 f(x) \preceq LI$$

in a number of iterations proportional to L/m. Nesterov's method scales proportionally to $\sqrt{L/m}$. Is this the best we can do?

In this lecture we discuss lower bounds for a variety of function classes. In some sense, we show that Nesterov is "optimal" for algorithms that only evaluate function values and gradients of convex functions. In sharp contrast, we show that for non-convex functions, even verifying local optimality is intractable. So though we can find points with small gradients relatively quickly, having a small gradient tells us nothing about the local optimality of a nonconvex function. We show that the situation is only worse for Lipschitz continuous functions (i.e., the function is Lipschitz, but the gradient may not be). In this case, even finding a stationary point is intractable.

Before we proceed, I want to emphasize that all lower bounds are suspect. We are establishing the existence of one function on which restricted algorithms must take a long time to optimize. But these counter-examples are very fragile. One must wonder does this show a deficiency in modeling my function class or in my algorithm family? All lower-bounds must be taken with a grain of salt. We will emphasize the short comings of these particular bounds in our discussion.

1 Lower bounds for convex functions

Let's first consider the class of convex functions. We would like to minimize a convex function f. Assume throughout that f has L-Lipschitz gradients and f is strongly convex function with parameter m. We will also examine the case where $m = 0$.

We will restrict our attention to a particular class of algorithms. Our algorithms will be able to evaluate the value of f and its gradient at any point in $\mathbb{R}^d$. We will allow ourselves to initialize our search at any $x^{(0)} \in \mathbb{R}^d$. Note that because the set of convex functions is translation invariant, we can always assume that $x^{(0)} = 0$.

We additionally make the very stringent assumption

- **Restriction** *The iterate $x^{(k)}$ must be a linear combination of*

$$\{x^{(1)}, \ldots, x^{(k-1)}\} \qquad \text{and} \qquad \{\nabla f(x^{(1)}), \ldots, \nabla f(x^{(k-1)})\}\,.$$

That is, the next iterate is in the span of all the previous iterates and all the previous gradients.

Our goal will be to minimize the number of function evaluations and gradient calls to achieve

$$||x^{(N)} - x_\star|| < \epsilon$$

Interestingly, the worst-case instance we will construct for this case is a quadratic function. So even though the class of strongly convex functions is considerably richer than the space of quadratics, our bad case will be a quadratic.

Methods obeying our restriction are often called *Krylov methods* when f is quadratic. So what would our algorithm be able to do when applied to the function

$$f(x) = x^T Ax + b^T x$$

Recall from last lecture: the optimal solution is given by $Ax = b$. Let's expand out the iterates, starting at $x^{(0)} = 0$.

$$\begin{aligned}
x^{(0)} &= 0 \\
x^{(1)} &= x^{(0)} + t_1(Ax^{(0)} - b) = t_1 b \\
x^{(2)} &= t_1 b + t_2(Ax^{(1)} - b) = t_{21}Ab + t_{20}b \text{ (for some } t_{21} \text{ and } t_{20}) \\
x^{(3)} &= t_{32}A^2 b + t_{31}Ab + t_{30}b
\end{aligned}$$

Repeating this calculation we see that

$$x^{(k)} = \sum_{i=0}^{k-1} t_{k-i} A^i b$$

and hence x_k is in the span of $\{b, Ab, A^2 b, \ldots, A^{k-1}b\}$. One can now probably devise how to construct a bad instance. We need to find a positive definite matrix A and a vector b such that the distance of $A^{-1}b$ is as far away from the span of $\{b, Ab, A^2 b, \ldots, A^{k-1}b\}$ as possible.

A particular construction, due to Nesterov is

$$\begin{aligned}
f(x) &= (1/2)x^T Ax - b^T x \\
A &= \alpha A_0 + \beta I \\
b &= \alpha e_1
\end{aligned}$$

where

$$(A_0)_{ij} = \begin{cases} 2 & i = j \\ -1 & |i - j| = 1 \\ 0 & o.w. \end{cases}$$

Thiat is,

$$A_0 = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & 0 & \ldots & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \end{bmatrix}$$

$A_0 \succeq 0$ because it is diagonally dominant (in fact it is p.d., but just barely). We can also see this by noticing it is a sum of squares:

$$f(x) = \tfrac{1}{2}(\alpha+\beta)(x_1^2 + x_d^2) + (\alpha/2)\sum_{k=1}^{d}(x_k - x_{k+1})^2 + \frac{\beta}{2}\sum_{k=2}^{d=1} x_k^2 - \alpha x_1$$

The Lipschitz constant of the quadratic is equal to the maximum eigenvalue (i.e. the operator norm) of A. We begin by looking at A_0:

$$\begin{aligned} x^T A_0 x &= x_1^2 + \sum_{i=1}^{n}(x_i - x_{i+1})^2 + x_n^2 \\ &\le x_1^2 + \sum_{i=1}^{n}(x_i^2 + x_{i+1}^2) + x_n^2 \\ &\le 4||x||^2 \end{aligned}$$

Thus $||A_0|| \le 4$. If we set $\alpha = \frac{L-m}{4}$ and $\beta = m$, we see that f has L-Lipschitz gradients and strong convexity parameter m.

For the remainder of the proof, we make the simplifying assumption that d is infinite. This will let us solve for the optimal solution in closed form. This argument can be truncated to finite d with some messy algebra. We leave this construction to the interested reader.

Writing out $Ax = b$ in coordinates, we see that the optimal x satisfies:

$$(2 + \frac{4}{\kappa - 1})x_1 - x_2 = 1 \tag{1}$$

$$-x^{(j+1)} + 2\left(\frac{\kappa+1}{\kappa-1}\right)x_j - x_{j+1} = 0 \tag{2}$$

Where $\kappa = \frac{L}{m}$ is the condition number L/m. This system has a simple ansatz. Indeed, the optimal x has coordinates $x_k = u^k$,where u^k satisfies

$$\begin{aligned} &-u^{k-1} + 2\left(\frac{\kappa+1}{\kappa-1}\right)u^k - u^{k+1} \\ &-u^{k-1}\left[1 - \frac{2(\kappa+1)}{\kappa-1}u + u^2\right] = 0 \end{aligned}$$

Solving the quadratic equation yields the solution

$$u = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right).$$

And the optimal solution is the sequence

$$x_\star = [u^k] = \left[\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k\right].$$

Now after k iterations

$$x^{(k)} \in span\Big\{\left[\frac{L-m}{4}A_0 + mI\right]^i e_1\Big\}_{i=0,\ldots,k-1}$$

In particular, we must have that $x_j^{(k)} = 0$ for all $j > k$. Thus, our error can be no better than the norm of the tail. That is, the sum over the remaining coordinates.

$$\begin{aligned}||x^{(k)} - x_\star|| &\geq \sum_{j=k+1}^{\infty} u^{2j} \\ &= \frac{u^{2(k+1)}}{1-u^2} \\ &= u^{2(k+1)}||x_\star||^2 \\ &= u^{2(k+1)}||x_\star - x^{(0)}||^2\end{aligned}$$

Here, the first equality sums the geometric series. The second follows by computing the ℓ_2 norm of $x_\star$. The final equality follows because we started at $x^{(0)} = 0$.

We can also bound the function value

$$\begin{aligned}f(x^{(k)}) - f(x_\star) &\geq \frac{m}{2}||x^{(k)} - x_\star|| \\ &\geq \frac{m}{2}u^{2(k+1)}||x_\star - x^{(0)}||^2 \\ &= \frac{m}{2}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2k+2}||x_\star - x^{(0)}||^2\end{aligned}$$

This calculation confirms that no Krylov method can find an ϵ approximate solution in less than $\Theta(\sqrt{\kappa}\log(1/\epsilon))$ iterations.

Thus, we can say that Nesterov's algorithm is an "optimal method." Of course, there are many caveats here. First, note that this only holds for small iteration counts. If we could run for d steps, then conjugate gradient is a Krylov method that solves the problem exactly. Moreover, we can solve the system of equations $Ax = b$ in closed form in $O(d)$ time whenever A is tridiagonal. Even more importantly, it is not at all clear that Nesterov is the best algorithm for *all* strongly convex functions. It is optimal for this particular instance, but other algorithms might be more efficient on other functions—even on other quadratics.

1.1 Weakly convex f

A similar quadratic function shows that $1/k^2$ iterations are optimal for weakly-convex f. In this case,

$$f(x) = x^T\frac{L}{8}A_0x - \frac{L}{4}x^Te_1\,.$$

We can write this out in coordinates

$$\begin{cases}2x_1 - x_2 = 1 \\ -x_{k-1} + 2_k - x_k = 0 \\ 2x_d - x_{d-1} = 0\end{cases}$$

Here we will run for $d/2$ iterations (which is also not fair... again, if we ran for d iterations, we would have solved the problem exactly with conjugate gradient). By forward substituting, we get

$$x_\star = [1 - \frac{j}{d+1}]$$

This means that

$$\begin{aligned} f(x_\star) &= \frac{L}{8} x_\star^T A x_\star - x_{opt}^{(1)} \\ &= \frac{L}{8}\left[1 - \frac{1}{d+1} - 2\left[1 - \frac{1}{d+1}\right]\right] \\ &= -\frac{L}{8}\left[1 - \frac{1}{d+1}\right] \end{aligned}$$

Moreover,

$$\begin{aligned} ||x_\star||^2 &= \sum_{i=1}^{d}\left(1 - \frac{i}{d+1}\right)^2 \\ &= d - \frac{2}{d+1}\sum_{i=1}^{d} i + \frac{1}{(d+1)^2}\sum_{i=1}^{d} i^2 \\ &= d - \frac{2(d \cdot d+1)}{(d+1)\cdot 2} + \frac{1}{(d+1)^2}\cdot\frac{d(d+1)(2d+1)}{6} \\ &= d - d + \frac{d(2d+1)}{6(d+1)} \\ &\leq \frac{d}{3} \end{aligned}$$

These calculations let us establish the following

Theorem 1 *For any $1 \leq k \leq \frac{1}{2}(d-1)$, and any $x^{(0)} \in \mathbb{R}^1$, there exists an f with L-Lipschitz gradients such that for any first order method*

$$f(x^{(k)}) - f(x_\star) \geq \frac{3L||x^{(0)} - x_\star||^2}{32(k+1)^2}$$

$$||x^{(k)} - x_\star||^2 \geq \frac{1}{8}||x^{(0)} - x_\star||^2$$

Proof Use

$$f_d := \frac{1}{8}x^T A_0 x - \frac{L}{4}e_1$$

with $d = 2k+1$. Note

$$f_{2k+1}(x_k) = f_k(x_k) \geq f_k(x_{opt}) = \frac{L}{8}\left(-1 + \frac{1}{k+1}\right)$$

Therefore

$$\begin{aligned} f(x_k) - f(x_\star) &\geq \frac{L}{8}\left(-1 + \frac{1}{k+1} + 1 - \frac{1}{2k+2}\right) \\ &= \frac{L}{8}\left(\frac{1}{2k+1}\right) \\ &\geq \frac{3}{8}\frac{1}{(2k+1)(2k+2)}||x_{opt} - x^{(0)}||^2 \\ &\geq \frac{3}{32}\frac{1}{(d+1)^2}||x_{opt} - x^{(0)}||^2 \end{aligned}$$

For the other inequality

$$\begin{aligned}||x_k - x_\star||^2 &\geq \sum_{i=k+1}^{2k+1} \left(1 - \frac{i}{2k+2}\right)^2 \\ &= k + 1 - \frac{1}{k-1} \sum_{i=k+1}^{2k+1} i + \frac{1}{4(k+1)^2} \sum_{i=k+1}^{2k+1} i^2 \\ &\geq \frac{2k^2 + 7k + 6}{24(k+1)} \cdot \frac{3}{2k+1} ||x_{opt} - x^{(0)}||^2 \\ &\geq \frac{1}{8} ||x_{opt} - x^{(0)}||^2\end{aligned}$$

■

2 Lower bounds for smooth functions

Let's now restrict our attention to differentiable, rather than convex functions. We know that we can efficiently find a solution with $\|\nabla f(x)\|^2 \leq \epsilon$ in $O(1/\epsilon^2)$ time. But what about a global minimum?

It turns out that even finding a *local* minimum is intractable. Consider the subset of homogenous quartics:

$$f(x) = \sum_{i,j=1}^{d} Q_{ij} x_i^2 x_j^2$$

where Q is some arbitrary symmetric matrix. If Q is positive definite or nonnegative then 0 is obviously a global minimizer. What about for general Q.

First note that

$$\frac{\partial f}{\partial x_i} = 4Q_{ii} x_i^3 + 2 \sum_{j \neq i} Q_{ij} x_i x_j^2$$

Thus, the gradient is necessarily 0 at $x = 0$. However,

$$\begin{aligned}\frac{\partial^2 f}{\partial x_i^2} &= 12 Q_{ii} x_i + 2 \sum_{j \neq i} Q_{ij} x_j^2 \\ \frac{\partial^2 f}{\partial x_i \partial x_j} &= 4 \sum_{j \neq i} Q_{ij} x_i x_j \qquad \text{for } i \neq j\end{aligned}$$

Meaning the *Hessian* is also zero at zero. So we cannot conclude that x is a local minimum from Hessian information alone.

It turns out that the situation is quite dire, actually:

Theorem 2 (Murty and Kabadi, 1987) *Given an integer square matrix Q, deciding if there exists and u with non-negative entries satisfying $u^T Q u < 0$ is NP complete.*

The proof of this fact follows by a relatively straightforward reduction to the SUBSET-SUM problem. Beginning with the formulation of this problem as an integer program, Murty and Kabadi produce a quadratic function such that 0 is the minimizer over the positive orthant if and only if the subset sum problem is feasible.

Reformulating this problem in our language, note that there is a 1 to 1 correspondence between vectors with nonnegative entries and vectors whose entries are squares. Thus, deciding if there exists and u with non-negative entries satisfying $u^T Qu < 0$ is equivalent to deciding if there exits a vector x such that $f(x) = \sum_{i,j=1}^d Q_{ij} x_i^2 x_j^2 < 0$. Note that by homogeneity, x is a local optimum if and only if there does not exist any x with $f(x) < 0$. Indeed, if there was such an x, then $f(tx)$ would be negative for $t \neq 0$. Thus, deciding if 0 is a local minimum of f is *co-NP complete*.

Now there are two caveats here. First, the hardness depends on P not equalling $co-NP$. Most people think this is true, but it remains a conjecture. Second, just because a problem is NP-hard doesn't mean you shouldn't try to solve it! NP-hard means that there are very difficult instances out there, so one must be careful. But many NP-hard problems have very large sets of instances that are efficiently solvable. One just has to make sure that the problem you care about is in the solvable part of the complexity zoo.

3 Lower bounds for continuous functions

Continuous functions are even more difficult than smooth functions. Consider the class of functions f that are L-Lipschitz. That is,

$$|f(x) - f(y)| \leq L|x - y|$$

for all x and y. These functions are necessarily continuous, but not necessarily differentiable. Thus, all we can do is evaluate function values and hope for the best. It turns out that the best is rather bad:

Theorem 3 *Let $\mathcal{A}$ be any algorithm that can access function values of Lipschitz $f : [0,1]^d \to \mathbb{R}$. Then there is no method that can find a solution with $f(x) - f(x_\star) \leq \epsilon$ in less than $(L/2\epsilon)^d$ function calls.*

This theorem says that no matter what you try to do, you can't optimize a Lipschitz continuous function in time less than exponential in the dimension. This is a rather hard limit, it doesn't rest on algorithmic restrictions nor on resolving the $P = NP$ question. As we'll see in the proof, it's just too easy to hide a needle in the haystack of Lipschitz continuous functions.

Proof Consider our algorithm $\mathcal{A}$ applied first to the 0 function. This function is certainly L-Lipschitz. Let's suppose this returns a sequence of points $\mathcal{P} \subset [0,1]^d$ with $|\mathcal{P}| = N$. By a dimension counting argument, there exists a point $\hat{x} \in [0,1]^d$ such that

$$\mathcal{P} \cap \{x \ : \ \hat{x}_i - N^{1/d}/2 \leq x_i \leq \hat{x}_i + N^{1/d}/2\} = \emptyset$$

Let's make the horrible function

$$f(x) = \min\{0, L\|x - \hat{x}\|_\infty - \epsilon\}$$

This function is clearly Lipschitz continous. Moreover, it is zero outside of the box

$$\{x \ : \ \hat{x}_i - \frac{\epsilon}{L} \leq x_i \leq \hat{x}_i + \frac{\epsilon}{L}\}$$

Thus, our algorithm when applied to this function will return a point with $f(x) = 0$, and hence will not find an ϵ optimal solution if $N \leq (\frac{L}{2\epsilon})^d$. ■