4 Lower complexity bounds for smooth minimization

Is the gradient method optimal? Or is there another algorithm that can achieve faster rate of convergence?

A *first-order* algorithm is one that has access to function values $f(x)$ and gradients $\nabla f(x)$. The complexity of such an algorithm is the number of queries it makes. We will consider in this lecture algorithms that satisfy the following assumption: the k'th iterate/query point x_k of the algorithm satisfies:

$$x_k \in x_0 + \operatorname{span}\{\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_{k-1})\}. \tag{1}$$

Clearly the gradient method satisfies this assumption.

Define $\mathcal{F}_L = \{f : \mathbb{R}^n \to \mathbb{R} \text{ convex with } L\text{-Lipschitz gradient}\}$. We want to understand how well can first-order algorithms behave on functions in $\mathcal{F}_L$.

Theorem 4.1. *Fix $L > 0$ and an integer $k \geq 1$. For any algorithm satisfying (1), there is a function $f \in \mathcal{F}_L$ on $n = 2k+1$ variables such that after k steps of the algorithm*

$$f(x_k) - f^* \geq \frac{3}{32} \frac{L\|x_0 - x^*\|_2^2}{(k+1)^2} \tag{2}$$

and

$$\|x_k - x^*\|_2^2 \geq \frac{1}{8}\|x_0 - x^*\|_2^2. \tag{3}$$

Proof. Let $n = 2k+1$ and consider the function $f : \mathbb{R}^n \to \mathbb{R}$ as follows

$$f(x) = \frac{L}{8}\left(x_n^2 + \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2 + x_1^2 - 2x_1\right). \tag{4}$$

Let also, for $i = 1, \dots, n$ $V_i = \{x \in \mathbb{R}^n : x_{i+1} = \cdots = x_n = 0\}$. Then we have the following properties about f:

(i) $f \in \mathcal{F}_L$

(ii) The minimum of f is attained at $x^* = \left(\frac{n}{n+1}, \dots, \frac{2}{n+1}, \frac{1}{n+1}\right)$ and the optimal value is $f^* = -\frac{L}{8}\frac{n}{n+1}$. More generally the minimum of f on the subspace V_i is $-\frac{L}{8}\frac{i}{i+1}$, attained at the point $\left(\frac{i}{i+1}, \dots, \frac{2}{i+1}, \frac{1}{i+1}, 0, \dots, 0\right) \in V_i$.

(iii) If $x \in V_i$ for $i < n$, then $\nabla f(x) \in V_{i+1}$.

We leave it to the reader to check these properties.

Assume without loss of generality that the first query point of the algorithm is $x_0 = 0$ (if it is not we simply consider the function $\tilde{f}(x) = f(x - x_0)$). By property (iii) of f, and by assumption (1) on the algorithm this means that the k'th query point x_k of the algorithm must belong to V_k. Thus this means that

$$f(x_k) \geq \min_{x \in V_k} f(x) = -\frac{L}{8}\frac{k}{k+1}.$$

Now using the fact that $n = 2k+1$ and $f^* = -\frac{L}{8}\frac{n}{n+1}$ we get

$$f(x_k) - f^* \geq \frac{L}{8}\left(\frac{2k+1}{2k+2} - \frac{k}{k+1}\right) = \frac{L}{8}\frac{1}{2k+2}.$$

Also note that $\|x_0 - x^*\|_2^2 = \|x^*\|_2^2 = \frac{1}{(n+1)^2}\sum_{i=1}^{n-1} i^2 = \frac{n}{n+1}\frac{2n+1}{6} \leq \frac{n+1}{3}$, thus

$$\frac{f(x_k) - f^*}{\|x_0 - x^*\|_2^2} \geq \frac{L}{8}\frac{1}{2k+2}\frac{3}{2k+2} = \frac{3L}{32}\frac{1}{(k+1)^2}$$

as desired.

We now prove (3). Since $x_k = (?, \ldots, ?, 0, \ldots, 0)$ then $x_k - x^* = \left(?, \ldots, ?, -\frac{n-k}{n+1}, \ldots, -\frac{1}{n+1}\right)$ which implies $\|x_k - x^*\|_2^2 \geq \frac{1}{(n+1)^2}\sum_{i=1}^{n-k} i^2$. Now using the fact that $n = 2k+1$ we get $\|x_k - x^*\|_2^2 \geq \frac{1}{24}(2k+3)$. Combining with $\|x_0 - x^*\|_2^2 \leq \frac{2k+2}{3}$ we get $\|x_k - x^*\|_2^2 \geq \frac{1}{8}\|x_0 - x^*\|_2^2$ as desired. □

We proved that the gradient method converges at a rate $C\frac{L\|x_0 - x^*\|_2^2}{k}$, whereas the lower bound we proved is of the form $C'\frac{L\|x_0 - x^*\|_2^2}{k^2}$. If the lower bound is tight, it suggests we might have an algorithm that is faster than the gradient method.