PTML 11: 16/06/2022

TABLE DES MATIÈRES

INTRODUCTION

In previous sessions, we have discussed several upper bounds on the performance of machine learning methods.

For instance, we have seen that for a strongly convex smooth function f defined on $\mathbb{R}^d$ and with real values, the convergence of gradient descent to the minimizer x^* of f can be upper bounded. More formally, we have seen that if $(x^t)_{t\in N}$ is the sequence of iterates, then

$$||x^t - x^*|| = \mathcal{O}(\alpha^t) \tag{1}$$

with $0 \leqslant \alpha < 1$. If f is only convex, and not necessary strongly convex, then we have a weaker upper bound :

$$f(x^t) - f(x^*) = \mathcal{O}(1/t) \tag{2}$$

Having an upper bound guarantees that the algorithm converges **faster** than the bound.

In this session we present the problem of obtaining **lower** bounds on the performance of machine learning methods. This means proving that an algorithm can not be guaranteed to converge too fast to the minimizer of f. We will formally define what this means in section 1.3, as it is not as straightforward as for upper bounds.

There are two kinds of lower bounds : statistical lower bounds and optimization lower bounds, which we will focus on.

1 LOWER BOUND ON THE PERFORMANCE OF GRADIENT-BASED ALGORITHMS

Let $\mathcal{F}$ be the space of functions $f : \mathbb{R}^d \to \mathbb{R}$ that

- are convex and differentiable
- and have a minimizer x_f^*
- are L-smooth, with $L \in \mathbb{R}_+$ (the gradient is L-Lipshitz continuous).

1.1 First-order methods

We consider optimization algorithms that are based on linear combinations of local estimations of the gradient of f. These methods are called **first-order methods.** GD is an example of first-order method. Formally, this means that each iterate x^t verifies

$$x^t \in x^0 + \text{span}\{\nabla f(x^0), \dots, \nabla f(x^{t-1})\} \tag{3}$$

"span" means "espace engendré par".

1.2 Nesterov acceleration (AGD)

Nesterov acceleration (also called accelerated gradient descent (AGD)), is another first-order method, slightly different than GD. It is possible to prove that with Nesterov acceleration, the rate of convergence is $\mathcal{O}(1/t^2)$, instead of $\mathcal{O}(1/t)$.
https://blogs.princeton.edu/imabandit/2013/04/01/acceleratedgradientdescent/

1.3 Minimax lower bound

It is possible to show that AGD is optimal among first order methods. But first, we have to define what this means. This optimality is defined as a minmax problem. Indeed, we are interested in finding the algorithm that has the best worst-case performance. It is necessary to restrict algorithms to a relevant class of algorithms $\mathcal{A}$ (here, first order methods), and the functions to a relevant class of functions $\mathcal{F}$ (for instance the set of convex, L-smooth functions on $\mathbb{R}^d$).

For $k \in \mathbb{N}$, we look for an algorithm $a \in \mathcal{A}$ defined by

$$a = \arg\min_{a \in \mathcal{A}} \max_{f \in \mathcal{F}} \left[\|x_a^k - x^*\| \right] \tag{4}$$

where x_a^k is the iterate returned by algorithm a at iteration k.

1.4 Optimality of Nesterov acceleration

If we manage to show that for any algorithm $a \in \mathcal{A}$, there exists a function $f \in \mathcal{F}$, such that

$$f(x^k) - f(x^*) \geqslant \frac{C}{k^2} \|x^0 - x_f^*\|^2 \tag{5}$$

where C is independent on the function, then this will prove that Nesterov acceleration is optimal in $\mathcal{A}$, as this means that for any algorithm in $\mathcal{A}$, there exists a function, for which the iterates produced by a converge slower than $\frac{C}{k^2}$ to its minimizer.

1.5 Hard function

Without loss of generality, we assume that for all algorithms in $\mathcal{A}$, the initialization is $x^0 = 0 \in \mathbb{R}^d$.

We consider $k \leqslant \frac{d-1}{2}$ and f defined by

$$f(x) = \frac{L}{4} \left(\frac{1}{2} x_1^2 + \frac{1}{2} \sum_{i=1}^{2k} (x_i - x_{i+1})^2 + \frac{1}{2} x_{2k+1}^2 - x_1 \right) \tag{6}$$

Exercice 1 : Show that f is convex.

We admit that f is L-smooth, and that the minimizer of f, x^*, is

$$x_i^* : \begin{cases} 1 - \frac{i}{2k+2} \text{ for } 1 \leqslant i \leqslant 2k+1 \\ 0 \text{ for } i \geqslant 2k+2 \end{cases}$$

and that the minimum value is

$$f^* = \frac{L}{8}\left(\frac{1}{2k+2} - 1\right) \tag{7}$$

We consider another utility function g defined as

$$g(x) = \frac{L}{4}\left(\frac{1}{2}x_1^2 + \frac{1}{2}\sum_{i=1}^{k-1}(x_i - x_{i+1})^2 + \frac{1}{2}x_k^2 - x_1\right) \tag{8}$$

We also admit that the minimum value of g, noted g^*, is

$$g^* = \frac{L}{8}\left(\frac{1}{k+1} - 1\right) \tag{9}$$

Exercice 2 : Compute the gradient of f for any point $x \in \mathbb{R}^d$.

Exercice 3 : Show that for all $l \leqslant d$, the components j with $j \geqslant l+1$ are null for the iterate x_l, for any first-order method that produces iterates with gradients of f (as defined in 3). In particular, this is true for $l = k$ and for the iterate k, we have

$$x_{k+1}^k = x_{k+2}^k = \cdots = x_d^k = 0 \tag{10}$$

Exercice 4 : Show that $f(x^k) = g(x^k)$.

Hence, $f(x^k) \geqslant g^*$ and

$$\frac{f(x^k) - f^*}{\|x^0 - x^*\|^2} \geqslant \frac{g^* - f^*}{\|x^0 - f^*\|^2} \tag{11}$$

Exercice 5 : Show that

$$\|x^*\|^2 \leqslant \frac{2k+2}{3} \tag{12}$$

Exercice 6 : Conclude.

1.6 Discussion

By essence, considering the worst-case performance (the hardest function to optimize) is pessimistic. Some functions optimized wit a first-order method by converge faster than $\mathcal{O}(1/t^2)$. We have seen this behavior for a least-squares setting in a previous session.

RÉFÉRENCES