What is the best output of an algorithm?

Yifei Liu
Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706
yifeiliu@stat.wisc.edu

Sijing Li
Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706
sijingli@stat.wisc.edu

1 Introduction

Thoughout the course we encountered numerous descent algorithms, and proved various convergence theorems under different assumptions such as Lipschitzness, strong convexity and smoothness on the loss function. However, the output used in the proofs of theorems sometimes differs a lot. For example, in Lecture 5, we showed that for convex and L-Lipschitz loss funtion f, the convergence rate of gradient descent algorithm is given by

$$f\left(\frac{1}{T}\sum_{k=1}^{T} x_k\right) - f^* \leq \frac{\Delta_1 L}{\sqrt{T}}. \tag{1}$$

Here, T is the total number of iterations, $\{x_k\}_{k=1}^T$ is the sequence of outputs along the optimization path, $f*$ is the minimal function value, and Δ_1 is the distance between the initial point x_1 and the optimal point x^*. For λ-strongly convex and β-smooth loss function, we have the following convergence result

$$f(x_{T+1}) - f^* \leq \frac{\beta}{2}\exp\left(\frac{-T}{\kappa}\right)\Delta_1^2, \tag{2}$$

where $\kappa = \beta/\lambda$ is the condition number. For λ-strongly convex and L-Lipschitz function, Bubeck (2014) proves that the projected subgradient descent have the following convergence rate

$$f(\sum_{k=1}^{T} \frac{2k}{T(T+1)} x_k) - f^* \leq \frac{2L^2}{\lambda(T+1)}. \tag{3}$$

From the above three examples we can see that when deriving the upper bound of an algorithm, differen kinds of output are used. And it's often *ad hoc* and tricky to pick the appropriate outputs to go through the proofs. This motivates us to find a general principle that can give a guideline on what output to use for loss functions satisfying different assumptions, and / or for different algorithms. A counterpart of this problem is that when people use a parallelized algorithm on, say, q machines and a sequence of length-q outputs are returned in total, then what will be the optimal way to combine these outputs to yield the best output. We formalize these problems in the following section.

2 Problem formulation

We state these problems as two versions of problem, the single-machine problem and the multiple-machine problem. For simplicity, we consider here the unconstrained convex optimization problem, namely,

$$\text{minimize } f(x), \tag{4}$$

where $x \in \mathbb{R}^d$.

2.1 Single machine problem

Consider a general descent algorithm implemented by a single machine, where the update is given by

$$x_{k+1} = x_k - \gamma\alpha_k.$$

Here α_k is the descent direction in each step. For example, $\alpha_k = \nabla f(x_k)$ correspondes to the gradient descent algorithm. And γ is the stepsize. Suppose we iterate T times and collect $\{x_k\}_{k=1}^T$. We aim to find diagonal matrices $\{\hat{D}_k\}_{k=1}^T$ such that $\{\hat{D}_k\}$ minimizes

$$f(\sum_{k=1}^{T} D_k x_k), \tag{5}$$

w.r.t. $\{D_k\}$, where $D_k = \text{diag}\{d_{11}^{(k)}, \cdots, d_{dd}^k\}$.

Notice that the above form allows us to perform not only linear combinations of the original outputs $\{x_k\}$ but also scaling them differently in different coordinates. Hence with this formulation, we can potentially get a final output whose objective function value is smaller than $min_k f(x_k)$!

2.2 Multiple machine problem

Now let's assume we have q machines in total, and each machine i runs an algorithm A and returns an output x_i. Now we would like to find $\{\hat{D}_i\}_{i=1}^q$ to minimize

$$f(\sum_{i=1}^{q} D_i x_i), \tag{6}$$

w.r.t. $\{D_i\}$, where $D_i = \text{diag}\{d_{11}^{(i)}, \cdots, d_{dd}^i\}$.

3 Potential approaches and directions

First of all we would like to approach these two questions from a theoretical point of view. We want to see whether the solutions of (5) and (6) are related to the structure (such as Lipschitzness, smoothness) of the loss fuction. Then we will try to solve these two problems with certain algorithm and see whether we can solve them efficiently. Notice that the size of the problems are closely related to the numbers of iterations T or the number of machines q, we would like to see numerically how the values of T or q will affects the results.

References

Bubeck, Sébastien (2014), "Convex optimization: Algorithms and complexity." *arXiv preprint arXiv:1405.4980.*