Understanding Performance of Gradient Based Algorithms

George Michailidis

gmichail@ucla.edu

STAT 102B

Key question

Do the gradient based algorithms studied throughout this term come with performance guarantees?

Short answer

Yes!

Short answer

Yes!

But, it depends on the nature of the function

A few useful definitions

The concept of convexity of a function plays a critical role in characterizing the performance of gradient based algorithms

Convex Functions

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$

Then, f is a convex function, if

$$f\big(\theta x + (1-\theta)y\big) \leq \theta f(x) + (1-\theta) f(y), \quad \forall\ x, y \in \mathbb{R}^n,\ \theta \in [0,1] \tag{1}$$

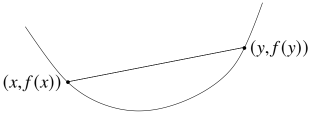

Figure 1: Illustrating the definition of a convex function

Important modifiers

1. Strictly convex: if

$$f\big(\theta x+(1-\theta)y\big) < \theta f(x)+(1-\theta)f(y), \ \ \forall\, x,y \in \mathbb{R}^n),\ x \neq y,\ \theta \in (0,1) \quad (2)$$

   In words, f is convex and has greater curvature than a linear function

2. Strongly convex: if

$$f - \frac{\mu}{2}\|x\|_2^2 \quad \text{is convex} \qquad (3)$$

   where $\mu > 0$
   In words, f is at least as convex as a quadratic function, or f is uniformly bounded below by the quadratic function $\frac{\mu}{2}\|x\|_2^2$

Note that strong convex $\Longrightarrow$ strict convex $\Longrightarrow$ convex

Characterization of convex functions: first order condition

Applies to differentiable functions

Suppose that f is differentiable and $\nabla f(x)$ exists at every $x \in \mathbb{R}^n$

$$f \text{ convex iff } f(y) \geq f(x) + \nabla f(x)^\top (y - x), \ \forall \ x, y \in \mathbb{R}^n \tag{4}$$

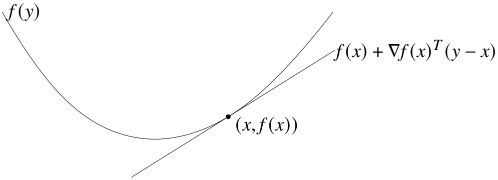

Figure 2: The first order Taylor approximation of convex f is a global underestimator of f

Characterization of convex functions: second order condition

Applies to twice differentiable functions

Suppose that f is twice differentiable and $\nabla^2 f(x)$ exists at every $x \in \mathbb{R}^n$

$$f \text{ convex iff } \nabla^2 f(x) \succeq 0, \ \forall\, x, y \in \mathbb{R}^n \tag{5}$$

i.e., the Hessian is positive semi-definite for all $x \in \mathbb{R}^n$

It is strictly convex iff $\nabla^2 f(x) \succ 0$

L-smooth functions

Suppose that f is differentiable

The Lipschitz constant L of the gradient is defined as the smallest constant such that:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall\; x, y \in \mathbb{R}^n$$

It quantifies how "smooth" the function's gradient $\nabla f(x)$
Geometrically, L controls the "curvature" of the function; larger L means the function can curve more sharply, requiring smaller gradient steps

Recall:

- For the quadratic function $L = \lambda_{\max}(Q)$ – i.e., the maximum eigenvalue of the coefficient matrix Q
- Hence, for the SSE(β) function $L = \lambda_{\max}(\frac{1}{n}X^\top X)$

In practice, L is usually unknown or expensive to compute

Convex functions and Optimization

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable and convex

- Suppose $x^\star$ is a local minimizer; then, $x^\star$ is also a global minimizer

  Hence,
  $$\nabla f(x^\star) = 0 \Rightarrow x^\star \text{ global minimizer}$$

  In contrast, for f non-convex, the gradient being zero could correspond to a local minimum, maximum, or saddle point

- For convex functions, the optimization algorithms discussed in this course come with convergence guarantees

ϵ-accurate solutions

Objective:

$$\min_{x \in \mathbb{R}^n} f(x)$$

The algorithms presented in this course aim to solve the gradient equation

$$\nabla f(x) = 0 \quad \text{iteratively}$$

In practice, the algorithm stops when the stopping criterion is satisfied, or in other words, when an ϵ-accurate solution is obtained

Definition:

ϵ-accurate solution: find $x \in \mathbb{R}^n$ such that

$$f(x) - f(x^\star) \leq \epsilon \quad \text{(convex)} \qquad \text{or} \qquad \|\nabla f(x)\|^2 \leq \epsilon \quad \text{(nonconvex)}$$

Goal:
Characterize number of iterations required to achieve an ϵ-accurate solution

Gradient Descent

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

Iteration complexities: (assuming constant step size $\eta \in (0, \frac{1}{L})$

- Strongly convex, L-smooth: $\mathcal{O}\left(\frac{L}{\mu}\log(\frac{1}{\epsilon})\right)$
- Convex, L-smooth: $\mathcal{O}\left(\frac{L}{\epsilon}\right)$
- Nonconvex, L-smooth: $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ for $\|\nabla f(x_k)\|^2 \leq \epsilon$

Practical Significance of iteration complexity results

Assuming required accuracy $\epsilon = 10^{-6}$, and a condition number $\frac{L}{\mu} = 100$ (i.e., fairly large), GD for a strongly convex function will require ≈ 1380 iterations, while for a convex function 10^6 iterations

In other words,

- Strongly convex functions exhibit logarithmic dependence on accuracy - high precision is "cheap"
- Convex function exhibit linear dependence on $1/\epsilon$ - high precision is expensive

Remark:
For L-smooth, convex functions, the relationships between tolerance and ϵ-accuracy is

$$\text{tolerance} \approx \sqrt{2L\epsilon}$$

GD with Nesterov momentum

$$y_k = x_k + \xi_k(x_k - x_{k-1})$$
$$x_{k+1} = y_k - \eta \nabla f(y_k)$$

Iteration Complexities:

- Strongly convex, L-smooth: $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\epsilon})\right)$
- Convex, L-smooth: $\mathcal{O}\left(\frac{L}{\epsilon^2}\right)$
- Nononvex, L-smooth: $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon^2}}\right)$ for $\|\nabla f(x_k)\|^2 \leq \epsilon$

Remark:
The impact of Nesterov momentum is fairly dramatic over GD (1000-fold improvement for tight ϵ-accuracy – 10^{-6})

No impact for nonconvex optimization problems

Stochastic Gradient Descent:

Analysis based on $s = 1$ and decreasing step size η_k along a fixed sequence, so that $\sum_{k=1}^{\infty} \eta_k = \infty$ and $\sum_{k=1}^{\infty} \eta_k^2 < \infty$

$$x_{k+1} = x_k - \eta_k \nabla f(x_k; \xi_k)$$

where ξ_k is a random sample.

Iteration complexities (with decaying step size):

- Strongly convex: $\mathcal{O}\left(\frac{1}{\epsilon}\right)$
- Convex: $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$
- Nonconvex: $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ for $\mathbb{E}[\|\nabla f(x_k)\|^2] \leq \epsilon$
  but with some modifications – otherwise it is $\mathcal{O}\frac{1}{\epsilon^4}$

Proximal Gradient Descent

Problem: Composite function minimization

$\min_x f(x)+g(x)$, where f is convex and L-smooth, g is convex, but possibly nonsmooth

Proximal update:

$$x_{k+1} = \text{prox}_{\eta g}\left(x_k - \eta \nabla f(x_k)\right)$$

Iteration complexities:

- Convex: $\mathcal{O}\left(\frac{1}{\epsilon}\right)$
- Strongly convex: $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$
- Nesterov momentum yields $\mathcal{O}(1/\sqrt{\epsilon})$ for convex and $\mathcal{O}(\log(1/\epsilon))$ for strongly convex f

Newton's Method

$$x_{k+1} = x_k - \eta \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$$

Convergence behavior:

- Far from the optimum, with "careful choice" of the step size, it closes the gap $f(x_k) - f(x^\star)$ by k/C (C determined by the Lipschitz constant of the gradient and the Hessian)
- Iteration complexity near the optimum: $\mathcal{O}(\log\log(1/\epsilon))$

In practice, once the algorithm comes "close" to the optimum, it requires less than 3-5 iterations for $\epsilon = 10^{-10}$