

Convex Optimization

Part 2: Accelerated gradient methods

Namhoon Lee

POSTECH

5 Oct 2022

Rates of convergence

So far we have seen rates of convergence for various classes of functions.

- ▶ For Lipschitz convex functions
- ▶ For smooth convex functions
- ▶ For smooth strongly convex functions

Q: Are they optimal? Can we do better?

First-order oracle model

Is it possible that there exists faster algorithms?

- ▶ In order to address this question, we need to consider our model first.

Black-box first-order oracle model of computation:

- ▶ At x_t it returns the evaluation of $f(x_t)$ and $\nabla f(x_t)$.
- ▶ The algorithm can do anything with these as long as it does not involve f .
- ▶ In general a black-box procedure is a mapping from “history” to the next query point, that it maps $(x_1, g_1, \dots, x_t, g_t)$ (with $g_s \in \partial f(x_s)$) to x_{t+1} .

Complexity of minimizing real-valued functions

Consider the following minimization problem

$$\min_{x \in [0,1]^d} f(x) ,$$

where f is a real-valued function.

Q: Suppose that you can use any algorithm under some oracle model. For example, how many zero-order oracle calls t do we need before we can guarantee $f(x_t) - f(x^*) \leq \epsilon$?

- ▶ It is impossible since given any algorithm we can construct an f where $f(x_t) - f(x^*) > \epsilon$ forever and real numbers are uncountable This means that to say anything in oracle model we need to make some assumptions on f .
- ▶ One of the simplest assumptions is Lipschitz f ; under this assumption, any algorithm requires at least $\Omega(1/\epsilon^d)$ iterations (e.g., $\mathcal{O}(1/\epsilon^d)$ by grid search).

Oracle lower bounds

For any $t \geq 0$, x_{t+1} is in the linear span of g_1, \dots, g_t , i.e., $x_{t+1} \in \text{Span}(g_1, \dots, g_t)$, and $B_2(R) = \{x \in \mathbb{R}^n : \|x\| \leq R\}$. Then we can prove oracle complexity lower bounds (Bubeck et al. 2015).

Theorem (non-smooth f)

Let $t \leq n, L, R > 0$. There exists a convex and L -Lipschitz function f such that

$$\min_{1 \leq s \leq t} f(x_s) - \min_{x \in B_2(R)} f(x) \geq \frac{RL}{2(1 + \sqrt{t})} .$$

- ▶ This means that the subgradient method is optimal (under oracle model).
- ▶ This does not mean that for a specific function that is Lipschitz and convex there does not exist a better algorithm than subgradient descent.

Theorem (smooth f)

Let $t \leq (n - 1)/2, \beta > 0$. There exists β -smooth convex function f such that

$$\min_{1 \leq s \leq t} f(x_s) - f(x^*) \geq \frac{3\beta}{32} \frac{\|x_1 - x^*\|^2}{(t + 1)^2} .$$

Theorem (smooth and strongly-convex f)

Let $\kappa > 1$. There exists β -smooth and α -strongly convex function $f : l_2 \rightarrow \mathbb{R}$ with $\kappa = \beta/\alpha$ such that for any $t \geq 1$ one has

$$f(x_t) - f(x^*) \geq \frac{\alpha}{2} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2(t-1)} \|x_1 - x^*\|^2 .$$

Momentum to reduce the gap

The oracle model of computations imply that under convexity (along with others) there might exist faster algorithms than the gradient methods we've seen which achieves faster rates of convergence.

- ▶ for smooth, strongly convex functions

Q: How can we accelerate the gradient methods to match the oracle bounds? What else do we have?

- ▶ The idea is to make use of “momentum” based on previous iterates $\{x_t, x_{t-1}, x_{t-2}, \dots\}$.

Polyak's momentum

Polyak's momentum, a.k.a. "heavy-ball method" (Polyak 1964)

$$x_{t+1} = x_t - \eta_t \nabla f(x_t) + \gamma_t (x_t - x_{t-1})$$

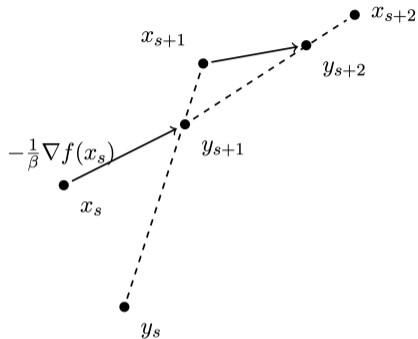
- ▶ Some reactive visualisation tool to show the effect of momentum ([link](#))

Nesterov's accelerated gradient descent (Nesterov 1983)

Start with an initial point $x_1 = y_1$ and iterate the following equations for $t \geq 1$

$$y_{t+1} = x_t - \frac{1}{\beta} \nabla f(x_t) ,$$
$$x_{t+1} = \left(1 + \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right) y_{t+1} + \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} y_t .$$

- ▶ This achieves the optimal rates for smooth (strongly) convex functions.



- First performs GD to go from x_t to y_{t+1} and then “slides” a bit further than y_{t+1} in the direction given by the previous point y_t .

Convergence analysis

Theorem (smooth and strongly convex)

Let f be α -strongly convex and β -smooth, then Nesterov's accelerated gradient descent satisfies

$$f(y_t) - f(x^*) \leq \frac{\alpha + \beta}{2} \|x_1 - x^*\|^2 \exp\left(-\frac{t-1}{\sqrt{\kappa}}\right).$$

Proof.

Define α -strongly convex quadratic functions Φ_s , $s \geq 1$ by induction as follows:

$$\begin{aligned}\Phi_1(x) &= f(x_1) + \frac{\alpha}{2} \|x - x_1\|^2, \\ \Phi_{s+1}(x) &= \left(1 - \frac{1}{\sqrt{\kappa}}\right) \Phi_s(x) + \frac{1}{\sqrt{\kappa}} \left(f(x_s) + \nabla f(x_s)^\top (x - x_s) + \frac{\alpha}{2} \|x - x_s\|^2 \right). \quad (1)\end{aligned}$$

Φ_s becomes a finer approximation (from below) to f in the following sense:

$$\Phi_{s+1}(x) \leq f(x) + \left(1 - \frac{1}{\sqrt{\kappa}}\right)^s (\Phi_1(x) - f(x)) \quad (2)$$

which can be proved by induction using α -strong convexity.

For now suppose the following inequality holds true (proof deferred to Bubeck):

$$f(y_s) \leq \min_{x \in \mathbb{R}^n} \Phi_s(x) . \quad (3)$$

Combining (1), (2), (3) and that $f(x) - f(x^*) \leq \frac{\beta}{2} \|x - x^*\|^2$ from β -smoothness obtains the rate given by theorem:

$$\begin{aligned} f(y_t) - f(x^*) &\leq \Phi_t(x^*) - f(x^*) \\ &\leq \left(1 - \frac{1}{\sqrt{\kappa}}\right)^{t-1} (\Phi_1(x^*) - f(x^*)) \\ &\leq \frac{\alpha + \beta}{2} \|x_1 - x^*\|^2 \left(1 - \frac{1}{\sqrt{\kappa}}\right)^{t-1} . \end{aligned}$$

□

Theorem (smooth and convex)

Let f be convex and β -smooth, then Nesterov's accelerated gradient descent satisfies

$$f(y_t) - f(x^*) \leq \frac{2\beta\|x_1 - x^*\|^2}{t^2} .$$

Proof.

First define a time-varying sequence λ_t and $\gamma_t \leq 0$ as follows

$$\lambda_0 = 0, \lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2}, \text{ and } \gamma_t = \frac{1 - \lambda_t}{\lambda_{t+1}}.$$

Then the algorithm becomes

$$y_{t+1} = x_t - \frac{1}{\beta} \nabla f(x_t), \quad x_{t+1} = (1 - \gamma_t)y_{t+1} + \gamma_t y_t.$$

Now use convexity, β -smoothness (at x_s) and the algorithm

$$\begin{aligned} f(y_{s+1}) - f(y_s) &\leq \nabla f(x_s)^\top (x_s - y_s) - \frac{1}{2\beta} \|\nabla f(x_s)\|^2 \\ &= \beta(x_s - y_{s+1})^\top (x_s - y_s) - \frac{\beta}{2} \|\nabla x_s - y_{s+1}\|^2 \end{aligned} \quad (4)$$

Similarly

$$f(y_{s+1}) - f(x^*) \leq \beta(x_s - y_{s+1})^\top (x_s - x^*) - \frac{\beta}{2} \|\nabla x_s - y_{s+1}\|^2 \quad (5)$$

Now define $\delta_s = f(y_s) - f(x^*)$, then $(\lambda_s - 1) \times (4) + (5)$ gives

$$\lambda_s \delta_{s+1} - (\lambda_s - 1) \delta_s \leq \beta(x_s - y_{s+1})^\top (\lambda_s x_s - (\lambda_s - 1)y_s - x^*) - \frac{\beta}{2} \lambda_s \|y_{s+1} - x_s\|^2$$

By multiplying λ_s to the above, using $\lambda_{s-1}^2 = \lambda_s^2 - \lambda_s$ gives

$$\begin{aligned} \lambda_s^2 \delta_{s+1} - \lambda_{s-1}^2 \delta_s &\leq \frac{\beta}{2} \left(2\lambda_s(x_s - y_{s+1})^\top (\lambda_s x_s - (\lambda_s - 1)y_s - x^*) - \|\lambda_s(y_{s+1} - x_s)\|^2 \right) \\ &= \frac{\beta}{2} \left(\underbrace{\|\lambda_s x_s - (\lambda_s - 1)y_s - x^*\|^2}_{u_s} - \underbrace{\|\lambda_s y_{s+1} - (\lambda_s - 1)y_s - x^*\|^2}_{u_{s+1}^?} \right) \\ &= \frac{\beta}{2} (\|u_s\|^2 - \|u_{s+1}\|^2) \end{aligned}$$

To check u_{s+1} , multiply λ_{s+1} to the algorithm definition and use $\gamma_s = \frac{1-\lambda_s}{\lambda_{s+1}}$

$$x_{s+1} = (1 - \gamma_s)y_{s+1} + \gamma_s y_s$$

$$\iff \lambda_{s+1}x_{s+1} = \lambda_{s+1}y_{s+1} - \lambda_{s+1}\gamma_s y_{s+1} + \lambda_{s+1}\gamma_s y_s$$

$$\iff \lambda_{s+1}x_{s+1} - (\lambda_{s+1} - 1)y_{s+1} = \lambda_s y_{s+1} - (\lambda_s - 1)y_s$$




Summing for $t - 1$ iterations gives

$$\delta_t \leq \frac{\beta}{2\lambda_{t-1}^2} \|u_1\|^2$$

which concludes proof with $\lambda_{t-1} \geq t/2$, achieving the convergence rate of $\mathcal{O}(1/t^2)$. \square

Any questions?

References I

-  Bubeck, Sébastien et al. (2015). “Convex optimization: Algorithms and complexity”. In: *Foundations and Trends® in Machine Learning* 8.3-4, pp. 231–357.
-  Nesterov, Yurii E (1983). “A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$ ”. In: *Dokl. akad. nauk Sssr*. Vol. 269, pp. 543–547.
-  Polyak, Boris T (1964). “Some methods of speeding up the convergence of iteration methods”. In: *Ussr computational mathematics and mathematical physics* 4.5, pp. 1–17.