CSED490Y: Optimization for Machine Learning

Week 11: Accelerated methods

Namhoon Lee

POSTECH

Spring 2022

Admin

Midterm exam results:

- (overall) mean: 21.1, std: 6.8.
- Your scores are available on PLMS.
- If you want to discuss your result, contact TA by Friday this week.

Midway group presentation signup:

- Groups 7, 10, 13 not done yet. Please sign up here ASAP.

Rates of convergence $\Rightarrow$ Q: can we accelerate?

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

G

- Lipschitz and convex

$\|g_x\| \le G, \; g_x \in \partial f(x) \qquad \varepsilon \sim O(1/\sqrt{T}) \Leftrightarrow T \sim 1/\varepsilon^2$

- Smooth and convex

L, β

$\varepsilon \sim 1/T \Leftrightarrow T \sim 1/\varepsilon$

- Smooth and strongly convex

$\beta \qquad \alpha$

$\varepsilon \sim \rho^t \Leftrightarrow T \sim \log(1/\varepsilon)$

condition number

$\rho = \left(\frac{\beta-\alpha}{\beta+\alpha}\right) = \frac{\kappa-1}{\kappa+1}$ with $\kappa = \beta/\alpha$

Rates of convergence

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

- Lipschitz and convex
- Smooth and convex
- Smooth and strongly convex

Questions: 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.

$$x_t \longrightarrow \boxed{\text{oracle}} \longrightarrow f(x_t),\ \nabla f(x_t)$$

only this → zero order oracle

derivative free methods

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 $(x_1, g_1, \ldots, x_t, g_t)$ (with $g_s \in \partial f(x_s)$) to x_{t+1}.

Complexity of minimizing real-valued functions

We need to analyse a class of functions under some assumptions.

- Lipschitz, smooth, and/or (strongly) convex functions.

For example, consider minimizing the following

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

and suppose that you can use any algorithm under some oracle model.

Complexity of minimizing real-valued functions

We need to analyse a class of functions under some assumptions.

- Lipschitz, smooth, and/or (strongly) convex functions.

For example, consider minimizing the following

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

and suppose that you can use any algorithm under some oracle model.

Q: How many zero-order oracle calls t 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, which means that to say anything in oracle model we need 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 $g_1, ..., g_t$, *i.e.*, $x_{t+1} \in \text{Span}(g_1, ..., g_t)$, 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

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

- 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.

Oracle lower bounds

Theorem (smooth f)

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

GD

$$\min_{1 \le s \le t} f(x_s) - f(x^*) \ge \frac{3\beta}{32} \frac{\|x_1 - x^*\|^2}{(t+1)^2} \,. \quad \sim \; 1/t^2 \overset{\text{better}}{>} 1/t$$

Theorem (smooth and strongly-convex f)

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

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

Momentum to reduce the gap

Under convexity (and other assumptions), we know the rates of convergence is faster than those previously seen under the oracle model of computations.

$x_t \rightarrow \boxed{\text{oracle}} \rightarrow f(x_t), \nabla f(x_t)$

Q: Can we accelerate the algorithm?

- Yes we can!

GD $x_{t+1} = x_t - \eta \nabla f(x_t)$ (Markovian)

Q: How can we match these bounds? What else do we have?

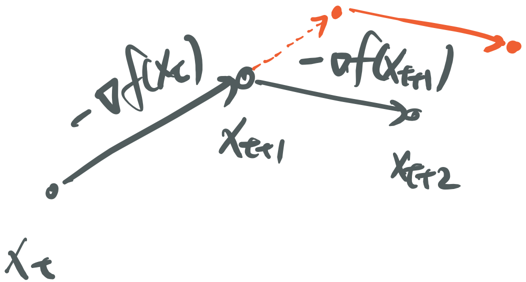
- Previous iterates $\{x_t, x_{t-1}, x_{t-2}, \ldots\}$.

The idea is to use the concept of "momentum".

Momentum

Heavy-Ball method

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



Nesterov's accelerated gradient descent

Nesterov's Accelerated Gradient Descent (initialized with $x_1 = y_1$):

$$y_{t+1} = x_t - \frac{1}{\beta}\nabla f(x_t) \ ,$$

$$x_{t+1} = (1-\gamma_t)y_{t+1} + \gamma_t y_t \ .$$

- 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.
- For smooth convex function, this achieves the optimal rate. $\sim 1/T^2$

Theorem (Nesterov 1983)

Let f be a convex and β-smooth function, then Nesterov's Accelerated Gradient Descent satisfies

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

Proof (1/3)   Sebastien Bubeck "Convex optimization"

$$\Rightarrow f(\underbrace{x - \tfrac{1}{\beta}\nabla f(x)}) - f(y) \le -\frac{1}{2\beta}\|\nabla f(x)\|_2^2 + \nabla f(x)^T(x-y)$$

convexity
smoothness

① $f(y_{t+1}) - f(y_t) \le -\frac{\beta}{2}\|y_{t+1} - x_t\|_2^2 - \beta(y_{t+1} - x_t)^T(x_t - y_t)$ by def. algo.

② $f(y_{t+1}) - f(x^*) \le -\frac{\beta}{2}\|y_{t+1} - x_t\|_2^2 - \beta(y_{t+1} - x_t)^T(x_t - x^*)$

① $\times (\lambda_t - 1)$ + ②, with $\delta_t = f(y_t) - f(x^*)$, $\delta_{t+1} = f(y_{t+1}) - f(x^*)$

↳ $\lambda_t \delta_{t+1} - (\lambda_t - 1)\delta_t \le -\frac{\beta}{2}\lambda_t \|y_{t+1} - x_t\|_2^2 - \beta(y_{t+1} - x_t)^T(\lambda_t x_t - (\lambda_t - 1)y_t - x^*)$

└ multiply by λ_t

Proof (2/3)

Choose $\boxed{\lambda_t = \dfrac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}}$, $\lambda_0 = 0$

$\hookrightarrow \quad \lambda_{t-1}^2 = \lambda_t^2 - \lambda_t$

$$\lambda_t^2 \delta_{t+1} - \lambda_{t-1}^2 \delta_t \le -\frac{L}{2}\left(\|\lambda_t (y_{t+1} - x_t)\|_2^2 + 2\lambda_t (y_{t+1} - x_t)^T (\lambda_t x_t - (\lambda_t - 1) y_t - x^*)\right)$$

$$= -\frac{L}{2}\left(\|\lambda_t y_{t+1} - (\lambda_t - 1) y_t - x^*\|_2^2 - \|\lambda_t x_t - (\lambda_t - 1) y_t - x^*\|_2^2\right) \quad - (A)$$

Proof (3/3)

$$x_{t+1} = (1-\gamma_t) y_{t+1} + \gamma_t y_t$$

multiply $\lambda_{t+1} \Rightarrow \quad \lambda_{t+1} x_{t+1} = \lambda_{t+1} y_{t+1} - \lambda_{t+1}\gamma_t y_{t+1} + \lambda_{t+1}\gamma_t y_t$

$$\boxed{\gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}}}$$

$$= (\lambda_{t+1}-1) y_{t+1} + \lambda_t y_{t+1} - (\lambda_t - 1) y_t$$

$$\Leftrightarrow \lambda_{t+1} x_{t+1} - (\lambda_{t+1}-1) y_{t+1} = \lambda_t y_{t+1} - (\lambda_t - 1) y_t \quad - \text{(B)}$$

put (A) & (B) together,

sum T iterations (*)

$$(*)\ \underbrace{\lambda_t^2 \delta_{t+1} - \lambda_{t-1}^2 \delta_t \le \frac{\beta}{2}\left(\|u_t\|_2^2 - \|u_{t+1}\|_2^2\right)} \Rightarrow \lambda_T^2 \delta_{T+1} \le \frac{\beta}{2}\|u_1\|_2^2$$

where $u_t = \lambda_t x_t - (\lambda_t - 1) y_t - x^*$

$$\delta_{T+1} \le \frac{\beta \|u_1\|_2^2}{2\lambda_T^2} \sim 1/T^2$$

Accelerated proximal gradient method

Consider minimizing a composite function f

$$\min_x f(x) = g(x) + h(x)$$

where g is convex and differentiable, and h is convex.

Accelerated proximal gradient method (Beck and Teboulle 2009):

$$v = x_{t-1} + \frac{t-2}{t+1}(x_{t-1} - x_{t-2})$$
$$x_t = \text{prox}_\eta(v - \eta \nabla g(v))$$

for $t = 1, 2, 3, \dots$

- First step $t = 1$ is just usual proximal gradient update.
- After that, v carries some "momentum" from previous iterations.
- $h = 0$ gives accelerated gradient method.

Convergence

Theorem

For $f(x) = g(x) + h(x)$ where g is convex and differentiable, and h is convex, accelerated proximal gradient method with fixed step size $\eta \leq 1/L$ satisfies

$$f(x_t) - f(x^*) \leq \frac{2\|x_0 - x^*\|_2^2}{\eta(t+1)^2} \ .$$

- It achieves the optimal rate of convergence $\mathcal{O}(1/t^2)$.

FISTA

L1-regularized least squares or Lasso problem

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \ .$$

Recall that proximal mapping results in the soft-thresholding operation $S_{\eta\lambda}(\cdot)$ applied to the gradient update (*i.e.* ISTA).

Applying acceleration gives FISTA (Beck and Teboulle 2009):

$$v = x_{t-1} + \frac{t-2}{t+1}(x_{t-1} - x_{t-2}) \ ,$$
$$x_t = S_{\eta_t \lambda}(v - \eta_t A^\top (Av - y)) \ .$$

Thank you

Any questions?

Credits

A lot of material in this course is borrowed or derived from the following:

- Numerical Optimization, Jorge Nocedal and Stephen J. Wright.
- Convex Optimization, Stephen Boyd and Lieven Vandenberghe.
- Convex Optimization, Ryan Tibshirani.
- Optimization for Machine Learning, Martin Jaggi and Nicolas Flammarion.
- Optimization Algorithms, Constantine Caramanis.
- Advanced Machine Learning, Mark Schmidt.

References I

- Beck, Amir and Marc Teboulle (2009). “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM journal on imaging sciences.*
- 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.