IE510 Applied Nonlinear Programming

Lecture 5: Optimal First Order Method

Ruoyu Sun

Feb 20, 2018

Side: Recent Q&A

Reddit: The Future (and Present) of Artificial Intelligence AMA (2018/02/18)

Question: What are some crucial skills/ knowledge I should possess in order to succeed in this field (AI)? (from a physics PhD)

Yann LeCun: **Crucial skills**:

- good skills/intuition in continuous mathematics (linear algebra, multivariate calculus, probability and statistics, optimization...).
- Good programming skills.
- Good scientific methodology.
- Above all: creativity and intuition

Review of Momentum

Last algorithm we learned (last last week): GD with momentum, a.k.a. heavy ball method.

Main result: achieve faster rate of $1 - 1/\sqrt{\kappa}$ for quadratic problems

How about non-quadratic problems, like logistic regression?

- For many problems, momentum seems to help
- But in theory, NO!
  e.g. [Lessard et al. 2016] 1-dim counter-example

Today

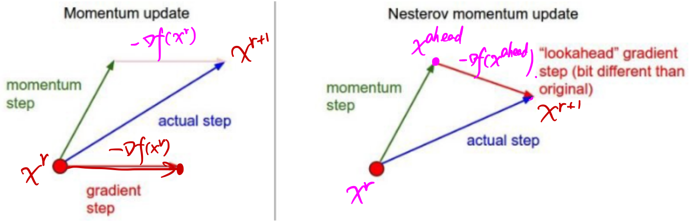
- **Today**: Optimal first order method, Nesterov's accelerated method
- After today's course, you will be able to
  - Draw figures to illustrate the difference of Nesterov's method and HB
  - Tell the difference of Nesterov's method and HB in theory
  - Describe in what sense Nesterov's accelerated method is "optimal"
- Advanced goals (optional):
  - Appreciate the beauty of analyzing "optimal" optimization method
    - Analogy: Shannon information theory 1948; NP-hardness 1960's,1970's
  - Identify possible ways to go beyond "optimal method"

Outline

1. Nesterov momentum: Intuitive Understanding
2. Formal Definition and Main Results
3. Optimal First Order Method

Nesterov's accelerated gradient method (Nesterov momentum)

Previous view: slip and shift together (update velocity)



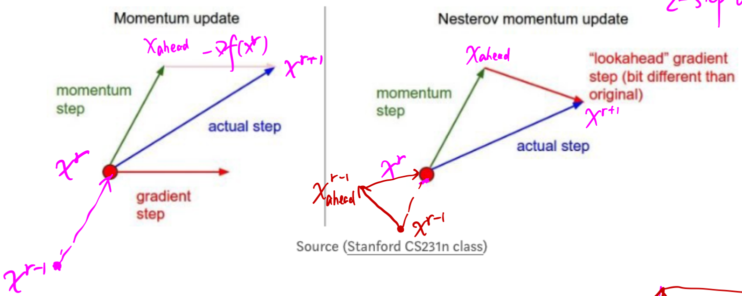
Source (Stanford CS231n class)

Another View: Slip then Shift

New view: let the car slip for a while, then stop and move

$$x_{ahead} - x^r = \beta(x^r - x^{r-1}).$$

2-step update

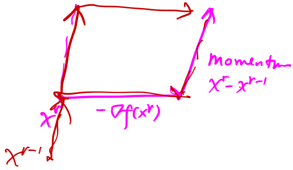


Source (Stanford CS231n class)

Question. Why not gradient step first, then momentum?

(i) If $-\nabla f(x^r)$ first, then $(x^r - x^{r-1})$, it is the same as first $(x^r - x^{r-1})$ + second $-\nabla f(x^r)$;

(ii) If go along $-\nabla f(x^r)$ first, the "true momentum" is no longer $x^r - x^{r-1}$.

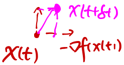


Discretization View

Best way: continuously changing direction (gradient step) + moving along momentum along the path .

$$x(t+\delta t) = x(t) - \alpha \nabla f(x(t)) + \beta(x(t) - x(t-\delta t))$$

Graph:



At time t, you're at $x(t)$.
After time δt, you are at $x(t+\delta t)$.
After time $2\delta t$, you are at $x(t+2\delta t)$

$$= x(t+\delta t) \underbrace{- \alpha \nabla f(x(t+\delta t))}_{\text{new grad}} \underbrace{+ \beta(x(t+\delta t) - x(t))}_{\text{new momentum}}$$

$\delta t = 0.1s$,
$\delta t = 0.01s$, ...

Equivalently: stepsize $\alpha \to 0$.

A better discretization?

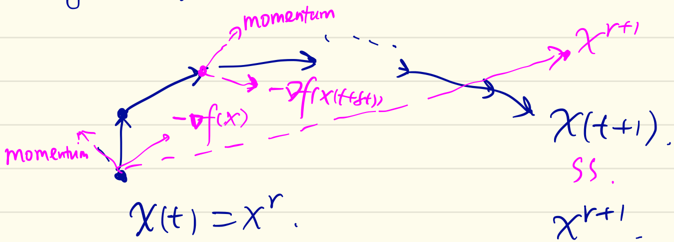
Question: Why $\alpha \to 0$ is slower?
It seems changing direction continuously is "faster" when driving?

Answer: Take more computation time.
either in brain or computer.

Remark: Discretization is nontrivial; see course SE420, and
Su et al.-2014, A differential equation for modeling Nesterov?s accelerated gradient method: Theory and insights.

Wilson et al.-2016: A Lyapunov Analysis of Momentum Methods in Optimization

Continuously changing direction:



Assume $\delta t = 1/1000$.

Use information. $x(t), x(t+\delta t), x(t+2\delta t), \cdots, x(t+999\delta t), x(t+1)$,

gradient: $\nabla f(x(t)), \nabla f(x(t+\delta t)), \nabla f(x(t+2\delta t)), \cdots, \nabla f(x(t+999\delta t)), \nabla f(x(t+1))$.

momentum $x(t)-x(t-\delta t), x(t+\delta t)-x(t), \cdots \cdots, x(t+1)-x(t+999\delta t)$

Goal: Approximate the final point $x(t+1)$ of the path $(x(t), x(t+\delta t), \cdots, x(t+1))$ by x^{r+1}, from few pieces of information from above.

HB: Just pick $x(t)-x(t-1)$ and $\nabla f(x(t))$.

Nesterov: Pick $x(t)-x(t-1)$ and $\cancel{\nabla f(x(t))}$ $\nabla f(x(t)+\text{momentum})$

Two Discretization Views

View 1: simultaneous discretization

- HB: Pick the gradient at the starting point x to approximate gradients along the way
- Nesterov: Pick the gradient at the look-ahead position (a "middle" point) to approximate the gradients along the way

2-step

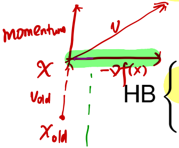
View 2: decomposed discretization

- HB: move along momentum first, then move along $-\nabla f(x)$
- Nesterov: move along momentum first to x_{ahead}, then move along $-\nabla f(x_{\text{ahead}})$

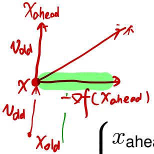
Can you write down the four update equations, 2 for HB and 2 for Nesterov's method?

Summary of Two Views

- **View 1**: Slip and Shift together



$$\text{HB}\begin{cases} v = \beta\, v_{\text{old}} - \alpha \nabla f(x), \\ x_{\text{new}} = x + v. \end{cases}$$



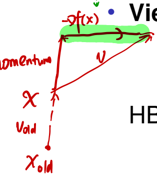
$$\text{Nesterov}\begin{cases} x_{\text{ahead}} = x + \beta v_{\text{old}}, \\ v = \beta\, v_{\text{old}} - \alpha \nabla f(x_{\text{ahead}}) \\ x_{\text{new}} = x + v. \end{cases}$$

(Stanford CS231n version)

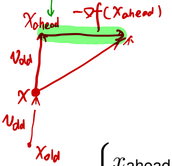
different view

different view

- **View 2**: Slip then Shift



$$\text{HB}\begin{cases} x_{\text{ahead}} = x + \beta(x - x_{\text{old}}), \\ x_{\text{new}} = x_{\text{ahead}} - \alpha \nabla f(x). \end{cases}$$



$$\text{Nesterov}\begin{cases} x_{\text{ahead}} = x + \beta(x - x_{\text{old}}), \\ x_{\text{new}} = x_{\text{ahead}} - \alpha \nabla f(x_{\text{ahead}}). \end{cases}$$

Outline

Nesterov's Momentum

Now we define a simple version of Nesterov's accelerated gradient method (1983).

x_{ahead}

$$\begin{cases} y^r = x^r + \beta_r(x^r - x^{r-1}), & \text{slip due to momentum} \\ x^{r+1} = y^r - \alpha \nabla f(y^r). & \text{move along gradient} \end{cases} \quad (1)$$

- Simplest stepsize (for strongly convex case): $\beta = (1 - \frac{1}{\sqrt{\kappa}})^2 \approx 1 - \frac{2}{\sqrt{\kappa}}$.

$$\alpha = 1/L, \quad \beta_r = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}. \quad \text{constant parameters} \quad (2)$$

$= 1 - \frac{2}{\sqrt{\kappa}+1}$

- Simplest stepsize (for convex case):

$$\alpha = 1/L, \quad \beta_r = \frac{r-1}{r+3}. \quad \text{time-dependent parameters} \quad (3)$$

$= 1 - \frac{4}{r+3} \approx 1 - O(\frac{1}{r})$.

Nesterov's Momentum: Results

Theorem 5.1 For strongly convex problems, Nesterov's method (4) with the stepsize choice in (2) satisfies

Assumption: $\mu I \leq \nabla^2 f(x) \leq LI$

$$f(x^r) - f^* \leq L(1 - \frac{1}{\sqrt{\kappa}})^{2r} \|x^0 - x^*\|^2.$$

Let $(1-\frac{1}{\sqrt{\kappa}})^r = \varepsilon$

$\Rightarrow r \approx \sqrt{\kappa} \log \frac{1}{\varepsilon}$.

For convex problems, Nesterov's method (4) with the stepsize choice in (3) satisfies

Assumption: $\nabla^2 f(x) \leq LI$.

$$f(x^r) - f^* \leq \frac{2L}{(r+1)^2} \|x^0 - x^*\|^2.$$

$O(\frac{1}{r^2}) = \varepsilon$

$\Rightarrow$ # of ite $r = \frac{1}{\sqrt{\varepsilon}}$.

- For strongly convex case, iteration complexity $O(\sqrt{\kappa} \log 1/\epsilon)$, faster than $O(\kappa \log 1/\epsilon)$ of GD.
- For convex case, iteration complexity $O(1/\sqrt{\epsilon})$, faster than ____$1/\varepsilon$____ of GD.

See Nesterov "Introductory lectures on convex optimization" for details. A shorter introduction in Donoghue,Candes "Adaptive Restart for Accelerated Gradient Schemes".

General Stepsize Rule

The original stepsize rule by Nesterov is rather general:

$$\begin{cases} y^r = x^r + \beta_r(x^r - x^{r-1}), \\ x^{r+1} = y^r - \alpha_r \nabla f(y^r). \end{cases} \tag{4}$$

$\alpha_r \le 1/L$, and one general choice of β_r is:

quadratic equation on θ_{r+1}

General Rule 1: $q \in [0,1]$, $\theta_{r+1}^2 = (1-\theta_{r+1})\theta_r^2 + q$.

$$\beta_{r+1} = \frac{\theta_r(1-\theta_r)}{\theta_r^2 + \theta_{r+1}}.$$

Freedom: q & θ_0.

Consider $\theta_0 = 1$ in this page. → plug into above, $1^2 = (1-1)\cdot 1 + 1$ holds.

Let $q = 1$: then $\theta_r = 1$, $\beta_r = 0$. Recover GD.

When convex, let $q = 0$: then $\theta_{r+1} = \frac{\theta_r(\sqrt{\theta_r^2+4} - \theta_r)}{2}$, then

$$f(x^r) - f^* \le \frac{4L}{(r+2)^2}\|x^0 - x^*\|^2.$$

When strongly convex, let $q = 1/\kappa = \mu/L$, linear convergence:

$$f(x^r) - f^* \le L\left(1 - \frac{1}{\sqrt{\kappa}}\right)^r \|x^0 - x^*\|^2.$$

Derive Two Simplest Stepsize Rules

Let the initial point of the auxiliary sequence and the parameter be

$$\theta_0 = 1/\sqrt{\kappa}, q = 1/\kappa.$$

Then we obtain

$$\text{constant} \quad \beta_r = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1},$$

which recovers (2).

Another general stepsize rule different from Rule 2 (for convex case):

General Rule 2: $a_0 \in [0, 1], a_{r+1} = (1 + \sqrt{4a_{r-1}^2 + 1})/2.$

$$\beta_r = \frac{a_r - 1}{a_{r+1}}.$$

Let $a_0 = 0$, then $a_r = \frac{r+1}{2}$, and

$$\beta_r = 1 - \frac{4}{r+3},$$

which recovers (3).

Lots of Interpretations

People found Nesterov's original proof HARD to understand.

Lots of interpretations: (many are obtained in the past 5 years)

- Chebychev polynomial (related); approximation theory
  - Hardt blog: "Zen of Gradient Descent", but does not recover Nesterov's method
  - HB method equivalent to Chebychev iteration method

$\cos(kx) = f(\cos(x))$.

- ODE interpretation (2nd order ODE, Hamiltonian system; not simple)
- geometric idea (related to ellipsoid method; different method)
- game in primal-dual method
- upper/lower bound estimate (still magical)
- ......

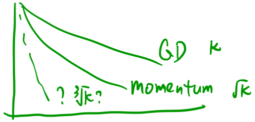
Outline

Further Improvement of Momentum?

- : Question: can we do better than momentum methods?
  - "Better" can mean many things...
  - What Nesterov's method does: extend $\sqrt{\kappa}$ result from quadratic to convex/strongly convex
- **Question**: Can we improve the bound $\tilde{O}(\sqrt{\kappa})$, even just for quadratic case?
- More history info: can we use three or four terms in history, to get $\tilde{O}(\kappa^{1/3})$ or even better bound?

  [Polyak'1974] mentioned: this is unclear.

- Better momentum: can we "discretize" better, to obtain a faster algorithm than Nesterov's method?



Optimal Methods

- Surprisingly, the answer is NO, in a certain sense.
- We will see:
  - Nesterov's method is (order) **optimal** for convex/strongly convex problems, in a certain sense. (not just quadratic!)
  - For strongly convex quadratic problems, both HB and Nesterov's method are (order) **optimal** in that sense.
- In what sense? We will discuss later.
- Why should you care?

Why Should You Care About Optimal Methods

- **Engineers** should care since:
  - If your boss pushes you to find faster algorithms, you tell him/her: no way! My algorithm is “optimal”.
  - Save your time. In an ideal world, for any problem, just find the “optimal” algorithm, then no need to worry.
- “Why momentum really matters” says: this result should be taken “spiritually”, not literally.
- **Theoreticians** should care since:
  - Don’t waste your time to look for a faster algorithm (in theory) unless...
  - unless you really understand the lower bound, and avoid those algorithms that will definitely fail to improve

Why Should You Care About Optimal Methods

- **Engineers** should care since:
  - If your boss pushes you to find faster algorithms, you tell him/her: no way! My algorithm is "optimal".
  - Save your time. In an ideal world, for any problem, just find the "optimal" algorithm, then no need to worry.

- "Why momentum really matters" says: this result should be taken "spiritually", not literally.

- **Theoreticians** should care since:
  - Don't waste your time to look for a faster algorithm (in theory) unless...
  - unless you really understand the lower bound, and avoid those algorithms that will definitely fail to improve

Oracle Model

- **Oracle model** Ω for the first order algorithms:
  - given any x^r, the oracle returns $\nabla f(x)$.
  - at iteration r, the algorithm generates x^{r+1} in span$(x^0, x^1, \ldots, x^r, \nabla f(x^0), \ldots, \nabla f(x^r))$.

  In short, the only allowable information is $\{x^i\}$ and $\{\nabla f(x^i)\}$
- **Definition**: The (iteration) complexity of algorithm $\mathcal{A} \in \Omega$, for a function f, is

$$C_\epsilon(\mathcal{A}; f) = \min\{r \mid f(x^r) - f(x^*) \le \epsilon\}.$$

  number of iterations to achieve error ϵ $O(\frac{1}{\varepsilon})$ for GD.
- **Definition**: The complexity of algorithm $\mathcal{A} \in \Omega$, for a function class F, is

$$C_\epsilon(\mathcal{A}; F) = \sup_{f \in F} \min\{r \mid f(x^r) - f(x^*) \le \epsilon\}.$$

  number of iterations to achieve error ϵ

What Algorithm is Covered?

What is covered by Ω?

- GD with constant stepsize;
- GD with diminishing stepsize, or any line search rule.
- HB method;
- Nesterov's method

What is NOT covered by Ω?

- Newton method
- Using $-D\nabla f(x^r)$ as direction, where D is positive definite
- Many others, e.g., AdaGrad, BFGS, etc.

Lower Bound

- Let $P(D, L)$ be the class of smooth unconstrained convex optimization problems with

  $\|x^0 - x^*\| \le D$;

  $\nabla^2 f(x) \preceq LI, \quad \forall x$.

- Let $S(D, L, \mu)$ be the class of smooth unconstrained convex optimization problems, which satisfies the conditions of $P(D, L)$ and additionally

  for some $\mu > 0$, $\mu I \preceq \nabla^2 f(x)$, $\forall x$

Remark: For simplicity, we use $\nabla^2 f(x) \preceq LI$, $\forall x$. It also works for $\|\nabla f(x) - \nabla f(y)\| \le LI$, $\forall x,y$.

- **Result:** For convex class $P(D, L)$

No matter what algorithm in Ω you pick, there always exists a problem such that the iteration complexity is at least xxx.

dimension of x

$$\inf_{\mathcal{A}\in\Omega} C_\epsilon(\mathcal{A}) \ge O(1)\min\{n, \frac{D\sqrt{L}}{\sqrt{\epsilon}}\}$$

it means $C_\epsilon(\mathcal{A}, P(D,L))$

For strongly convex class $S(D, L, \mu)$

$$\inf_{\mathcal{A}\in\Omega} C_\epsilon(\mathcal{A}) \ge O(1)\min\{n, \sqrt{\kappa}\log(1/2\epsilon)\}$$

Limitation of Lower Bound

Is that the end?

Two big issues:

- Only about $\#$ of iterations; per-iteration time ignored
- Bound of n on number of iterations

Active area of research!

Conclusion of Today

Can you summarize yourself?

- Nesterov's method is "optimal" for convex/strongly convex problems, under certain assumptions
  - only among 1st order methods
  - number of iterations upper bounded
- Save your time on trying different momentum!! More history does not help (too much).

Conclusion of Today

Can you summarize yourself?

- Nesterov's method is "optimal" for convex/strongly convex problems, under certain assumptions
  - only among 1st order methods
  - number of iterations upper bounded by n.

For engineers:

- Save your time on trying different momentum!! More history does not help (too much).