Deeper Exploration of Momentum-type Methods

Ruoyu Sun

1/30

Learning Goals of Today

Deeper exploration of momentum.
After today's lecture, you should be able to:

- tune momentum coefficients better
- write down the matrix recursion form of HB
- analyze empirical behavior of HB based on eigenvalues of the update matrix
- explain which methods are optimal first order methods (optional)

2/30

Outline

GD with Momentum

We solve a linear regression problem by **GD with momentum**

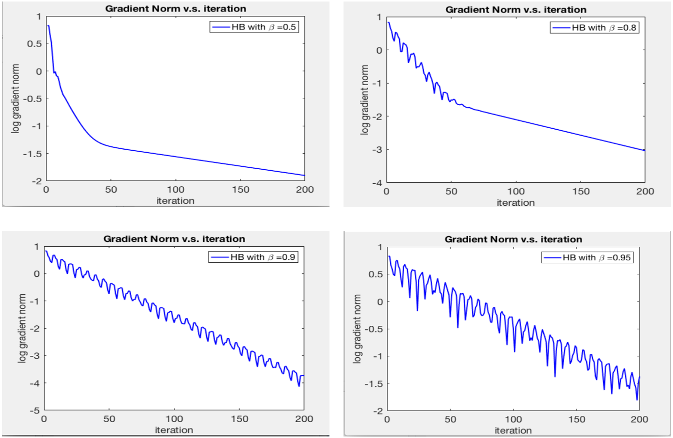
$$w^{r+1} = w^r - \alpha \nabla f(w^r) + \beta(w^r - w^{r-1}).$$

Algorithm parameters

- $\alpha = 1/L$, where $L = \lambda_{\max}(X'X)$
- $\beta \in (0, 1)$ and close to 1

Small data setting: $N = 10, d = 5$

Plots for Different β: Observations



Observations:

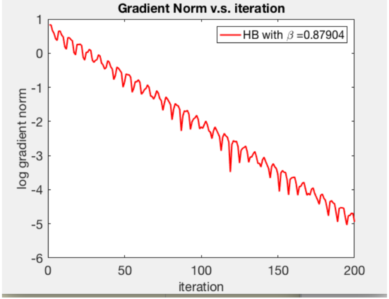
- There are oscillations (NOT descent algorithm!)
- $\beta = 0.9$ is approx. the best choice of β (for high accuracy)
- For low accuracy (e.g. 10^{-1}), $\beta = 0.5$ is about the best choice

Optimal β

What is the optimal β?

According to theory, for $\alpha = 1/L$, "best" β is $(1 - 1/\sqrt{\kappa})^2$

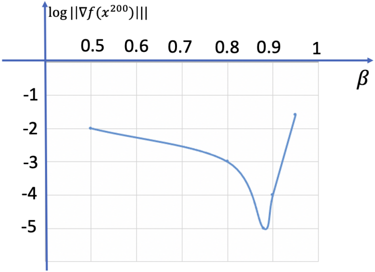
- About 0.88 for this problem instance



How to Tune β Based on Plots?

- What we have observed:
  - Optimal β (and larger): straight line, more fluctuation
  - Small β: multi-stage behavior, similar to GD
- **Empirical guidance**: (when solving quadratic problems) in HB, assume $\alpha = 1/L$, then tune β till (for log of gradient norm):
  - whole plot has fluctuation
  - plot becomes straight

Gradient Norm v.s. β



Sensitivity of parameter?

- The range 0.8-0.9 is good (in terms of achieving 10^{-3} grad-norm in 200 iterations)
- Conservative is better than aggresive β

How to Tune β (without checking plots)?

Most blogs suggest $\beta = 0.9$ or 0.99. Why?

- condition number is often between 10^2 to 10^4, corresponding to an optimal $\beta \approx (1 - \frac{1}{\sqrt{\kappa}})^2 \in [0.8, 0.98]$
- **Low accuracy**: Even for some problems with $\kappa > 10^6$, we may only need low accuracy like 10^{-3} of 10^{-5}, thus the effective condition number is below 10^4 (recall the lecture on "effects of eigenvalues")
- **High accuracy** $< 10^{-6}$: may still try larger β, e.g. 0.9999

Tune β:

- **Method 1** (naive): try $\beta = 0.5, 0.7, 0.8, 0.9, 0.99$ and pick the best
- **Method 2** (suggest): try β s.t.

$$\frac{1}{1-\beta} \in \{10, 100, 300, 600, 1000, 2000\}$$

and pick the best.

What Else are Mysterious?

From the plots, we have obtained a few insights on tuning.

One more mystery:

Q1: why is there fluctuation at all?

Shouldn't we design methods that are "descent" algorithms?

Q2: Why do we allow non-descent algorithm here?

Next section.

Outline

Matrix Recursion

The objective $f(\mathbf{x}) = .5\mathbf{x}^T Q\mathbf{x}$. (ignore linear term)

HB method: $\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha\nabla f(\mathbf{x}^r) + \beta(\mathbf{x}^r - \mathbf{x}^{r-1})$

Together with $\mathbf{x}^r = \mathbf{x}^r$, we write a matrix form

$$\begin{pmatrix} \mathbf{x}^{r+1} \\ \mathbf{x}^r \end{pmatrix} = \begin{pmatrix} I - \alpha Q + \beta I & -\beta I \\ I & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}^r \\ \mathbf{x}^{r-1} \end{pmatrix} \tag{1}$$

Write $\mathbf{y}^r \triangleq \begin{pmatrix} \mathbf{x}^r \\ \mathbf{x}^{r-1} \end{pmatrix}$, $M = \begin{pmatrix} I - \alpha Q + \beta I & -\beta I \\ I & 0 \end{pmatrix}$
then we have

$$\mathbf{y}^{r+1} = M\mathbf{y}^r.$$

The behavior of HB is determined by eigenvalues of matrix M

Compare HB v.s. GD

GD update equation for $f(\mathbf{x}) = .5\mathbf{x}^T Q\mathbf{x}$:

$$\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha\nabla f(\mathbf{x}^r) = (I - \alpha Q)\mathbf{x}^r$$

$$\mathbf{x}^{r+1} = M_{\mathrm{GD}}\mathbf{x}^r \text{ where } M_{\mathrm{GD}} = I - \alpha Q. \tag{2}$$

HB update equation $\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha\nabla f(\mathbf{x}^r) + \beta(\mathbf{x}^r - \mathbf{x}^{r-1})$.
Write as

$$\mathbf{y}^{r+1} = M_{\mathrm{HB}}\mathbf{y}^r \text{ where } M_{\mathrm{HB}} = \begin{pmatrix} I - \alpha Q + \beta I & -\beta I \\ I & 0 \end{pmatrix} \tag{3}$$

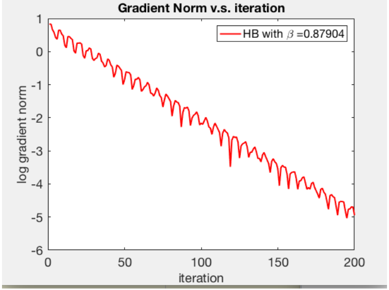
Major difference:
M_{GD} is a symmetric matrix; M_{HB} is a non-symmetric matrix.

Eigenvalues of Update Matrix M

Consider the case $\beta = 0.879$ (optimal β).

Eigenvalues of M are
$(0.44 \pm 0.82i,\ 0.82 \pm 0.44i,\ 0.88 \pm 0.33i,\ ,0.92 \pm 0.19i, 0.9376, 0.9376)$

Magnitude of eigenvalues of M are
$(0.9376, 0.9376, ..., 0.9376, 0.9376, 0.9376,\ 0.9376)$



How to explain the figure using the eigenvalues?

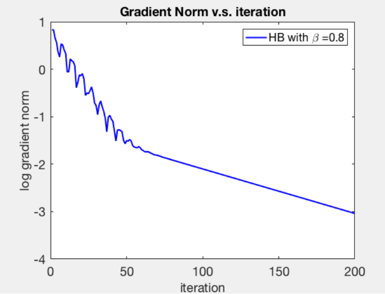
- 1 stage(s): $|\lambda_i| = 0.9376, \forall i$
- Fluctuation: due to the phase of eigenvalues (like cosine curve)

Eigenvalues of Update Matrix M

$\beta = 0.8$.

Eigenvalues of M are
$(0.4 \pm 0.8i,\ 0.79 \pm 0.43i,\ 0.84 \pm 0.31i,\ ,0.88 \pm 0.17i, 0.82\ , 0.98)$

Absolute eigenvalues of M are $(0.89, 0.89, ..., 0.89, 0.89, 0.82,\ 0.98)$

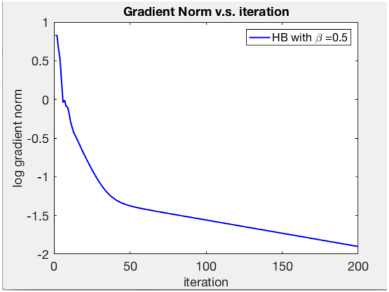


How to explain the figure using the eigenvalues?

- 3 stages: due to 3 clusters of eigenvalues (only considering magnitude)
- Fluctuation: only in the 2nd stage (since 0.89 corresponds to complex eigenvalues; while $0.82, 0.89$ correspond to real

Eigenvalues of Update Matrix M

$\beta = 0.5$.



Eigenvalues of M are
$(0.25 \pm 0.66i,\ 0.64 \pm 0.31i,\ 069 \pm 0.16i, 0.50,\ 0.55\ , 0.91, 0.99)$

Absolute eigenvalues of M are

$$(0.71, ..., 0.71, 0.71, 0.50\ , 0.55,\ 0.91, 0.99)$$

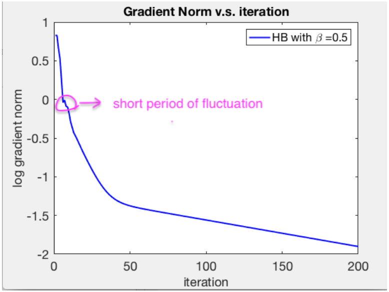
$$(\pm 69°, \quad \pm 26°, \quad \pm 13°, \quad 0, \quad 0, \quad 0, \quad 0)$$

Why little fluctuation?

-

Eigenvalues of Update Matrix M

$\beta = 0.5$.



Eigenvalues of M are

$(0.25 \pm 0.66i,\ 0.64 \pm 0.31i,\ 069 \pm 0.16i, 0.50,\ 0.55\ , 0.91, 0.99)$

Absolute eigenvalues of M are

$$(0.71, ..., 0.71, 0.71, 0.50\ , 0.55,\ 0.91, 0.99)$$

$$(\pm 69^\circ, \quad \pm 26^\circ, \quad \pm 13^\circ, \quad 0, \quad 0, \quad 0, \quad 0)$$

Why little fluctuation?

- That stage is too short

Nesterov's Accelerated Method

We solve the linear regression problem by Nesterov's accelerated method.

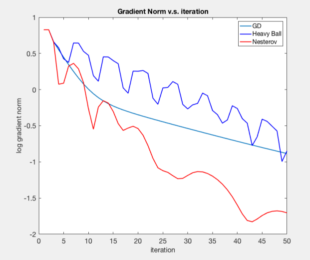
$$\begin{cases} y^r = x^r + \beta_r(x^r - x^{r-1}), \\ x^{r+1} = y^r - \alpha_r \nabla f(y^r). \end{cases} \tag{4}$$

Stepsize $\alpha = 1/L$.

Try different β. (at home)

Comparison: Nesterov momentum and momentum

Using Optimal parameters for Nesterov ($\beta = 0.8825$), and HB ($\beta = 0.879$)



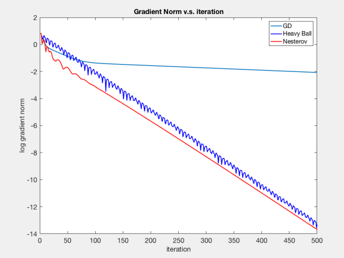


Figure: Left: 50 iterations (early stage); Right: 500 iterations (global picture)

Observations:

- Nesterov's method is faster than GD in early stage, and similar to HB in later stages
- Nesterov's method has less fluctuation than HB

Answers To Earlier Questions

Q1: why is there fluctuation at all?

- Because the update matrix has complex eigenvalues.

Q2: Why do we allow non-descent algorithm here?

Intuition: Assume $z = \rho e^{i\theta}$, where $\rho < 1$. Then Re$(z^r) =$ Re$(\rho^r(\cos(r\theta) + i\sin(r\theta))) = \rho^r \cos(r\theta)$ converges to zero at least at rate ρ, but fluctuation occurs due to $\cos$ part.

- A more formal analysis is not easy; skip in the course

Outline

Further Improvement of Momentum?

- Question: can we do better than HB and NAG?
  - "Better" can mean many things...
  - e.g. better per-iteration cost; better iteration complexity; ...
- **Question**: Can we improve the iteration complexity $\tilde{O}(\sqrt{\kappa})$, even just for quadratic case?
- E.g. (more history info): can we use three or four terms in history, to get $\tilde{O}(\kappa^{1/3})$ or even better bound?

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 go beyond the conditions for lower bound (more discussion later)

Oracle Model

- **Oracle model** Ω for the first order algorithms:
  - given any x, 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^r and $\nabla f(x^r), r = 0, 1, \ldots$
- **Definition**: The (iteration) complexity of algorithm $\mathcal{A} \in \Omega$, for a function f, is the minimal number of iterations to achieve error ϵ, i.e.,

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

- **Definition**: The complexity of algorithm $\mathcal{A} \in \Omega$, for a function class F, is thelargest minimal number of iterations to achieve error ϵ, i.e.,

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

What Algorithm is Covered?

What is covered by the oracle model 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 $P_n(D, L)$ be the class of smooth unconstrained convex optimization problems $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ with

$$\|x^0 - x^*\| \leq D,$$
$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

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

$$\nabla^2 f(\mathbf{x}) \succeq \mu I, \quad \forall \mathbf{x}.$$

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

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

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

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

- For any first-order method in Ω, there exists a strongly convex problem s.t. to get error $< \epsilon$, the required iteration is at least $\min\{n, \sqrt{\kappa}\log(1/2\epsilon)\}$

Limitation of Lower Bound

Is that the end?

Two big limitations:

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

Lower bounds are "negative", but not just negative.

They provide directions for new research: shall design algorithms that

- save per-iteration time! (SGD, CD, etc.)
- go beyond "pure" first-order methods (BFGS, BB, etc.)

Summary

Summary of this lecture (exercise).

29/30

Summary

How to tune β?

- Start from $\beta = 0.9, 0.95$ or 0.99
- Try $\frac{1}{1-\beta} \in \{10, 100, 300, 600, 1000, 2000\}$ and pick the best β
- Reminder: in practice, may need to tune α (till knife's edge) together

Write HB and NAG as matrix recursion, and check their eigenvalues

- Update matrix is non-symmetric; has complex eigenvalues, thus fluctuation
- This is done for quadratic problem

Even for quadratic problems, NAG is better than HB

- in early stage, as good as GD
- in late stage, as good as or slightly better than HB

NAG is the optimal method among (certain class of) first-order methods

- Reminder: HB is NOT! (for non-quadratic problem, its benefit may not exist)