# SDSC6015 Stochastic Optimization for Machine Learning

Lu Yu

Department of Data Science, City University of Hong Kong

October 9, 2025

# Momentum Methods

# Motivation

Consider minimizing the function $f \in \mathbb{R}^d \to \mathbb{R}$, we turn to SGD

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)$$

This method works well for smooth convex functions, but it struggles in situations where the function has elongated contours[1]!

[1] Anlogy: rolling the ball on a long, narrow hill-it's easy for the ball to move quickly in the flat direction but slow and harder to roll in the steep direction

# Heavy-Ball Method (Polyak's Momentum)

Polyak's momentum, also known as the "heavy ball method", introduces a "momentum" term $\beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$. The update rule for momentum is

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t) + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) \,.$$

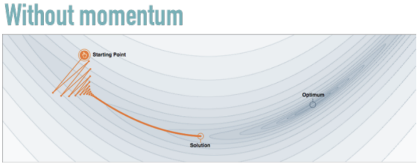
This is equivalent to

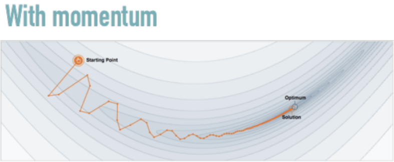
$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{x}_t + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) & \quad \text{momentum step} \\ \boldsymbol{x}_{t+1} &= \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) & \quad \text{gradient step} \end{aligned}$$

where $\beta_t$ is a hyperparameter (typically $\beta_t \in [0, 1]$), which scales down the previous step.

- This algorithm was first proposed in the 60s.
- It combines the current gradient with a history of the previous step to accelerate the convergence of the algorithm.
- It recovers gradient descent when $\beta_t = 0$.

Without momentum, gradient descent oscillates, whereas with momentum, we find that it converges much closer to the optimal point in the same number of iterations.



# Convergence of Heavy-Ball Method

Consider the strongly convex quadratic function:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top Q\boldsymbol{x} - \mathbf{b}^\top \boldsymbol{x}\,,$$

where $Q$ is a symmetric positive definite matrix, and $b$ is a vector.

- $\mu = \lambda_{\min}(Q)$ is the smallest eigenvalue of $Q$ (strong convexity constant)
- $L = \lambda_{\max}(Q)$ is the largest eigenvalue of $Q$ (smoothness constant)
- $\kappa = L/\mu > 1$ is the condition number of $Q$

# Convergence of Heavy-Ball Method

Comparison of the convergence rates between the heavy-ball method and gradient descent:

| Method | Step size | Momentum | Convergence rate |
|---|---|---|---|
| GD | $\eta_t = \frac{2}{\mu+L}$ | $\beta_t = 0$ | $\rho_{\mathsf{GD}} = 1 - \frac{2}{1+\kappa}$ |
| Heavy-Ball | $\eta_t = \frac{4}{(\sqrt{\mu}+\sqrt{L})^2}$ | $\beta_t = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ | $\rho_{\mathsf{HB}} = 1 - \frac{1}{\sqrt{\kappa}}$ |

- Heavy-Ball method converges faster than Gradient Descent.
- However, there exist strongly-convex and smooth functions for which, by choosing carefully the hyperparameters $\eta_t$ and $\beta_t$ and the initial condition $\boldsymbol{x}_0$, the heavy-ball method fails to converge.

# Counter Example

Consider piece-wise quadratic function $f$ [LRP16]

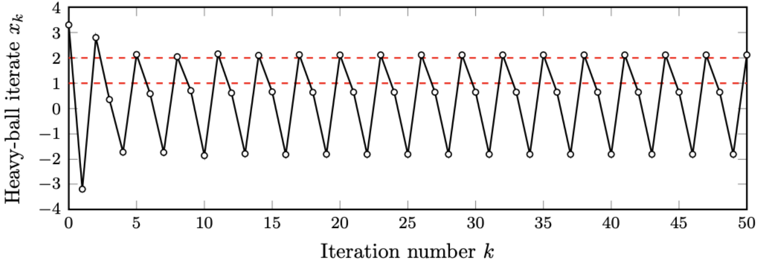
$$f(x) = \begin{cases} \frac{25}{2}x^2 & x < 1 \\ \frac{1}{2}x^2 + 24x - 12 & 1 \leqslant x < 2 \\ \frac{25}{2}x^2 - 24x + 36 & 2 \leqslant x \end{cases}$$

whose gradient is

$$\nabla f(x) = \begin{cases} 25x & x < 1 \\ x + 24 & 1 \leqslant x < 2 \\ 25x - 24 & 2 \leqslant x \end{cases}$$

By construction, $\forall x_1, x_2 \|\nabla f(x_1) - \nabla f(x_2)\| \leqslant 25\|x_1 - x_2\|$, therefore $f$ is 25-smooth, and $\nabla^2 f(x) \geqslant 1 > 0$, therefore $f$ is 1-strongly convex.

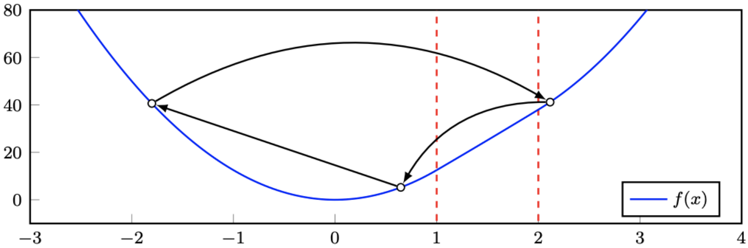
# Counter Example




- This figure from [LRP16] gives the first 50 iterates of Polyak's momentum algorithm applied to $f$, using $\eta_t = \frac{1}{9}, \beta_t = \frac{4}{9}$ and $x_0 = 3.3$.
- Despite the function $f$ being 1-strongly convex and 25-smooth, the output values of the heavy-ball method cycle through 3 points indefinitely.

# Counter Example

Illustration of the limit values of the failing case of Polyak's momentum algorithm.




There exists a sequence of iterates $\{x_t\}$ such that as $n \to \infty$

$$x_{t=3n} \to 0.65, \quad x_{t=3n+1} \to -1.80, \quad x_{t=3n+2} \to 2.12$$

# Failing case of Heavy-Ball Method

- It is worth pointing out that heavy-ball method has guaranteed convergence for quadratic functions (and not piece-wise quadratic).
- Discontinuous gradients may make the momentum term ineffective.

$$\nabla f(x) = \begin{cases} 25x & x < 1 \\ x + 24 & 1 \leqslant x < 2 \\ 25x - 24 & 2 \leqslant x \end{cases}$$

# Nesterov's Accelerated Gradient Descent

Heavy-ball method

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{x}_t + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) && \text{momentum step} \\ \boldsymbol{x}_{t+1} &= \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{x}_t) && \text{gradient step} \end{aligned}$$

Nesterov's Accelerated Gradient Descent (Nesterov's AGD)

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{x}_t + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) && \text{momentum step} \\ \boldsymbol{x}_{t+1} &= \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{y}_t) && \text{gradient step} \end{aligned}$$

As we see below, Nesterov's AGD enjoys convergence guarantees for (strongly) convex functions beyond quadratic functions!

# Nesterov's AGD on Strongly and Smooth Functions

Initialized at $\boldsymbol{x}_0$, set $\boldsymbol{x}_{-1} = \boldsymbol{x}_0$, the iterates of Nesterov's AGD for $t = 0, 1, \ldots, T$

$$
\begin{aligned}
\boldsymbol{y}_t &= \boldsymbol{x}_t + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) && \text{momentum step} \\
\boldsymbol{x}_{t+1} &= \boldsymbol{y}_t - \eta_t \nabla f(\boldsymbol{y}_t) && \text{gradient step}
\end{aligned}
$$

**Theorem 1**

For Nesterov's AGD Algorithm applied to $\mu$-strongly convex and $L$-smooth function $f$, we have

$$
f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leqslant \Big(1 - \sqrt{\frac{\mu}{L}}\Big)^T \frac{(L+\mu)\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{2}
$$

provided that

$$
\eta_t = \frac{1}{L}, \quad \beta_t = \frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1}.
$$

# Nesterov's AGD on Strongly and Smooth Functions

Proof.

Without loss of generality, assume $\boldsymbol{x}^* = \mathbf{0}$.[2] Set $\rho^2 := 1 - \frac{1}{\sqrt{\kappa}}$, with $\kappa = L/\mu$. Set $u_t := \frac{1}{L}\nabla f(\boldsymbol{y}_t)$ and

$$V_t := f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) + \frac{L}{2}\|\boldsymbol{x}_t - \rho^2 \boldsymbol{x}_{t-1}\|_2^2 \,.$$

The proof involves two steps

- Step 1: show that $V_{t+1} \leqslant \rho^2 V_t, \forall t \geqslant 0$
- Step 2: show that $f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leqslant \left(1 - \sqrt{\frac{\mu}{L}}\right)^T \frac{(L+\mu)\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{2}$

[2] This can be done by translation of coordinate.

# Nesterov's AGD on Strongly and Smooth Functions

Step 1: show that $V_{t+1} \leqslant \rho^2 V_t, \forall t \geqslant 0$

$$
\begin{aligned}
V_{t+1} &= f(\boldsymbol{x}_{t+1}) - f^* + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \rho^2 \boldsymbol{x}_t\|^2 \\
&\leqslant f(\boldsymbol{y}_t) - f^* + \langle L u_t, \boldsymbol{x}_{t+1} - \boldsymbol{y}_t \rangle + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{y}_t\|^2 + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \rho^2 \boldsymbol{x}_t\|^2 \\
&\leqslant f(\boldsymbol{y}_t) - f^* - \frac{L}{2}\|u_t\|^2 + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \rho^2 \boldsymbol{x}_t\|^2 \\
&= \rho^2 \Big[ f(\boldsymbol{y}_t) - f^* + L\langle u_t, \boldsymbol{x}_t - \boldsymbol{y}_t \rangle \Big] - \rho^2 L \langle u_t, \boldsymbol{x}_t - \boldsymbol{y}_t \rangle \\
&\qquad + (1-\rho^2)\Big[ f(\boldsymbol{y}_t) - f^* - L\langle u_t, \boldsymbol{y}_t \rangle \Big] + (1-\rho^2)\langle u_t, \boldsymbol{y}_t \rangle \\
&\qquad - \frac{L}{2}\|u_t\|^2 + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \rho^2 \boldsymbol{x}_t\|^2 .
\end{aligned}
$$

# Nesterov's AGD on Strongly and Smooth Functions

Step 1: show that $V_{t+1} \leqslant \rho^2 V_t, \forall t \geqslant 0$

By strong convexity of $f$

$$
\begin{aligned}
f(\boldsymbol{y}_t) + L\langle u_t, \boldsymbol{x}_t - \boldsymbol{y}_t\rangle &\leqslant f(\boldsymbol{x}_t) - \frac{\mu}{2}\|\boldsymbol{x}_t - \boldsymbol{y}_t\|^2 \\
f(\boldsymbol{y}_t) - f^* - L\langle u_t, \boldsymbol{y}_t\rangle &\leqslant -\frac{\mu}{2}\|\boldsymbol{y}_t\|^2
\end{aligned}
$$

Thus,

$$
\begin{aligned}
V_{t+1} \leqslant \rho^2\Big[f(\boldsymbol{x}_t) - f^* - \frac{\mu}{2}\|\boldsymbol{x}_t - \boldsymbol{y}_t\|^2\Big] &- \rho^2 L\langle u_t, \boldsymbol{x}_t - \boldsymbol{y}_t\rangle \\
&- (1-\rho^2)\frac{\mu}{2}\|\boldsymbol{y}_t\|^2 + (1-\rho^2)L\langle u_t, \boldsymbol{y}_t\rangle \\
&- \frac{L}{2}\|u_t\|^2 + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \rho^2\boldsymbol{x}_t\|^2 \\
= \rho^2 & \underbrace{\Big[f(\boldsymbol{x}_t) - f^* + \frac{L}{2}\|\boldsymbol{x}_t - \rho^2\boldsymbol{x}_{t-1}\|^2\Big]}_{V_t} + R_t
\end{aligned}
$$

# Nesterov's AGD on Strongly and Smooth Functions

Step 1: show that $V_{t+1} \leqslant \rho^2 V_t, \forall t \geqslant 0$

Plugging the definitions of $\eta_t, \beta_t, \rho, \boldsymbol{x}_{t+1}, \boldsymbol{y}_t$ into the definition of $R_t$

$$
\begin{aligned}
R_t := & -\rho^2 \frac{\mu}{2}\|\boldsymbol{x}_t - \boldsymbol{y}_t\|^2 - (1-\rho^2)\frac{\mu}{2}\|\boldsymbol{y}_t\|^2 \\
& + L\langle u_t, \boldsymbol{y}_t - \rho^2 \boldsymbol{x}_t\rangle - \frac{L}{2}\|u_t\|^2 \\
& + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \rho^2 \boldsymbol{x}_t\|^2 - \frac{\rho^2 L}{2}\|\boldsymbol{x}_t - \rho^2 \boldsymbol{x}_{t-1}\|^2 \\
= & -\frac{1}{2} L\rho^2 \Big(\frac{1}{\kappa} + \frac{1}{\sqrt{\kappa}}\Big)\|\boldsymbol{x}_t - \boldsymbol{y}_t\|^2 \leqslant 0
\end{aligned}
$$

Thus,

$$
V_{t+1} \leqslant \rho^2 V_t, \quad \forall t \geqslant 0\,.
$$

# Nesterov's AGD on Strongly and Smooth Functions

Step 2: show that $f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leqslant \left(1 - \sqrt{\frac{\mu}{L}}\right)^T \frac{(L+\mu)\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{2}$

By the definition of $V_t$

$$f(\boldsymbol{x}_t) - f^* \leqslant V_t \leqslant \rho^{2t} V_0$$

Moreover,

$$\begin{aligned}
V_0 &= f(\boldsymbol{x}_0) - f^* + \frac{L}{2}\|\boldsymbol{x}_0 - \rho^2 \boldsymbol{x}_0\|^2 \\
&= f(\boldsymbol{x}_0) - f^* + \frac{\mu}{2}\|\boldsymbol{x}_0\|^2 \\
&= f(\boldsymbol{x}_0) - f^* + \frac{\mu}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 \\
&\leqslant \frac{L}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + \frac{\mu}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 \\
&= \frac{L+\mu}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 .
\end{aligned}$$

Thus,

$$f(\boldsymbol{x}_T) - f^* \leqslant \left(1 - \sqrt{\frac{\mu}{L}}\right)^T \cdot \frac{L+\mu}{2} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 .$$

- Set $R^2 = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$.

$$f(\boldsymbol{x}_T) - f^* \leqslant \left(1 - \sqrt{\frac{\mu}{L}}\right)^T \cdot \frac{(L+\mu)R^2}{2}$$

- Gradient Descent on $\mu$-strongly convex and $L$-smooth functions[3]

$$f(\boldsymbol{x}_{\text{best}}^{(T)}) - f(\boldsymbol{x}^*) \leqslant \left(1 - \frac{\mu}{L}\right)^T \frac{RL}{2}$$

- Nesterov's AGD improves by a factor of $\sqrt{\kappa} = \sqrt{\frac{L}{\mu}}$

[3] Theorem 1 from Lecture 3

**Theorem 2**

For Nesterov's AGD Algorithm applied to convex and $L$-smooth function $f$, we have

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leqslant \frac{2L\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{T^2}$$

provided that

$$\eta_t = \frac{1}{L}, \quad \beta_t = \frac{\lambda_{t-1} - 1}{\lambda_t},$$

where

$$\lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2}, \ \lambda_0 = 0, \ \beta_0 = 0$$

[4]

[4] $\lambda_{t+1}^2 - \lambda_{t+1} = \lambda_t^2$

Proof.

By sufficient decrease (Lemma 3 from Lecture 2),

$$f(\boldsymbol{x}_{t+1}) \leqslant f(\boldsymbol{y}_t) - \frac{1}{2L}\|\nabla f(\boldsymbol{y}_t)\|^2 \leqslant f(\boldsymbol{y}_t)\,.$$

Therefore,

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) &= f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{y}_t) + f(\boldsymbol{y}_t) - f(\boldsymbol{x}_t) \\ &\leqslant -\frac{1}{2L}\|\nabla f(\boldsymbol{y}_t)\|^2 + \langle \nabla f(\boldsymbol{y}_t),\, \boldsymbol{y}_t - \boldsymbol{x}_t\rangle \\ &= -\frac{L}{2}\|\boldsymbol{y}_t - \boldsymbol{x}_{t+1}\|^2 + L\langle \boldsymbol{y}_t - \boldsymbol{x}_{t+1},\, \boldsymbol{y}_t - \boldsymbol{x}_t\rangle\,. \qquad (1) \end{aligned}$$

Similarly,

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) - f^* &= f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{y}_t) + f(\boldsymbol{y}_t) - f^* \\ &\leqslant -\frac{1}{2L}\|\nabla f(\boldsymbol{y}_t)\|^2 + \langle \nabla f(\boldsymbol{y}_t),\, \boldsymbol{y}_t - \boldsymbol{x}^*\rangle \\ &= -\frac{L}{2}\|\boldsymbol{y}_t - \boldsymbol{x}_{t+1}\| + L\langle \boldsymbol{y}_t - \boldsymbol{x}_{t+1},\, \boldsymbol{y}_t - \boldsymbol{x}^*\rangle \qquad (2) \end{aligned}$$

# Nesterov's AGD on Smooth and Convex Functions

Define the optimality gap $\Delta_t := f(\boldsymbol{x}_t) - f^*$. Taking (1)$\times\lambda_t(\lambda_t - 1)$ + (2)$\times\lambda_t$, we get

$$\begin{aligned}
&\lambda_t(\lambda_t - 1)(\Delta_{t+1} - \Delta_t) + \lambda_t \Delta_{t+1} \\
&\qquad \leqslant L\langle \boldsymbol{y}_t - \boldsymbol{x}_{t+1},\, \lambda_t(\lambda_t - 1)(\boldsymbol{y}_t - \boldsymbol{x}_t) + \lambda_t(\boldsymbol{y}_t - \boldsymbol{x}^*)\rangle - \frac{L}{2}\lambda_t^2 \|\boldsymbol{y}_t - \boldsymbol{x}_{t+1}\|^2 .
\end{aligned}$$

Rearranging terms gives

$$\begin{aligned}
&\lambda_t^2 \Delta_{t+1} - (\lambda_t^2 - \lambda_t)\Delta_t \\
&\qquad \leqslant \frac{L}{2} \cdot \left[ 2\langle \lambda_t(\boldsymbol{y}_t - \boldsymbol{x}_{t+1}),\, \lambda_t \boldsymbol{y}_t - (\lambda_t - 1)\boldsymbol{x}_t - \boldsymbol{x}^* \rangle - \|\lambda_t(\boldsymbol{y}_t - \boldsymbol{x}_{t+1})\|^2 \right]
\end{aligned}$$

# Nesterov's AGD on Smooth and Convex Functions

Using $\lambda_t^2 - \lambda_t = \lambda_{t-1}^2$ and $2\langle a,\, b\rangle - \|a\|^2 = \|b\|^2 - \|b-a\|^2$, we obtain

$$
\begin{aligned}
&\lambda_t^2 \Delta_{t+1} - \lambda_{t-1}^2 \Delta_t \\
&\leqslant \frac{L}{2} \cdot \left[ \|\lambda_t \boldsymbol{y}_t - (\lambda_t - 1)\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 - \|\lambda_t \boldsymbol{x}_{t+1} - (\lambda_t - 1)\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 \right] \\
&= \frac{L}{2} \cdot \left[ \|\lambda_t \boldsymbol{y}_t - (\lambda_t - 1)\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 - \|\lambda_{t+1} \boldsymbol{y}_{t+1} - (\lambda_{t+1} - 1)\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \right]
\end{aligned}
$$

Summing over $t = 0, \ldots, T$, and note that $\lambda_0 = 0, \lambda_1 = 1, \beta_1 = -1, \boldsymbol{y}_1 = \boldsymbol{x}_0$

$$
\lambda_T^2 \Delta_{T+1}^2 - \lambda_0^2 \Delta_1 = \lambda_T^2 \Delta_{T+1} \leqslant \frac{L}{2} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 .
$$

Finally, note that

$$\lambda_k \geqslant \frac{1+\sqrt{4\lambda_{k-1}^2}}{2} = \lambda_{k-1} + \frac{1}{2}$$

which, together with $\lambda_1 = 1$, implies $\lambda_T \geqslant \frac{T+1}{2}$. It follows that

$$f(\boldsymbol{x}_{T+1}) - f^* = \Delta_{T+1} \leqslant \frac{L\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{2\lambda_T^2} \leqslant \frac{2L\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{(T+1)^2}$$

# Nesterov's AGD on Smooth and Convex Functions

Thus, when $R^2 = \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$,

$$f(\boldsymbol{x}_{T+1}) - f(\boldsymbol{x}^*) \leqslant \frac{2LR^2}{T^2}$$

- Gradient Descent on convex and smooth function[5]

$$f(\boldsymbol{x}_{\text{best}}^{(T)}) - f(\boldsymbol{x}^*) \leqslant \frac{R^2 L}{2T}$$

- Significant improvement by Nesterov's AGD

[5] Theorem 2 from Lecture 2

Questions?

# SGD with Classical Momentum

Idea: include an additional weight $\beta \in [0, 1]$ which controls how much the update follows the current gradient versus past momentum.

The algorithm is defined over $t = 1, 2, \ldots$

$$
\begin{aligned}
\boldsymbol{g}_t &= \nabla f_{i_t}(\boldsymbol{x}_t) \\
\boldsymbol{v}_t &= \beta \boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t \\
\boldsymbol{x}_t &= \boldsymbol{x}_{t-1} - \boldsymbol{v}_t
\end{aligned}
$$

- A small $\beta$ favors the current gradient
- A large $\beta$ prioritizes previous movement.

# SGD with Classical Momentum

Idea:

$$\boldsymbol{v}_t = \beta \boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t$$
$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \boldsymbol{v}_t\,.$$

In practice, it's common to use two hyperparameters: $\beta$ affects the terminal velocity and $\eta$ is a learning rate.

$$\boldsymbol{v}_t = \beta \boldsymbol{v}_{t-1} + \eta \boldsymbol{g}_t$$
$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \boldsymbol{v}_t$$

# SGD with Nesterov Momentum

Key Difference Between Classical Momentum and Nesterov Momentum

- In classical momentum, we compute the gradient at the current position
- In Nesterov momentum, we first take a lookahead step based on momentum and then compute the gradient at this predicted next position.

The algorithm is defined for $t = 1, 2, \ldots$

$$\begin{aligned}
\boldsymbol{g}_t &= \nabla f_{i_t}(\boldsymbol{x}_{t-1} - \eta \boldsymbol{v}_{t-1}) \\
\boldsymbol{v}_t &= \beta \boldsymbol{v}_{t-1} + \eta \boldsymbol{g}_t \\
\boldsymbol{x}_t &= \boldsymbol{x}_{t-1} - \eta \boldsymbol{v}_t
\end{aligned}$$

# SGD with Momentum

SGD with momentum is used as a practical trick to speed up training, even though it lacks the theoretical guarantees...

# Adaptive Methods

# Adaptive Learning Rates

- So far, we've looked at update steps that look like

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{g}_t$$

- Here, the step size $\eta_t$ is fixed a priori for each iteration.
- What if we use a step size that varies depending on the model?
- This is the idea of an adaptive learning rate.

# Example: Polyak's Step Length

- This is a simple step size scheme for gradient descent that works when the optimal value is known.

$$\eta_t = \frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)}{\|\nabla f(\boldsymbol{x}_t)\|^2}$$

- Can also use this with an estimated optimal value.

# Intuition behind Polyak's Step Length

- Approximate the objective with a linear approximation at the current iterate.

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}_t) + (\boldsymbol{x} - \boldsymbol{x}_t)^\top \nabla f(\boldsymbol{x}_t)$$

- Choose the step size that makes the approximation equal to the known optimal value.

$$f(\boldsymbol{x}^*) = \hat{f}(\boldsymbol{x}_{t+1}) = \hat{f}(\boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t)) = f(\boldsymbol{x}_t) - \eta \|\nabla f(\boldsymbol{x}_t)\|^2$$

which implies

$$\eta = \frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)}{\|\nabla f(\boldsymbol{x}_t)\|^2}$$

# Example: Line Search

- Idea: just choose the step size that minimizes the objective.

$$\eta_t = \arg\min_{\eta > 0} f(\boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t))$$

- Only works well for gradient descent, not SGD.
  - SGD moves in random directions that don't always improve the objective.
  - Doing line search on full objective is expensive relative to SGD update.

# Adaptive Methods for SGD

Several methods exist

- AdaGrad (Adaptive Gradient Descent)
- RMSProp (Root Mean Squared Propagation)
- ADAM (AdaGrad with momentum)

## AdaGrad (Adaptive Gradient Descent)

- Main idea: use history of sampled gradients to choose the step size for the next SGD step to be inversely proportional to the usual magnitude of gradient steps in that direction.
- Adaptive subgradient methods for online learning and stochastic optimization
- J Duchi, E Hazan, Y Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, 2011

# AdaGrad

The standard AdaGrad algorithm updates each element of parameter $\boldsymbol{x}$ independently with an adaptive learning rate

$$\boldsymbol{x}_{t+1,i} = \boldsymbol{x}_{t,i} - \frac{\alpha}{\sqrt{G_{t,i}}}\boldsymbol{g}_{t,i}, \quad t = 1, 2, \ldots$$

where

- $\alpha > 0$, $\boldsymbol{g}_t := \nabla f_{i_t}(\boldsymbol{x}_t), i_t \in \{1, 2, \ldots, n\}$ are uniformly sampled at random
- $\boldsymbol{x}_{t,i}$ denotes the $i$-th element of the iterate $\boldsymbol{x}_t$
- $G_{t,i}$ accumulates squared gradients for each element $i$ separately

$$G_{t,i} = \sum_{j=1}^{t} \boldsymbol{g}_{j,i}^2$$

- Each element of $\boldsymbol{x}_t$ has its own adaptive learning rate

# AdaGrad (Scalar Version)

In the scalar version, we use a single scalar learning rate for all parameters:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \frac{\alpha}{\sqrt{G_t}}\boldsymbol{g}_t, \quad t = 1, 2, \ldots$$

where

- $\alpha > 0$, $\boldsymbol{g}_t := \nabla f_{i_t}(\boldsymbol{x}_t), i_t \in \{1, 2, \ldots, n\}$ are uniformly sampled at random
- $G_t$ is the global sum of squared gradients across all dimensions.

$$G_t = \sum_{j=1}^{t} \|\boldsymbol{g}_j\|^2$$

- The same scaling factor $\frac{\alpha}{\sqrt{G_t}}$ is applied to all elements of $\boldsymbol{x}_t$ uniformly

# Key Differences Between Standard and Scalar AdaGrad

| Feature | Standard AdaGrad | Scalar AdaGrad |
|---|---|---|
| Learning Rate | Element-wise adaptive | Single global adaptive |
| $G_t$ | $G_{t,i} = \sum_{j=1}^{t} \boldsymbol{g}_{j,i}^2$(element-wise) | $G_t = \sum_{j=1}^{t} \|\boldsymbol{g}_j\|^2$(global) |
| Use Case | sparse and non-uniform gradient | simpler but less adaptive |

Why Use Scalar AdaGrad:

✔ Computationally cheaper (avoids per-element storage of $G_t$).

✔ Still provides adaptive step size decay without tracking gradients individually.

✘ Less adaptive than standard AdaGrad, making it less useful for problems with sparse features.

# Convergence of AdaGrad on Convex Functions

For simplicity, we will focus on the nonstochastic and scalar version of AdaGrad[6]

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \frac{\alpha}{\sqrt{G_t}} \nabla f(\boldsymbol{x}_t), \quad t = 1, 2, \ldots$$

where $G_t = \sum_{j=1}^{t} \|\nabla f(\boldsymbol{x}_j)\|^2$.

**Theorem 3**

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable and let $\boldsymbol{x}^*$ be the unique global minimum of $f$. Assume that $\|\nabla f(\boldsymbol{x}_t)\| \leqslant L$. Scalar AdaGrad with $\alpha = R$ yields

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t\right) - f(\boldsymbol{x}^*) \leqslant \frac{3RL}{2\sqrt{T}}$$

where $R = \max_{t=1}^{T} \|\boldsymbol{x}_t - \boldsymbol{x}^*\|$.

[6] The analysis in the presence of stochastic gradients is analogous.

# Convergence of AdaGrad on Convex Functions

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t\right) - f(\boldsymbol{x}^*) \leqslant \frac{3RL}{2\sqrt{T}}$$

- We implicitly assume that the domain has bounded diameter and we know an upper bound $R$ on the diameter.
- The convergence rate is the same as SGD on convex and $L$-Lipschitz functions.

# Convergence of AdaGrad on Convex Functions

Proof. Let $\eta_t = \frac{R}{\sqrt{G_t}}$. From Lecture 4,

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)$$

is equivalent to

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \underbrace{\left\{ \nabla f(\boldsymbol{x}_t)^\top (\boldsymbol{x} - \boldsymbol{x}_t) + \frac{1}{2\eta_t} \|\boldsymbol{x} - \boldsymbol{x}_t\|^2 \right\}}_{\phi(\boldsymbol{x})}$$

The first-order optimality condition (Lemma 8 from Lecture 1) gives

$$\begin{aligned}
\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}^* \rangle &\leqslant \frac{1}{\eta_t} \langle \boldsymbol{x}_t - \boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1} - \boldsymbol{x}^* \rangle \\
&= \frac{1}{2\eta_t} \left( \|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2 \right)
\end{aligned}$$

Thus,

$$
\begin{aligned}
\langle \nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_t - \boldsymbol{x}^* \rangle &= \langle \nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_{t+1} - \boldsymbol{x}^* \rangle - \langle \nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \rangle \\
&\leqslant \frac{1}{2\eta_t}\Big( \|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \Big) \\
&\qquad - \langle \nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \rangle - \frac{1}{2\eta_t}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2 \\
&\leqslant \frac{1}{2\eta_t}\Big( \|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \Big) \\
&\qquad + \|\nabla f(\boldsymbol{x}_t)\| \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\| - \frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\leqslant \frac{1}{2\eta_t}\Big( \|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \Big) + \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_t)\|^2
\end{aligned}
$$

Summing up and collecting terms

$$\sum_{t=1}^{T}\langle\nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_t-\boldsymbol{x}^*\rangle \leqslant \sum_{t=2}^{T}\left(\frac{1}{2\eta_t}-\frac{1}{2\eta_{t-1}}\right)\underbrace{\|\boldsymbol{x}_t-\boldsymbol{x}^*\|^2}_{\leqslant R^2}+\frac{1}{2\eta_t}\underbrace{\|\boldsymbol{x}_2-\boldsymbol{x}^*\|^2}_{\leq R^2}$$

$$+\sum_{t=1}^{T}\frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_t)\|^2$$

$$\leqslant \frac{R^2}{2\eta_T}+\sum_{t=1}^{T}\frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_t)\|^2$$

# Convergence of AdaGrad on Convex Functions

$$\sum_{t=1}^{T} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}^* \rangle \leqslant \frac{R^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\nabla f(\boldsymbol{x}_t)\|^2$$

Recall the update rule for the step sizes

$$\frac{R^2}{\eta_T} = R\sqrt{\sum_{t=1}^{T} \|\nabla f(\boldsymbol{x}_t)\|^2} \qquad \sum_{t=1}^{T} \eta_t \|\nabla f(\boldsymbol{x}_t)\|^2 = R \sum_{t=1}^{T} \frac{\|\nabla f(\boldsymbol{x}_t)\|^2}{\sqrt{\sum_{i=1}^{t} \|\nabla f(\boldsymbol{x}_i)\|^2}}$$

Lemma: For any positive number $a_1, \ldots, a_T$, we have

$$\sqrt{\sum_{t=1}^{T} a_t} \leqslant \sum_{t=1}^{T} \frac{a_t}{\sqrt{\sum_{s=1}^{t} a_s}} \leqslant 2\sqrt{\sum_{t=1}^{T} a_t}$$

Using the inequality, we obtain

$$\sum_{t=1}^{T} \frac{\|\nabla f(\boldsymbol{x}_t)\|^2}{\sqrt{\sum_{i=1}^{t} \|\nabla f(\boldsymbol{x}_i)\|^2}} \leqslant 2\sqrt{\sum_{t=1}^{T} \|\nabla f(\boldsymbol{x}_t)\|^2}$$

# Convergence of AdaGrad on Convex Functions

Using the bounded gradient assumption

$$\sum_{t=1}^{T} \langle \nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_t - \boldsymbol{x}^* \rangle \leqslant \frac{3}{2} RL\sqrt{T}$$

Hence,

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t\right) - f(\boldsymbol{x}^*) \leq \frac{1}{T}\sum_{t=1}^{T} \langle \nabla f(\boldsymbol{x}_t),\, \boldsymbol{x}_t - \boldsymbol{x}^* \rangle \leqslant \frac{3RL}{2\sqrt{T}}$$

# Convergence of AdaGrad on Smooth Functions

**Theorem 4**

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex, $L$-smooth and differentiable and let $\boldsymbol{x}^*$ be the unique global minimum of $f$. Scalar AdaGrad with $\alpha = R$ yields

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t\right) - f(\boldsymbol{x}^*) \leqslant \frac{2R^2L}{T}$$

where $R = \max_{t=1}^{T} \|\boldsymbol{x}_t - \boldsymbol{x}^*\|$.

The proof is based on the proof of Theorem 3, which is left as a homework exercise.

# Convergence of AdaGrad on Smooth Functions

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t\right) - f(\boldsymbol{x}^*) \leqslant \frac{2R^2L}{T}$$

- The convergence rate is the same as SGD on convex and $L$-smooth functions.
- Usually, AdaGrad performs better than SGD in sparse optimization problems[7], eg. Lasso regression (Lecture 4)

[7] See motivating example from [Duchi et.al. 2011]

**RMSProp (Root Mean Squared Propagation)**

- Main idea: replacing the gradient accumulation of AdaGrad with an exponential moving average.
- Introduced by Geoffrey Hinton in his lecture on neural networks.

# RMSProp

$$\boldsymbol{x}_{t+1,i} = \boldsymbol{x}_{t,i} - \frac{\alpha}{\sqrt{G_{t,i}}}\boldsymbol{g}_{t,i}, \qquad t = 1, 2, \ldots$$

where

- $\alpha > 0$, $\boldsymbol{g}_t := \nabla f_{i_t}(\boldsymbol{x}_t), i_t \in \{1, 2, \ldots, n\}$ are uniformly sampled at random
- $G_{t,i}$ uses an exponentially decaying average

$$G_{t,i} = \sum_{j=1}^{t} \beta^{j-1}(1-\beta)\boldsymbol{g}_{j,i}^2$$

where $\beta \in (0, 1]$ is the decay factor.

# RMSProp

Key Differences from AdaGrad:

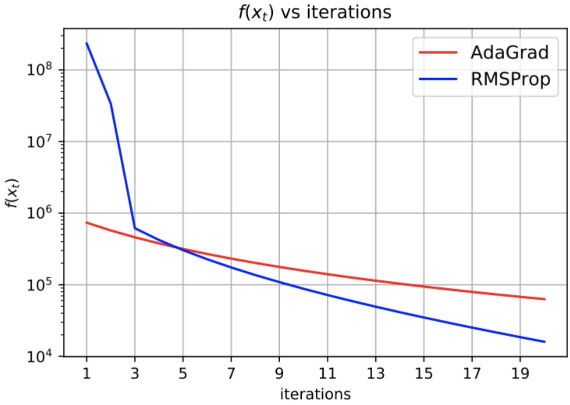
- AdaGrad accumulates squared gradients over time, which can lead to very small learning rates.
- RMSProp uses an exponentially decaying average, preventing the learning rate from shrinking too much.

Result: RMSProp maintains a more stable and effective learning rate throughout training.

# Example: AdaGrad vs. RMSProp

Setting:

- $f(x) = x^4$ (one-dimensional function)
- $x_0 = 10, x^* = 0.$



# Questions?

- The midterm exam will be held in class on October 16, 2025, from 7:00 PM to 9:30 PM.

- It will cover material from Lecture 1 to Lecture 6 (page 31), including content from the lecture slides, notes, and homework assignments.

- You are allowed to use an double-sided A4 cheat sheet.

- Exam questions are conceptual/theoretical; no coding.
- The exam includes true/false questions, multiple-choice questions, and theoretical questions.
- The theoretical questions will require short proofs, similar to those in Assignment 1.

**In-person Office Hours for Midterm Exam**

October 15, 4:00PM–5:30 PM

LAU 16/279.

# References


- Yudong Chen, *Uw-madison cs/isye/math/stat 726 lecture 9-10*, Spring 2023.
- John Duchi, Elad Hazan, and Yoram Singer, *Adaptive subgradient methods for online learning and stochastic optimization.*, Journal of machine learning research **12** (2011), no. 7.
- Mor Harchol-Balter, *Introduction to probability for computing*, Cambridge University Press, 2023.
- Laurent Lessard, Benjamin Recht, and Andrew Packard, *Analysis and design of optimization algorithms via integral quadratic constraints*, SIAM Journal on Optimization **26** (2016), no. 1, 57–95.
- Arkadij Semenovič Nemirovskij and David Borisovich Yudin, *Problem complexity and method efficiency in optimization.*
- Stephen J Wright and Benjamin Recht, *Optimization for data analysis*, Cambridge University Press, 2022.