SDSC6015 Stochastic Optimization for Machine Learning

Lu Yu

Department of Data Science, City University of Hong Kong

October 2, 2025

Mirror Descent

Mirror Descent: Motivation

Consider the simplex-constrained optimization problem

$$\min_{\boldsymbol{x} \in \triangle_d} f(\boldsymbol{x}),$$

where the simplex $\triangle_d := \{\boldsymbol{x} \in \mathbb{R}^d : \sum_{i=1}^d x_i = 1, x_i \geqslant 0,\ \forall i\}$

Now, we assume $\|\nabla f(\boldsymbol{x})\|_\infty = \max_{i=1,\dots,d} |[\nabla f(\boldsymbol{x})]_i| \leqslant 1, \forall \boldsymbol{x} \in \triangle_d$.

- The largest element of any gradient is bounded by 1.
- All the elements of any gradient are bounded by 1.
- The extreme cases here are the following two vectors taken as the gradient

$$\text{(the minimal vector) } \mathbf{0}_d = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \text{(the maximal vector) } \mathbf{1}_d = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Mirror Descent: Motivation

- For the vector $\mathbf{1}_d$, it has ℓ_2-norm $\|\mathbf{1}_d\|_2 = \sqrt{d}$
- In other words, $\|\nabla f(\boldsymbol{x})\|_\infty \leqslant 1$ gives $\|\nabla f(\boldsymbol{x})\|_2 \leqslant L = \sqrt{d}$
- Convergence of GD (on convex and L-Lipschitz functions):

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

- It turns out the rate $\mathcal{O}\big(\sqrt{\frac{d}{T}}\big)$ is not optimal, mirror descent can do better as $\mathcal{O}\big(\sqrt{\frac{\log d}{T}}\big)$

Mirror Descent: Preliminary

- Fix an arbitrary norm $\|\cdot\|$ on $\mathbb{R}^d$, and a compact convex set $X \subseteq \mathbb{R}^d$. The dual norm $\|\cdot\|_*$ is defined as

$$\|\boldsymbol{g}\|_* = \sup_{\|\boldsymbol{x}\| \leqslant 1} \boldsymbol{g}^\top \boldsymbol{x}.$$

- We say that a convex function $f : X \to \mathbb{R}^d$ is
  - L-Lipschitz w.r.t. $\|\cdot\|$ if $\forall x \in X, \boldsymbol{g} \in \partial f(\boldsymbol{x}), \|\boldsymbol{g}\|_* \leqslant L$
  - β-smooth w.r.t. $\|\cdot\|$ if $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_* \leqslant \beta\|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in X$
  - μ-strongly convex w.r.t. $\|\cdot\|$ if $f(\boldsymbol{x}) - f(\boldsymbol{y}) \leqslant \boldsymbol{g}^\top(\boldsymbol{x} - \boldsymbol{y}) - \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \forall \boldsymbol{x}, \boldsymbol{y} \in X, \boldsymbol{g} \in \partial f(\boldsymbol{x})$

Mirror Descent

Consider the mirror descent [Nemirovski and Yudin (1983)] iteration

$$\boldsymbol{y}_{t+1} = (\nabla\Phi)^{-1}(\nabla\Phi(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t) \quad \text{and} \quad \boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in X} D_\Phi(\boldsymbol{x}, \boldsymbol{y}_{t+1})\,,$$

with $\boldsymbol{g}_t \in \partial f(\boldsymbol{x}_t)$.

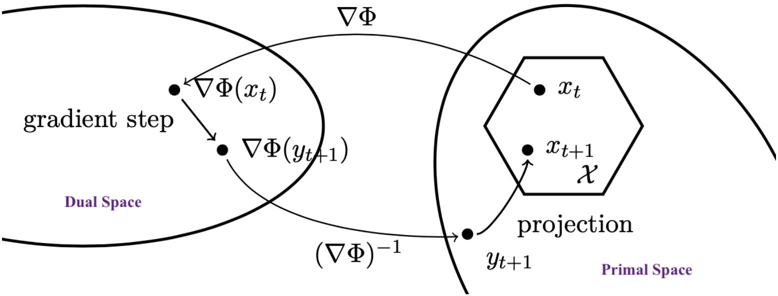


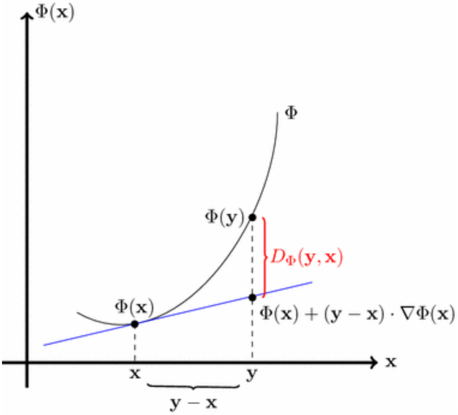
Figure: The "geometry" of mirror descent from [Bubeck 2015].

Mirror Descent: Key elements

- Mirror potential $\Phi : \mathbb{R}^d \to \mathbb{R}$ is strictly convex, continuously differentiable with $\lim_{\|\boldsymbol{x}\|_2 \to \infty} \|\nabla \Phi(x)\| = \infty$.

- Define the Bregman divergence associated to Φ as

$$D_\Phi(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{x}) - \Phi(\boldsymbol{y}) - \nabla \Phi(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) \,.$$



Mirror Descent: Key elements

- The projection via Bregman divergence associated to Φ

$$\Pi_X^\Phi(\boldsymbol{y}) = \arg\min_{\boldsymbol{x} \in X} D_\Phi(\boldsymbol{x}, \boldsymbol{y}), \quad \forall \boldsymbol{y} \in X\,.$$

- Properties of Φ ensures the existence and uniqueness of this projection Π_X^Φ.

Let $\boldsymbol{x}_1 \in \arg\min_{\boldsymbol{x} \in X} \Phi(\boldsymbol{x})$. For $t \geqslant 1$, let $\boldsymbol{y}_{t+1} \in \mathbb{R}^d$ such that

$$\nabla \Phi(\boldsymbol{y}_{t+1}) = \nabla \Phi(\boldsymbol{x}_t) - \eta \boldsymbol{g}_t, \text{ where } \boldsymbol{g}_t \in \partial f(\boldsymbol{x}_t)\,,$$

and

$$\boldsymbol{x}_{t+1} \in \Pi_X^{\Phi}(\boldsymbol{y}_{t+1})\,.$$

Theorem 1

Let

- Φ be a mirror map ρ-strongly convex on X w.r.t $\|\cdot\|$.
- $R^2 = \sup_{\boldsymbol{x} \in X} \Phi(\boldsymbol{x}) - \Phi(\boldsymbol{x}_1)$.
- f be convex and L-Lipschitz w.r.t $\|\cdot\|$.

Mirror descent with $\eta = \frac{2R}{L}\sqrt{\frac{\rho}{T}}$ satisfies

$$f\Big(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t\Big) - f(\boldsymbol{x}^*) \leqslant RL\sqrt{\frac{1}{\rho T}}\,.$$

Standard Setups for Mirror Descent

- **"Ball setup".** The mirror potential

$$\Phi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

  - Associated Bregman divergence $D_\Phi(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$.
  - This is exactly equivalent to the projected subgradient descent.

Standard Setups for Mirror Descent

- **"Simplex setup".** The mirror potential

$$\Phi(\boldsymbol{x}) = \sum_{i=1}^{d} x_i \log(x_i), \quad \boldsymbol{x} \in \mathbb{R}^d_{++} = \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, i = 1, \ldots, d\}.$$

  - When $\boldsymbol{x}, \boldsymbol{y} \in \triangle_d \cap \mathbb{R}^d_{++}$, the Bregman divergence of Φ is

$$D_\Phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} x_i \log(x_i / y_i) \quad \text{(Kullback-Leibler divergence)}$$

  - Projection of $\boldsymbol{y}$ onto the simplex $\triangle_d$ under the KL divergence leads to renormalization $\boldsymbol{y} \to \boldsymbol{y}/\|\boldsymbol{y}\|_1$ (see notes).
  - For $X = \triangle_d, \boldsymbol{x}_1 = (1/d, \ldots, 1/d)$ and $R^2 = \log d$ (see notes).

Questions?

Stochastic Gradient Descent

Main problem

Consider

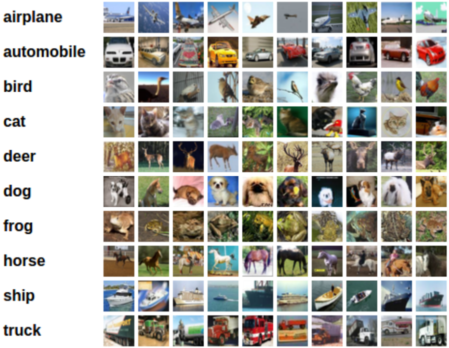
$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{\theta}),$$

where (usually)

$$f(\boldsymbol{\theta}) := \int \ell(\boldsymbol{\theta}, Z) dP(Z)$$

- $Z \in \mathbb{R}^p$
- $P(Z)$ is an unknown distribution
- $\ell(\boldsymbol{\theta}, Z)$ is the loss function parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$

Main problem



Given a set of labeled training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$, find the set of weights $\boldsymbol{\theta}$ which classifies the data via

$$\text{minimize } f(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{\theta}, (\boldsymbol{x}_i, y_i)) =: \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta}), \quad n \text{ large.}$$

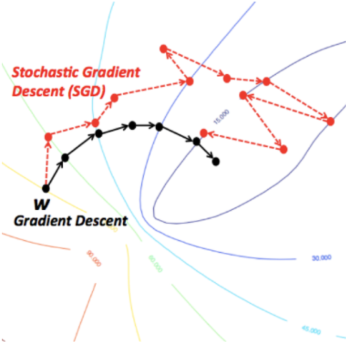
Motivation

- $f_i(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}, (\boldsymbol{x}_i, y_i))$ is the cost function of the i-th observation, taken from a training set of n observation.
- When $n \gg 1,000,000$, computing a single gradient $\nabla f(\boldsymbol{\theta}) = \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta})$ becomes too costly.
- Cheaper to compute gradient in a single component $\nabla f_i(\boldsymbol{\theta})$ (or "batch" of components).

Stochastic Gradient Descent

Set initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$. For $t = 0, 1, \ldots$:

- sample $i_t \in \{1, 2, \ldots, n\}$ uniformly at random.
- $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f_{i_t}(\boldsymbol{x}_t)$.



Only update with the gradient of f_{i_t} instead of the full gradient!
Iteration is n times cheaper than in full gradient descent.
The vector $\boldsymbol{g}_t := \nabla f_{i_t}(\boldsymbol{x}_t)$ is called a stochastic gradient.

Set initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$. For $t = 0, 1, \ldots$:
1. Draw a random index $i_t \in \{1, 2, \ldots, n\}$
2. Compute

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

Using a single component does not necessarily lead to convergence!

Consider the problem

$$\underset{x \in \mathbb{R}}{\text{minimize}}\ \frac{1}{2}\big(f_1(x) + f_2(x)\big)\,,$$

with

$$f_1(x) = 2x^2, \quad f_2(x) = -x^2\,.$$

Starting from $x_k > 0$, drawing $i_k = 2$ will necessarily lead to an increase in the function value.

But it can however produce descent in expectation...

Unbiasedness

- Can't use convexity

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \leqslant \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$$

on top of the vanilla analysis, as this may hold or not hold, depending on how the stochastic gradient $\boldsymbol{g}_t$ turns out.

- We will show (and exploit): the inequality holds in expectation.
- For this, we use that by definition, $\boldsymbol{g}_t$ is an unbiased estimate of $\nabla f(\boldsymbol{x}_t)$:

$$\mathbb{E}[\boldsymbol{g}_t | \boldsymbol{x}_t = \boldsymbol{x}] = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{x}) = \nabla f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d$$

This property ensures that, on average, our update direction is correct, despite the randomness in each individual step.

Conditional Expectation and Its Properties

Definition: Let X and Y be random variables. The conditional expectation of X given Y, denoted as

$$\mathbb{E}[X|Y],$$

is the best approximation of X using only information about Y.

- Instead of computing $\mathbb{E}[X]$, which gives an overall average, we compute $\mathbb{E}[X|Y]$, which gives the average value of X when we already know Y.
- For $Z = X + Y$, it holds that

$$\mathbb{E}[Z|X = x] = \mathbb{E}[X + Y|X = x] = x + \mathbb{E}[Y|X = x]$$

- For $Z = XY$, it holds that

$$\mathbb{E}[Z|X = x] = \mathbb{E}[XY|X = x] = x\mathbb{E}[Y|X = x]$$

Conditional Expectation and Its Properties

Definition: Let X and Y be random variables. The conditional expectation of X given Y, denoted as

$$\mathbb{E}[X|Y],$$

is the best approximation of X using only information about Y.

- (Partition Theorem)[1] Given a discrete random variable Y, it holds that

$$\mathbb{E}[X] = \sum_y \mathbb{E}[X|Y = y]\mathsf{Pr}(Y = y).$$

[1] The formal definition of conditional expectation and its related properties are provided in the book [HB23]

Convexity in Expectation

- For any fixed $\boldsymbol{x}$, linearity of conditional expectations yields

$$\mathbb{E}[\boldsymbol{g}_t^\top(\boldsymbol{x}-\boldsymbol{x}^*)|\boldsymbol{x}_t=\boldsymbol{x}] = \mathbb{E}[\boldsymbol{g}_t|\boldsymbol{x}_t=\boldsymbol{x}]^\top(\boldsymbol{x}-\boldsymbol{x}^*) = \nabla f(\boldsymbol{x})^\top(\boldsymbol{x}-\boldsymbol{x}^*)\,.$$

- Event $\{\boldsymbol{x}_t=\boldsymbol{x}\}$ can occur only for $\boldsymbol{x}$ in some finite set X ($\boldsymbol{x}_t$ is determined by the choices of indices in all iterations so far). Partition Theorem gives

$$\begin{aligned}\mathbb{E}[\boldsymbol{g}_t^\top(\boldsymbol{x}_t-\boldsymbol{x}^*)] &= \sum_{\boldsymbol{x}\in X}\mathbb{E}[\boldsymbol{g}_t^\top(\boldsymbol{x}-\boldsymbol{x}^*)|\boldsymbol{x}_t=\boldsymbol{x}]\mathsf{Pr}(\boldsymbol{x}_t=\boldsymbol{x})\\ &= \sum_{\boldsymbol{x}\in X}\nabla f(\boldsymbol{x})^\top(\boldsymbol{x}-\boldsymbol{x}^*)\mathsf{Pr}(\boldsymbol{x}_t=\boldsymbol{x})\\ &= \mathbb{E}[\nabla f(\boldsymbol{x}_t)^\top(\boldsymbol{x}_t-\boldsymbol{x}^*)]\end{aligned}$$

- Hence,

$$\mathbb{E}[\boldsymbol{g}_t^\top(\boldsymbol{x}_t-\boldsymbol{x}^*)] = \mathbb{E}[\nabla f(\boldsymbol{x}_t)^\top(\boldsymbol{x}_t-\boldsymbol{x}^*)] \geqslant \mathbb{E}[f(\boldsymbol{x}_t)-f(\boldsymbol{x}^*)]\,.$$

Stochastic Gradient Descent on Convex Functions

Theorem 2

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable, $\boldsymbol{x}^*$ a global minimum; furthermore, suppose that $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\| \leqslant R$ and that $\mathbb{E}[\|\boldsymbol{g}_t\|^2] \leqslant B^2$ for all t. Choosing the constant step size

$$\eta = \frac{R}{B\sqrt{T}},$$

stochastic gradient descent yields

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[f(\boldsymbol{x}_t)] - f(\boldsymbol{x}^*) \leqslant \frac{RB}{\sqrt{T}}.$$

Same procedure as every week...except

- we assume bounded stochastic gradients in expectation
- error bound holds in expectation

Stochastic Gradient Descent on Convex Functions

Proof.

Vanilla analysis (this time, $\boldsymbol{g}_t$ is the stochastic gradient):

$$\sum_{t=0}^{T-1} \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) \leqslant \frac{\eta}{2} \sum_{t=0}^{T-1} \|\boldsymbol{g}_t\|^2 + \frac{1}{2\eta} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 .$$

Taking expectations and using "convexity in expectation":

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{E}\big[f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)\big] &\leqslant \sum_{t=0}^{T-1} \mathbb{E}\big[\boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)\big] \\
&\leqslant \frac{\eta}{2} \sum_{t=0}^{T-1} \mathbb{E}[\|\boldsymbol{g}_t\|^2] + \frac{1}{2\eta} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 \\
&\leqslant \frac{\eta}{2} B^2 T + \frac{1}{2\eta} R^2 .
\end{aligned}$$

Result follows by optimizing η.

Convergence Rate Comparison: SGD v.s. GD

Classic GD: For vanilla analysis, we assumed that $\|\nabla f(\boldsymbol{x})\|^2 \leqslant B_{\mathsf{GD}}^2$ for all $\boldsymbol{x} \in \mathbb{R}^d$, where B_{GD} is a constant. So for sum-objective:

$$\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x})\right\|^2 \leqslant B_{\mathsf{GD}}^2, \quad \forall \boldsymbol{x}\,.$$

SGD: Assuming same for the expected squared norms of our stochastic gradients, now called B_{SGD}^2

$$\mathbb{E}[\|\boldsymbol{g}_t\|^2] = \mathbb{E}[\|\nabla f_{i_t}(\boldsymbol{x})\|^2] = \frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{x})\|^2 \leqslant B_{\mathsf{SGD}}^2, \quad \forall \boldsymbol{x}\,.$$

So by Jensen's inequality for $\|\cdot\|^2$

- $B_{\mathsf{GD}}^2 \approx \|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x})\|^2 \leqslant \frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{x})\|^2 \approx B_{\mathsf{SGD}}^2$.
- B_{GD}^2 can be smaller than B_{SGD}^2, but often comparable.

SGD on Strongly Convex Functions

Theorem 3

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and strongly convex with parameter $\mu > 0$, and $\boldsymbol{x}^*$ be the unique global minimum of f. With decreasing step size

$$\eta_t = \frac{2}{\mu(t+1)},$$

stochastic gradient descent yields

$$\mathbb{E}\Big[f\Big(\frac{2}{T(T+1)}\sum_{t=1}^{T} t \cdot \boldsymbol{x}_t\Big) - f(\boldsymbol{x}^*)\Big] \leqslant \frac{2B^2}{\mu(T+1)},$$

where $B^2 = \max_{t=1}^{T} \mathbb{E}[\|\boldsymbol{g}_t\|^2]$.

Almost same result as for subgradient descent, but in expectation.

SGD on Strongly Convex Functions

Proof.
Take expectations over vanilla analysis, before summing up (with varying stepsize η_t):

$$\mathbb{E}[\boldsymbol{g}_t^\top(\boldsymbol{x}_t - \boldsymbol{x}^*)] = \frac{\eta_t}{2}\mathbb{E}[\|\boldsymbol{g}_t\|^2] + \frac{1}{2\eta_t}\left(\mathbb{E}[\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2] - \mathbb{E}[\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2]\right).$$

"Strong convexity in expectation":

$$\mathbb{E}[\boldsymbol{g}_t^\top(\boldsymbol{x}_t - \boldsymbol{x}^*)] = \mathbb{E}[\nabla f(\boldsymbol{x}_t)^\top(\boldsymbol{x}_t - \boldsymbol{x}^*)] \geqslant \mathbb{E}[f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)] + \frac{\mu}{2}\mathbb{E}[\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2].$$

Putting it together with $\mathbb{E}[\|\boldsymbol{g}_t\|^2] \leqslant B^2$

$$\mathbb{E}[f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)] \leqslant \frac{B^2\eta_t}{2} + \frac{(\eta_t^{-1} - \mu)}{2}\mathbb{E}[\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2] - \frac{\eta_t^{-1}}{2}\mathbb{E}[\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2].$$

Proof continues as for subgradient descent, this time with expectations.

Mini-batch SGD

Instead of using a single element f_i, use an average of several of them:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \frac{1}{|S_t|} \sum_{i \in S_t} \nabla f_i(\boldsymbol{x}_t)\,,$$

where $S_t \subset \{1, 2, \ldots, n\}$ is drawn at random.

- S_t consists in a single index, recover the stochastic gradient descent algorithm
- $|S_t| = n$ and the n indices are drawn without replacement, then $S_t = \{1, \ldots, n\}$, recover the gradient descent algorithm
- $|S_t| = n_b \ll n$, called mini-batching. The resulting method is called mini-batch stochastic gradient.

Mini-batch SGD

Set initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$. For $t = 0, 1, \ldots$:

1. Draw a random subset $S_t \subset \{1, 2, \ldots, n\}$
2. Compute

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \frac{1}{|S_t|} \sum_{i \in S_t} \nabla f_i(\boldsymbol{x}_t)$$

Again, we are approximating full gradient by an unbiased estimate

$$\mathbb{E}\left[\frac{1}{|S_t|} \sum_{i \in S_t} \nabla f_i(\boldsymbol{x}) \right] = \nabla f(\boldsymbol{x}), \quad \forall \boldsymbol{x}\,.$$

Unified View

Consider the finite-sum optimization problem

$$\text{minimize } f(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta}) \, .$$

Definition. (Epoch) For problem of above, an epoch represents n calculations of a sample gradient ∇f_i.

- one iteration of gradient descent is an epoch
- n iterations of stochastic gradient descent on page 17 is an epoch
- n/n_b iterations of the mini-batch SGD (with a fixed batch size of n_b) is an epoch

Mini-batch SGD

Variance Intuition: Taking an average of many independent random variables reduces the variance. So for larger size of the mini-batch $m = |S_t|$, $\tilde{\boldsymbol{g}}_t = \frac{1}{|S_t|} \sum_{i \in S_t} \nabla f_i(\boldsymbol{x}_t)$ will be closer to the true gradient, in expectation:

$$
\begin{aligned}
\mathbb{E}\left[\|\tilde{\boldsymbol{g}}_t - \nabla f(\boldsymbol{x}_t)\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{m} \sum_{i \in S_t} \nabla f_i(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2\right] \\
&= \frac{1}{m} \mathbb{E}[\|\nabla f_1(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\|^2] \\
&= \frac{1}{m} \mathbb{E}[\|\nabla f_1(\boldsymbol{x}_t)\|^2] - \frac{1}{m} \|\nabla f(\boldsymbol{x}_t)\|^2 \leqslant \frac{B^2}{m}.
\end{aligned}
$$

Using a modification of the SGD analysis, can use this quantity to relate convergence rate to the rate of full gradient descent.

Stochastic Subgradient Descent

For problems which are not necessarily differentiable, we modify SGD to use a subgradient of f_i in each iteration. The update of stochastic subgradient descent is given by

- sampling $i \in \{1, 2, \ldots, n\}$ uniformly at random
- let $\boldsymbol{g}_t \in \partial f_i(\boldsymbol{x}_t)$
- $x_{t+1} = x_t - \eta_t \boldsymbol{g}_t$

In other words, we are using an unbiased estimate of a subgradient at each step, $\mathbb{E}[\boldsymbol{g}_t | \boldsymbol{x}_t] \in \partial f(\boldsymbol{x}_t)$.

Convergence in $\mathcal{O}(1/\varepsilon^2)$, by using the subgradient property at the beginning of the proof, where convexity was applied.

Constrained Optimization

For constrained optimization, our theorem for the SGD convergence in $\mathcal{O}(1/\varepsilon^2)$ steps directly extends to constrained problems as well.

After every step of SGD, projection back to X is applied as usual. The resulting algorithm is called projected SGD: select randomly an index $i_t \in \{1, \ldots, n\}$ at the t-th iteration.

$$\boldsymbol{y}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f_{i_t}(\boldsymbol{x}_t) \text{ and } \boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in X} \|\boldsymbol{x} - \boldsymbol{y}_{t+1}\|^2, t \geq 0,$$

Questions?

Momentum Methods

Motivation

Consider minimizing the function $f : \mathbb{R}^d \to \mathbb{R}$, 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[2]!

[2] Anology: 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

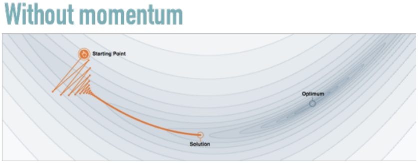
$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{x}_k + \beta_t(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) && \text{momentum step} \\ \boldsymbol{x}_{k+1} &= \boldsymbol{y}_k - \eta_t \nabla f(\boldsymbol{x}_k) && \text{gradient step} \end{aligned}$$

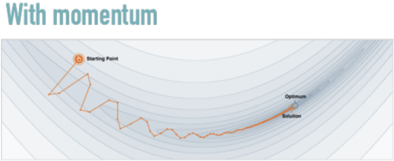
where β_t is a hyperparameter (typically $\beta_t \in [0, 1]$), which scales down the previous step.

- 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 $\beta_t = 0$.

Heavy-Ball Method

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

Theorem 4

Consider minimizing the quadratic function on the previous page. With the choice

$$\eta_t = \frac{4}{(\sqrt{\mu} + \sqrt{L})^2}, \quad \beta_t = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}},$$

the heavy-ball method converges at a linear rate[3]

$$\|\boldsymbol{x}_t - \boldsymbol{x}^*\| \leqslant \left(1 - \frac{1}{\sqrt{\kappa}}\right)^t \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|.$$

[3] Proof is provided in Chapter 4 of [WR22]

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 η_t and β_t and the initial condition $\boldsymbol{x}_0$, the heavy-ball method fails to converge.

Counter Example

Consider piece-wise quadratic function f [LRP16]

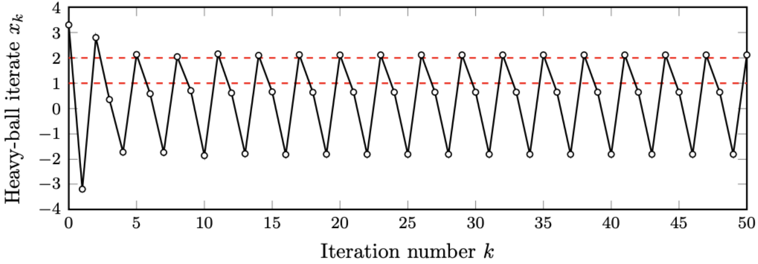
$$f(x) = \begin{cases} \frac{25}{2}x^2 & x < 1 \\ \frac{1}{2}x^2 + 24x - 12 & 1 \leqslant x < 2 \\ \frac{25}{2}x^2 - 24x + 36 & 2 \leqslant x \end{cases}$$

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.

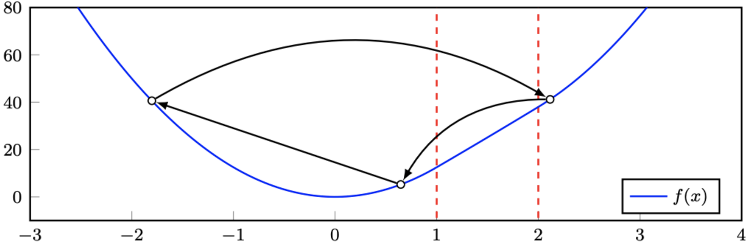
Counter Example



- 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

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

Nesterov's Accelerated Gradient Descent

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

Questions?

References

Sébastien Bubeck, *Convex optimization: Algorithms and complexity*, Foundations and Trends in Machine Learning **8** (2015), no. 3-4, 231–357.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra, *Efficient projections onto the l 1-ball for learning in high dimensions*, Proceedings of the 25th international conference on Machine learning, 2008, pp. 272–279.

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.