Lecture 9. Optimism for Acceleration

Advanced Optimization 2024 Fall

Peng Zhao
Nanjing University, Nanjing, 210023, China
zhaop@lamda.nju.edu.cn
February 10, 2025

1 Background: Accelerated Methods

Recall that accelerated convergence rates for smooth convex optimization can be achieved using Nesterov's Accelerated Gradient Descent (Nesterov's AGD, or simply AGD). Specifically, while the standard gradient descent method achieves a convergence rate of $\mathcal{O}(1/T)$ after T iterations, AGD improves this to $\mathcal{O}(1/T^2)$. Moreover, for σ-strongly convex and L-smooth functions with $\kappa = L/\sigma$ being the condition number, GD achieves an $\mathcal{O}(\exp(-T/\kappa))$ convergence rate, whereas AGD can attain an accelerated rate of order $\mathcal{O}(\exp(-T/\sqrt{\kappa}))$.

In fact, Nesterov's AGD is not the only algorithm capable of achieving accelerated convergence rates for smooth convex optimization. Equally important is its demonstration of the possibility of achieving accelerated rates for general smooth convex optimization (note that Polyak's heavy ball method, while accelerated, is restricted to the quadratic case). In this lecture, we will explore acceleration from the lens of *optimistic online learning*. Specifically, we will design an accelerated algorithm based on the optimistic OMD framework that we have covered so far, and see how these online learning techniques can truly shine in the optimization community.

Problem Setup. We focus on the *constrained* optimization problem:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex and compact feasible domain, and $f : \mathbb{R}^d \to \mathbb{R}$ is a convex function.

Assumptions. We assume that f is L-smooth over $\mathcal{X}$, i.e., $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Moreover, we assume the feasible domain is bounded, namely, $\|\mathbf{x} - \mathbf{y}\| \leq D$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. By default, we focus on the ℓ_2 norm without loss of generality.

2 Acceleration via Optimistic Online Learning

As we have seen in the previous lecture, a standard approach to applying online learning techniques to optimization is the *online-to-batch conversion*. Given an offline optimization problem $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$, we can use an online learning algorithm $\mathcal{A}_{\mathsf{OL}}$ to learn a sequence of iterates $\{\mathbf{x}_t\}_{t=1}^T$. By averaging these iterates, we obtain $\bar{\mathbf{x}}_T = \frac{1}{T}\sum_{t=1}^T \mathbf{x}_t$, which is then used as the solution to the offline optimization problem. The convergence rate of the offline problem is related to the regret of the online learning algorithm $\mathcal{A}_{\mathsf{OL}}$. A typical example is using online gradient descent (OGD) to solve a convex problem, which achieves a convergence rate of $\mathcal{O}(1/\sqrt{T})$.

To achieve accelerated rates, like $\mathcal{O}(1/T^2)$ for smooth convex functions, we need steps further. First, we need to enhance the power of the conversion to gain a speedup. Second, we need to employ *optimistic* online learning methods, rather than standard problem-independent algorithms, to improve the learning behavior. Thus, there are two key components:

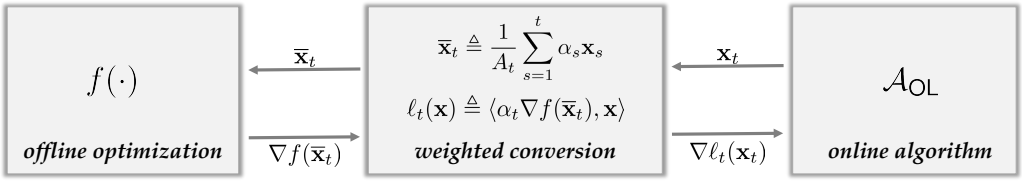

Figure 1: A pipeline of weighted online-to-batch conversion.

(i) **Weighted Online-to-Batch Conversion**: reduce offline optimization to online optimization, but now a weighted and stabilized version is needed to achieve acceleration.

(ii) **Optimistic OMD with Optimism Design**: achieve the desired constant regret in online optimization, where the optimism design is crucial by leveraging the problem structure.

In the following two subsections, we will present the two components in detail. We will then combine them to form the final algorithm and provide the convergence guarantee in Section 2.3.

2.1 Weighted Online-to-Batch Conversion

Algorithm 1 presents a template of the weighted online-to-batch conversion. To solve an offline optimization problem $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$, we will require an online learning algorithm $\mathcal{A}_{\mathsf{OL}}$ and a sequence of weights $\{\alpha_t\}_{t=1}^T$ with $\alpha_t > 0$.

The reduction scheme goes as follows: at each round $t \in [T]$, we will first calculate the weighted average $\bar{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s\mathbf{x}_s$ with $A_t \triangleq \sum_{s=1}^t \alpha_s$, and then query the gradient $\nabla f(\bar{\mathbf{x}}_t)$; then we will define an online function $\ell_t(\mathbf{x}) = \langle \alpha_t \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}\rangle$ and run the online learning algorithm $\mathcal{A}_{\mathsf{OL}}$ on $\{\ell_s(\mathbf{x})\}_{s=1}^t$. The output of $\mathcal{A}_{\mathsf{OL}}$ $\mathbf{x}_{t+1}$ is then fed to the weighted summation calculation for the next round. The final output of the offline optimization problem is then given by $\bar{\mathbf{x}}_T = \frac{1}{A_T}\sum_{t=1}^T \alpha_t\mathbf{x}_t$.

An illustration of the pipeline is given in Figure 1.

Algorithm 1 Weighted Online-to-Batch Conversion Template

Input: Online learning algorithm $\mathcal{A}_{\mathsf{OL}}$, weights $\{\alpha_t\}_{t=1}^T$ with $\alpha_t > 0$.
1: **Initialization:** $\mathbf{x}_1 \in \mathcal{X}$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Calculate $\bar{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s\mathbf{x}_s$ with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and receive $\nabla f(\bar{\mathbf{x}}_t)$
4: Define $\ell_t(\mathbf{x}) = \langle \alpha_t \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}\rangle$ as t-th round online function to $\mathcal{A}_{\mathsf{OL}}$
5: Obtain $\mathbf{x}_{t+1}$ from $\mathcal{A}_{\mathsf{OL}}(\mathbf{x}_1, \{\ell_s(\mathbf{x})\}_{s=1}^t)$
6: **end for**
Output: $\bar{\mathbf{x}}_T = \frac{1}{A_T}\sum_{t=1}^T \alpha_t\mathbf{x}_t$

For the weighted online-to-batch conversion, the following theorem relates the convergence rate of the offline optimization problem to the weighted regret of the online learning algorithm.

Theorem 1 (weighted online-to-batch conversion)**.** *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated):*

$$\bar{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s\mathbf{x}_s, \tag{2}$$

with a weight $\alpha_t > 0$ and cumulative sum $A_t \triangleq \sum_{s=1}^t \alpha_s$, the following holds for any $\mathbf{x}^\star \in \mathcal{X}$:

$$f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^\star) \le \frac{1}{A_T}\cdot\sum_{t=1}^T \langle \alpha_t \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle \triangleq \frac{1}{A_T}\cdot \textsc{Reg}_T^{\alpha}(\mathbf{x}^\star), \tag{3}$$

where $\textsc{Reg}_T^{\alpha}(\mathbf{x}^\star)$ is the weighted regret of the online learning algorithm $\mathcal{A}_{\mathsf{OL}}$.

Proof. First, by convexity of f, we have

$$\sum_{t=1}^{T} \alpha_t (f(\bar{\mathbf{x}}_t) - f(\mathbf{x}^\star)) \leq \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{\mathbf{x}}_t), \bar{\mathbf{x}}_t - \mathbf{x}^\star \rangle$$
$$= \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle + \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{\mathbf{x}}_t), \bar{\mathbf{x}}_t - \mathbf{x}_t \rangle.$$

The first term represents the weighted regret $\textsc{Reg}_T^\alpha(\mathbf{x}^\star)$, while the second term is a bias term. By definition, we have

$$A_t \bar{\mathbf{x}}_t = \sum_{s=1}^{t} \alpha_s \mathbf{x}_s = A_{t-1} \bar{\mathbf{x}}_{t-1} + \alpha_t \mathbf{x}_t,$$

which further implies

$$\alpha_t \bar{\mathbf{x}}_t - \alpha_t \mathbf{x}_t = A_{t-1}(\bar{\mathbf{x}}_{t-1} - \bar{\mathbf{x}}_t).$$

Consequently, we obtain

$$\sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{\mathbf{x}}_t), \bar{\mathbf{x}}_t - \mathbf{x}_t \rangle = -\sum_{t=1}^{T} A_{t-1} \langle \nabla f(\bar{\mathbf{x}}_t), \bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1} \rangle \leq -\sum_{t=1}^{T} A_{t-1} \Big(f(\bar{\mathbf{x}}_t) - f(\bar{\mathbf{x}}_{t-1}) \Big).$$

Combining this with the first inequality, we get

$$\sum_{t=1}^{T} \alpha_t (f(\bar{\mathbf{x}}_t) - f(\mathbf{x}^\star)) + \sum_{t=1}^{T} A_{t-1} \Big(f(\bar{\mathbf{x}}_t) - f(\bar{\mathbf{x}}_{t-1}) \Big) \leq \textsc{Reg}_T^\alpha(\mathbf{x}^\star).$$

Applying telescoping and dividing both sides by A_T completes the proof. □

Stabilization Effect. There is actually another alterative weighted online-to-batch conversion:

$$f\left(\frac{1}{A_T} \sum_{t=1}^{T} \alpha_t \mathbf{x}_t \right) - f(\mathbf{x}^\star) \leq \frac{1}{A_T} \sum_{t=1}^{T} \alpha_t \Big(f(\mathbf{x}_t) - f(\mathbf{x}^\star) \Big) \leq \frac{1}{A_T} \sum_{t=1}^{T} \alpha_t \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle,$$

where we only use the convexity of f and the definition of $\bar{\mathbf{x}}_T$. The key difference is that here the gradient $\nabla f(\mathbf{x}_t)$ is taken at the online algorithm's iterate $\mathbf{x}_t$, whereas the proposed conversion in Theorem 1 computes $\nabla f(\bar{\mathbf{x}}_t)$ using the weighted average $\bar{\mathbf{x}}_t$ as in Eq. (3).

The advantage of the weighted sequence $\{\bar{\mathbf{x}}_t\}_{t=1}^T$ is its significantly more *stable* behavior due to the smoothing effect of averaging, compared to the original iterates $\{\mathbf{x}_t\}_{t=1}^T$, see Remark 1 for a comparison. This gradient stability can be highly beneficial if the online algorithm can take advantage of it. In this sense, the conversion in Eq. (3) can be viewed not only as a weighted generalization but also as a *stabilized* form of online-to-batch conversion. Consequently, we also refer to the conversion in Algorithm 1 as the *stabilized weighted online-to-batch conversion*.

2.2 Optimism Design

With the weighted online-to-batch conversion, we now need to design the online algorithm $\mathcal{A}_{\mathsf{OL}}$ and choose the weights to achieve the targeting $\mathcal{O}(1/T^2)$ convergence rate for convex and smooth optimization. A natural choice is to set $\alpha_t = t$ such that $A_T = \Theta(T^2)$. This reduces our task to designing $\mathcal{A}_{\mathsf{OL}}$ with a constant weighted regret, i.e., $\textsc{Reg}_T^\alpha(\mathbf{x}^\star) \leq \mathcal{O}(1)$.

However, achieving this requirement is non-trivial. While increasing α_t enlarges the denominator A_T, it also amplifies the weighted regret $\textsc{Reg}_T^\alpha(\mathbf{x}^\star)$, as the gradient norm of online function is scaled by α_t. Therefore, an important requirement of this online algorithm is the *adaptivity* that can effectively take advantage of the gradient stability. Recall that this capability has been discussed in the previous lecture: we use optimistic OMD to achieve $\mathcal{O}(\sqrt{V_T})$ gradient-variation regret with $V_T = \sum_{t=2}^{T} \sup_{\mathbf{x} \in \mathcal{X}} \|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|^2$ for online convex optimization over $\{f_t(\mathbf{x})\}_{t=1}^T$.

To this end, we adopt the framework of Optimistic OMD (OOMD) with a carefully crafted optimism. For simplicity, we focus on OOMD with a constant step size, whose update rule is:

$$\begin{aligned} \mathbf{x}_t &= \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \eta \langle M_t, \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2 \right\} \\ \widehat{\mathbf{x}}_{t+1} &= \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \eta \langle \nabla \ell_t(\mathbf{x}_t), \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2 \right\} \end{aligned} \tag{4}$$

Here, the gradient is given by $\nabla \ell_t(\mathbf{x}_t) = \alpha_t \nabla f(\bar{\mathbf{x}}_t)$, as specified in Line 4 of Algorithm 1. A key issue remains is the design of the optimism M_t. To ensure a good approximation of $\nabla \ell_t(\mathbf{x}_t)$ (and thereby achieve a constant weighted regret), we define it in the following form:

$$M_t = \alpha_t \nabla f(\widetilde{\mathbf{x}}_t) \tag{5}$$

with $\widetilde{\mathbf{x}}_t$ to be determined (using only available information). By definition,

$$\bar{\mathbf{x}}_t = \frac{1}{A_t}\left(\sum_{s=1}^{t-1} \alpha_s \mathbf{x}_s + \alpha_t \mathbf{x}_t \right), \tag{6}$$

and thus we can set the optimism as

$$\widetilde{\mathbf{x}}_t \triangleq \frac{1}{A_t}\left(\sum_{s=1}^{t-1} \alpha_s \mathbf{x}_s + \alpha_t \mathbf{x}_{t-1} \right). \tag{7}$$

By the regret guarantee of OOMD (with a bounded domain diameter D), we have

$$\begin{aligned} \sum_{t=1}^{T} \ell_t(\mathbf{x}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{u}) &\le \frac{D^2}{2\eta} + \eta \sum_{t=1}^{T} \|\alpha_t \nabla f(\bar{\mathbf{x}}_t) - \alpha_t \nabla f(\widetilde{\mathbf{x}}_t)\|_2^2 - \frac{1}{4\eta} \sum_{t=1}^{T} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2 \\ &\le \frac{D^2}{2\eta} + \eta \sum_{t=1}^{T} \alpha_t^2 L^2 \|\bar{\mathbf{x}}_t - \widetilde{\mathbf{x}}_t\|_2^2 - \frac{1}{4\eta} \sum_{t=1}^{T} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2 \\ &= \frac{D^2}{2\eta} + \sum_{t=1}^{T} \left(\eta \frac{\alpha_t^4 L^2}{A_t^2} - \frac{1}{4\eta} \right) \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 . \end{aligned} \tag{8}$$

The second inequality is by L-smoothness. Recall that $\alpha_t = t$ and $A_t = \frac{t(t+1)}{2}$. By setting $\eta = \frac{1}{4L}$, it holds that $\frac{\eta \alpha_t^4 L^2}{A_t^2} - \frac{1}{4\eta} \le 0$, which indicates that $\textsc{Reg}_T^{\alpha}(\mathbf{x}^\star) \le 2D^2 L = \mathcal{O}(1)$.

Combining the constant weighted regret guarantee with the fact $A_T = \Theta(T^2)$, by Theorem 1 we have obtained an $\mathcal{O}(T^{-2})$ convergence rate!

Remark 1. *A key feature of the weighted online-to-batch conversion is the stability of the weighted average $\bar{\mathbf{x}}_t$, which plays a crucial role in the analysis. Specifically, Eq. (8) holds by noting that*

$$\|\bar{\mathbf{x}}_t - \widetilde{\mathbf{x}}_t\|_2^2 = \left(\frac{\alpha_t}{A_t} \right)^2 \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 = \Theta\left(\frac{1}{t^2} \right) \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2. \tag{9}$$

This indicates that the weighted average is $\Theta(1/t^2)$ more stable than the original sequence, enabling the negative term cancellation in Eq. (8), which finally leads to a constant weighted regret.

2.3 Algorithm and Convergence Rate

The above two subsections have already outlined the final algorithm to achieve the accelerated rate for convex and smooth functions, built on two key components: *(i)* the weighted online-to-batch conversion, and *(ii)* optimistic OMD with a carefully designed optimism. The concrete updates are summarized in Algorithm 2, and we have the following accelerated convergence rate guarantee.

Theorem 2. *Set the weight sequence as $\alpha_t = t$ for all $t \in [T]$, and choose the step size as $\eta = \frac{1}{4L}$. The* AcceleOOMD *algorithm in Algorithm 2 guarantees that*

$$f(\bar{\mathbf{x}}_T) - \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \le \frac{4D^2 L}{T(T+1)} = \mathcal{O}\left(\frac{1}{T^2} \right) \tag{10}$$

Proof. The result follows immediately from the (stabilized) weighted online-to-batch conversion in Theorem 1, along with the results of $\mathcal{O}(1)$ weighted regret and $A_T = \Theta(T^2)$ established in the previous two subsections. □

Algorithm 2 AcceleOOMD: Acceleration via Optimistic OMD

Input: smoothness parameter L, weight sequence $\{\alpha_t\}_{t=1}^T$, step size $\eta = \frac{1}{4L}$.

Initialization: $\mathbf{x}_1 \in \mathcal{X}$ and $\widehat{\mathbf{x}}_1 \in \mathcal{X}$.

for $t = 1, 2, \ldots, T$ **do**

  Update $\widehat{\mathbf{x}}_{t+1} = \Pi_{\mathcal{X}}[\widehat{\mathbf{x}}_t - \eta\alpha_t \nabla f(\bar{\mathbf{x}}_t)]$ with $\bar{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$

  Update $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\widehat{\mathbf{x}}_{t+1} - \eta\alpha_{t+1} \nabla f(\widetilde{\mathbf{x}}_{t+1})]$ with $\widetilde{\mathbf{x}}_{t+1} = \frac{1}{A_{t+1}}\left(\sum_{s=1}^t \alpha_s \mathbf{x}_s + \alpha_{t+1}\mathbf{x}_t\right)$

end for

Output: $\bar{\mathbf{x}}_T = \frac{1}{A_T}\sum_{t=1}^T \alpha_t \mathbf{x}_t$

3 Discussions

The analysis presented here is crystalized from UnixGrad [Kavis et al., 2019]. The idea of introducing weights into the online-to-batch conversion dates back at least to [Abernethy et al., 2018], and the stabilized version that we presented in Theorem 1 originates from [Cutkosky, 2019].

Unconstrained Optimization. The above analysis can be extended to the unconstrained optimization setting, where the constraint set is $\mathcal{X} = \mathbb{R}^d$. This can be achieved by using optimistic FTRL template, which is more amenable to the unconstrained setting in online learning.

Smooth and Strongly Convex Functions. To achieve the accelerated rate of $\mathcal{O}(\exp(-T/\sqrt{\kappa}))$ for L-smooth and σ-strongly convex functions (where $\kappa = L/\sigma$), one can also employ optimistic OMD and the weighted online-to-batch conversion with $\alpha_t = a^t$ for a suitable $a > 0$. An additional useful ingredient is the negative Bregman divergence term arising from the linearization [Joulani et al., 2020], i.e., $f(\mathbf{x}) - f(\mathbf{y}) = \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y}\rangle - \mathcal{D}_f(\mathbf{y}, \mathbf{x})$, which follows directly from the definition of the Bregman divergence. While this term is often simplified to $f(\mathbf{x}) - f(\mathbf{y}) \le \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y}\rangle$ by dropping the negative term $-\mathcal{D}_f(\mathbf{y}, \mathbf{x})$, but here it turns out to be crucial for the analysis.

References

Ali Kavis, Kfir Y. Levy, Francis R. Bach, and Volkan Cevher. UniXGrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 6257–6266, 2019.

Jacob D. Abernethy, Kevin A. Lai, Kfir Y. Levy, and Jun-Kun Wang. Faster rates for convex-concave games. In *Proceedings of the 31st Conference on Learning Theory (COLT)*, pages 1595–1625, 2018.

Ashok Cutkosky. Anytime online-to-batch, optimism and acceleration. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 1446–1454, 2019.

Pooria Joulani, Anant Raj, Andras Gyorgy, and Csaba Szepesvári. A simpler approach to accelerated optimization: iterative averaging meets optimism. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 4984–4993, 2020.