Basics of Online Convex Optimization

Yanze Song

November 14, 2023

1 Intro to OCO

1.1 Problem Formulation

Online Convex Optimization (*OCO*) can be considered as a repetitive game between the player/algorithm and the environment/adversary. Denote the convex decision space by $\mathcal{C}$, the convex loss function by $f : \mathcal{C} \to \mathbb{R}$, the time horizon by T. For $t = 1, 2, ..., T$:

- Algorithm: Picks a decision vector $\boldsymbol{w}_t \in \mathcal{C}$,
- Adversary: Picks a convex loss function f_t, and send *feedback* to the algorithm,
- Algorithm: Suffer from loss, learn, and adapt from the feedback.

1.2 Performance Metric

We use *regret*, the difference between the loss generated by the algorithm and the loss by the best **fixed** decision in hindsight, to quantify the performance of the algorithm:

$$\text{Regret}_T \coloneqq \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{C}} \sum_{t=1}^{T} f_t(\boldsymbol{u}).$$

and sometimes it's defined w.r.t. some $\boldsymbol{u} \in \mathcal{C}$:

$$\text{Regret}_T(\boldsymbol{u}) \coloneqq \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}).$$

This measures how much we "regret" for not picking the best fixed decision in hindsight. Since we're comparing with a fixed decision, this is called *static regret* more precisely, and it can indeed be negative. Then two natural questions may arise:

- Why do we use the best fixed decision instead of the offline optimal as the benchmark? The best fixed decision already has all information ahead of time, while the algorithm only receives one piece at a time, and hence, competing with it is already highly non-trivial.
- What if the best fixed decision doesn't perform well on the losses either? Then we will switch to other stronger regret measures, e.g., comparing with the offline optimal, *dynamic regret*:

$$\text{D-Regret}_T \coloneqq \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u}_t \in \mathcal{C}} \sum_{t=1}^{T} f_t(\boldsymbol{u}_t).$$

Another reason for studying static regret is that the algorithms for those stronger regret measures are often designed by extending the ideas from static regret.

1.3 Goal

Due to the online setting, it's unrealistic to achieve zero regret, so we reasonably allow some mistakes. If we can achieve *sub-linear regret*, i.e.,

$$\text{Regret}_T \in o(T),$$

then $\lim_{T\to\infty} \text{Regret}/T = 0$, which means that on average the algorithm is performing as well as the best fixed decision. In this case, the algorithm/learner is called "no-regret", indicating that we don't regret for making those decisions.

1.4 Feedback Model

Throughout this note, we assume **full information feedback**, that is, f_t can be fully observed at each round.

1.5 Adversary Model

2 OLO & OCO

Theorem 2.0.1. *The regret of any algorithm on an OLO problem is no better than if it is on the corresponding OCO problem. In other words, for any algorithm* $\mathcal{A}$*,*

$$\max_{f_t} \operatorname{Regret}(\mathcal{A}) \leq \max_{l_t} \operatorname{Regret}(\mathcal{A}),$$

which makes linear loss the worst-case instance in OCO.

Proof. Let $f_t : \mathcal{C} \to \mathbb{R}$ be a convex function, $\boldsymbol{w}_t \in \mathcal{C}$ be our t^{th} decision, and $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t)$. Then for all $\boldsymbol{w} \in \mathcal{C}$, by convexity we have

$$f_t(\boldsymbol{w}_t) + \langle \boldsymbol{g}_t, \boldsymbol{w} - \boldsymbol{w}_t \rangle \leq f_t(\boldsymbol{w}) \Rightarrow f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}) \leq \langle \boldsymbol{g}_t, \boldsymbol{w}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{w} \rangle,$$

and if we sum over t, we have

$$\sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}) \leq \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{w} \rangle.$$

Denote the argmax of the LHS by $\boldsymbol{w}^*$. Substituting $\boldsymbol{w}^*$ into above, and by the maximality of the RHS, we have

$$\max_{\boldsymbol{w} \in \mathcal{C}} \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}) \leq \max_{\boldsymbol{w} \in \mathcal{C}} \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{w} \rangle,$$

which completes the proof. Therefore, it's sufficient to only consider the losses are linear. The regret bound of an algorithm on linear loss (with some constraints) can be extended to that on general convex loss, provided that the constraints on the loss are equivalent. □

Proof. Here's a more general proof: if we can find $\hat{f}_t$ s.t.

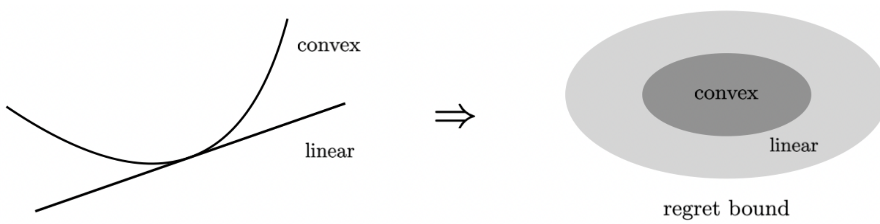
- $\hat{f}_t(\boldsymbol{w}_t) = f_t(\boldsymbol{w}_t)$,
- $\hat{f}_t(\boldsymbol{w}) \leq f_t(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathcal{C}$,

then we will have

$$\sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{C}} \sum_{t=1}^{T} f_t(\boldsymbol{u}) \leq \sum_{t=1}^{T} \hat{f}_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{C}} \sum_{t=1}^{T} \hat{f}_t(\boldsymbol{u}),$$

where $\hat{f}_t$ can be chosen as the linearization of f_t at $\boldsymbol{w}_t$. □

The following graph may help visualize the proof:



3 OCO Algorithms

3.1 Follow The Leader

A natural idea is to use the decision that minimizes the total loss of all previous rounds. This is called the *Follow-the-Leader* (*FTL*) algorithm:

$$\boldsymbol{w}_{t+1} \coloneqq \operatorname*{argmin}_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^{t} f_i(\boldsymbol{w}).$$

Lemma 3.1.1 (*Regret Bound of FTL*)**.** *Let $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ...\}$ be the decisions generated by FTL, then for all $\boldsymbol{u} \in \mathcal{C}$, we have*

$$\text{Regret}_T(\boldsymbol{u}) \le \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})).$$

Proof. By definition of the regret, we can rewrite the above, and prove the following by induction:

$$\sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})) \le \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})) \Leftrightarrow \sum_{t=1}^{T} f_t(\boldsymbol{w}_{t+1}) \le \sum_{t=1}^{T} f_t(\boldsymbol{u}),$$

When $T = 1$, we have LHS $= f_1(\boldsymbol{w}_2) \le f_1(\boldsymbol{u})$, because $w_2 \coloneqq \operatorname{argmin}_{\boldsymbol{w}\in\mathcal{C}} f_1(\boldsymbol{w})$, and this proves the base case. Then we assume $\sum_{t=1}^{T} f_t(\boldsymbol{w}_{t+1}) \le \sum_{t=1}^{T} f_t(\boldsymbol{u})$ for some $T \ge 1$, but we'll only use when $\boldsymbol{u} = \boldsymbol{w}_{T+2}$. We then have,

$$\begin{aligned}
\sum_{t=1}^{T+1} f_t(\boldsymbol{w}_{t+1}) &= \sum_{t=1}^{T} f_t(\boldsymbol{w}_{t+1}) + f_{T+1}(\boldsymbol{w}_{T+2}) \\
&\le \sum_{t=1}^{T} f_t(\boldsymbol{w}_{T+2}) + f_{T+1}(\boldsymbol{w}_{T+2}) \\
&\le \sum_{t=1}^{T+1} f_t(\boldsymbol{w}_{T+2}),
\end{aligned}$$

and this completes the proof, since $\boldsymbol{w}_{T+2} \coloneqq \operatorname{argmin}_{\boldsymbol{w}\in\mathcal{C}} \sum_{t=1}^{T+1} f_t(\boldsymbol{w})$. □

Remark 3.1.1. If the decisions generated by FTL are not stable, i.e., two consecutive decisions are far from each other, then $f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})$ will likely be large, and hence the regret bound can be high (not necessarily tight) after summing up T terms.

Example 3.1.1 (*Failure of FTL*)**.** The regret of FTL is $\Omega(T)$, which can be achieved by a convex feasible set $[0, 1]$, a sequence of linear loss $f_t(w) = z_t w$, where

$$z_t = \begin{cases} -0.5, & \text{if } t = 1, \\ 1, & \text{elif } t \text{ is even}, \\ -1, & \text{elif } t \text{ is odd}, \end{cases}$$

and pick $u = 0$. We can observe the instability: decisions are alternating between the boundaries. This happens because linear function has no curvature (or has linearity), allowing that

- Upshape/Downshape can be switched easily, which destabilizes the decisions,
- Optima are always on the boundary, keeping a non-decreasing (fixed) distance between decisions.

3.2 Follow the Regularized Leader

We now know that the flaw of FTL, and therefore, we are to stabilize the algorithm. One question may arise: why do we need stability while the losses are adversarially chosen? This is because we're competing with a **fixed** decision, so we don't have to chase the loss.

3.3 Online Gradient Descent

The *Online Gradient Descent* algorithm is an instance of FTRL. Consider an OLO with $f_t(\boldsymbol{w}) = \langle \boldsymbol{z}_t, \boldsymbol{w}_t \rangle$, and $\mathcal{C} = \mathbb{R}^n$, we're to instantiate FTRL with $R(\boldsymbol{w}) = \frac{1}{2\eta}||\boldsymbol{w}||_2^2$ for some $\eta > 0$, then

$$\boldsymbol{w}_{t+1} \coloneqq \operatorname*{argmin}_{\boldsymbol{w} \in \mathcal{C}} \sum_{i=1}^{t} \langle \boldsymbol{z}_t, \boldsymbol{w} \rangle + \frac{1}{2\eta}||\boldsymbol{w}||_2^2.$$

We then have $\boldsymbol{w}_{t+1} = -\eta \sum_{i=1}^{t} \boldsymbol{z}_i = -\eta \sum_{i=1}^{t-1} \boldsymbol{z}_i - \eta \boldsymbol{z}_t = \boldsymbol{w}_t - \eta \boldsymbol{z}_t$, where $\boldsymbol{z}_t = \nabla f_t(\boldsymbol{w})$ for all $\boldsymbol{w}$, so in particular we have $\boldsymbol{z}_t = \nabla f_t(\boldsymbol{w}_t)$. Rewriting this yields OGD:

$$\boldsymbol{w}_{t+1} \coloneqq \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t).$$

3.4 Online Mirror Descent

3.5 Online Newton Step

4 Lower Bound of OCO

5 Beyond General Convexity Assumptions

6 References