Basics of Online Convex Optimization

Yanze Song

February 27, 2024

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 $f : \mathcal{C} \to \mathbb{R}$, and the time horizon by T. For $t = 1, 2, ..., T$:

- Algorithm: Picks a decision vector $\boldsymbol{w}_t \in \mathcal{C}$,
- Adversary: Picks and reveals a convex loss f_t with **full information** to the algorithm,
- Algorithm: Suffers, learns and adapts from the loss.

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:

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

and it can also be defined w.r.t. some fixed decision $\boldsymbol{u} \in \mathcal{C}$:

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

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 precisely called *static regret*, and it can indeed be negative. Then two natural questions may arise:

1. *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.

2. *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, which serves as the foundation.

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.,

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

then $\lim_{T\to\infty} \mathrm{Regret}_T/T = 0$, 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 they don't regret for making those decisions.

1.4 Feedback Model

We assume full information feedback for general OCO, i.e., f_t will be fully observed when revealed.

2 OCO & OLO

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

$$\operatorname{Regret}(\mathcal{A}, \mathcal{C}') \leq \operatorname{Regret}(\mathcal{A}, \mathcal{L}),$$

provided that the classes of convex and linear functions, $\mathcal{C}'$ and $\mathcal{L}$, are equivalent.

Proof. Let $\boldsymbol{w}_t \in \mathcal{C}$ be our t^{th} decision, then for all $\boldsymbol{w} \in \mathcal{C}$, by convexity we have

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

Let $\boldsymbol{w}^*, \boldsymbol{w}^{**}$ be the maximizers of the LHS and RHS over $\mathcal{C}$ respectively, then instantiating with $\boldsymbol{w} = \boldsymbol{w}^*$ yields

$$\begin{aligned}\sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}^*) &\leq \sum_{t=1}^{T} \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}_t \rangle - \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}^* \rangle \\ &\leq \sum_{t=1}^{T} \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}_t \rangle - \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}^{**} \rangle\end{aligned}$$

which completes the proof. □

Remark 2.0.1. The theorem implies that linear loss is the worst-case instance in OCO, so it suffices to consider the linear loss only. In general, for any OCO problem, our decisions will exactly be the ones generated by solving the corrsponding OLO problem (we don't solve OCO directly).

Proof. Here's a more general proof: for each round, if we can find $\hat{f}_t \in \hat{\mathcal{F}}$ (some function class) 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 $\boldsymbol{w} \in \mathcal{C}$,

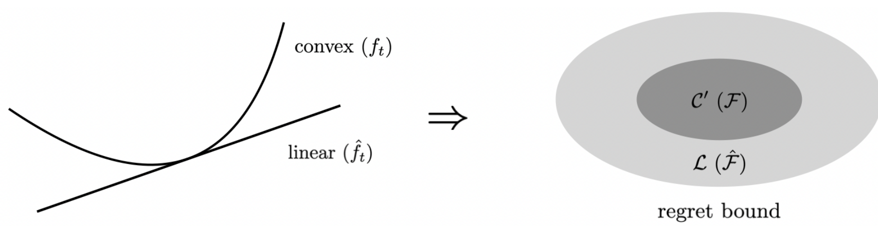
or slightly more strictly, $\hat{f}_t$ should be s.t. for all $\boldsymbol{w} \in \mathcal{C}$,

$$f_t \textit{ has an under-estimator } \hat{f}_t \in \hat{\mathcal{F}} \textit{ tangent at } (\boldsymbol{w}_t, f_t(\boldsymbol{w}_t)),$$

and then we'll have

$$\underbrace{\sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{C}} \sum_{t=1}^{T} f_t(\boldsymbol{u})}_{\operatorname{Regret}(\mathcal{F})} \leq \underbrace{\sum_{t=1}^{T} \hat{f}_t(\boldsymbol{w}_t) - \min_{\boldsymbol{u} \in \mathcal{C}} \sum_{t=1}^{T} \hat{f}_t(\boldsymbol{u})}_{\operatorname{Regret}(\hat{\mathcal{F}})}.$$

In our case where $\mathcal{F} = \mathcal{C}'$, we choose $\hat{\mathcal{F}} = \mathcal{L}$ and $\hat{f}_t$ as the linearization of f_t at $\boldsymbol{w}_t$. The following graph may help visualize the relationship:



□

3 OCO Algorithms

3.1 Follow The Leader

The benchmark decisions are $\{\boldsymbol{u}_t^* = \arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^t f_i(\boldsymbol{w})\}_{t=1}^T$. From the algorithm perspective, f_{t+1} won't be revealed until the end of time $t+1$, and therefore, it's natural to use the previous cumulative loss only, regardless of f_{t+1}. This gives us the *Follow-the-Leader* (*FTL*) algorithm:

$$\boldsymbol{w}_{t+1} \coloneqq \arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^t f_i(\boldsymbol{w}) = \boldsymbol{u}_t^*.$$

In this case, we're only left behind one step, so for FTL to work, we have to ensure that counting the extra f_{t+1} at each step won't change the output that much, i.e., $\boldsymbol{w}_t = \boldsymbol{u}_{t-1}^*$ and $\boldsymbol{u}_t^*$ must be getting closer and closer. In other words, the outputs $\{\boldsymbol{u}_t^*\}_{t=1}^T$ must be stabilized.

This fully depends on the curvature of the loss. Informally speaking, we should require the loss to have some curvature s.t. *the whole shape is stable after adding any new loss.* This of course doesn't hold for general convex loss such as the linear ones. Now we'll see some lemmas to quantify the arguments above.

Algorithm 1 Follow the Leader

Input: $\boldsymbol{w}_1 \in \mathcal{C}$
for $t = 1, 2, \ldots, T$ **do**
  Output $\boldsymbol{w}_t \in \mathcal{C}$
  Receive $f_t : \mathcal{C} \to \mathbb{R}$
  Update $\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^t f_t(\boldsymbol{w})$
end for

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

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

Proof. By the definition of 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 $w_2 \coloneqq \arg\min_{\boldsymbol{w}\in\mathcal{C}} f_1(\boldsymbol{w})$, and this proves the base case. Then 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 $\boldsymbol{w}_{T+2} \coloneqq \arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{t=1}^{T+1} f_t(\boldsymbol{w})$. □

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, 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.

Example 3.1.2 (*Success of FTL*)**.** Consider an online quadratic optimization problem with $\mathcal{C} = \mathbb{R}^n$ and $f_t(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}_t\|^2$, FTL enjoys $\mathcal{O}(L^2 \log T)$ regret bound, where $L = \max_t\{\|\boldsymbol{z}_t\|\}$. For all t, we have

$$\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{t-1} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}_t\|^2 = \frac{1}{t-1}\sum_{i=1}^{t-1} \boldsymbol{z}_i,$$

and hence

$$\boldsymbol{w}_{t+1} = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{z}_i = (1 - \frac{1}{t})\boldsymbol{w}_t + \frac{1}{t}\boldsymbol{z}_t.$$

Then we use Lemma 3.1.1,

$$\begin{aligned} f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1}) &= \frac{1}{2}\|\boldsymbol{w}_t - \boldsymbol{z}_t\|^2 - \frac{1}{2}\|\boldsymbol{w}_{t+1} - \boldsymbol{z}_t\|^2 \\ &= \frac{1}{2}(1 - (1 - \frac{1}{t})^2)\|\boldsymbol{w}_t - \boldsymbol{z}_t\|^2 \\ &= (\frac{1}{t} - \frac{1}{2t^2})^2\|\boldsymbol{w}_t - \boldsymbol{z}_t\|^2 \\ &\leq \frac{1}{t}\|\boldsymbol{w}_t - \boldsymbol{z}_t\|^2. \end{aligned}$$

Then for all t, $\|\boldsymbol{w}_t\| = \|\frac{1}{t-1}\sum_{i=1}^{t-1} \boldsymbol{z}_t\| \leq \frac{1}{t-1}\sum_{i=1}^{t-1}\|\boldsymbol{z}_t\| \leq \frac{1}{t-1}\sum_{i=1}^{t-1} L = L$, and hence for all i, t, $\|\boldsymbol{w}_i - \boldsymbol{z}_t\| \leq \|\boldsymbol{w}_i\| + \|\boldsymbol{z}_t\| \leq 2L$. Summing up over t yields

$$\sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})) \leq 4L^2\sum_{t=1}^{T}\frac{1}{t} \leq 4L^2(\log T + 1) \in \mathcal{O}(L^2 \log T),$$

since $\sum_{t=1}^{T} 1/t \leq \log T + 1$ for all t.

Remark 3.1.2. In fact, for any quadratic loss, and more generally for any strongly convex loss (supported by Proof 2), the curvature is nice enough s.t. the whole shape is more and more stable, which leads to a more and more stable benchmark decision sequence $\{\boldsymbol{u}_t^*\}_{t=1}^T$. Now, we shall state that given stability, FTL is very good at chasing the best fixed decision.

3.2 Follow the Regularized Leader

The regret bound for FTL can be high if the decisions are unstable. One way to guarantee stability as we saw in Example 3.1.2, is to assume the loss has some nice curvature such as strong convexity. However, this is not applicable in the general OCO setting.

To achieve this, a naive idea is to add an extra term which is strongly convex, and once the strong convexity is incurred to the problem, it never decays, quantified by the fact that convex plus σ-strongly convex is σ-strongly convex. Now, we're ready to implement this idea.

Instead of directly minimizing the previous t losses, we instead minimize a slightly different expression with an extra strongly convex term $\frac{1}{\eta}\psi$, for some tunable parameter $\eta > 0$, and this gives the *Follow-the-Regularized-Leader* (*FTRL*) algorithm, which defines a family of algorithms:

$$\boldsymbol{w}_{t+1} \coloneqq \arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^{t} f_i(\boldsymbol{w}) + \frac{1}{\eta}\psi(\boldsymbol{w}).$$

The ψ is called the *regularizer*, but this is essentially different from the overfitting issue in statistical machine learning. This is all about bringing some nice geometry to the problem.

Algorithm 2 Follow the Regularized Leader

Input: ψ, η, $\boldsymbol{w}_1 = \arg\min_{\boldsymbol{w}\in\mathcal{C}} \frac{1}{\eta}\psi(\boldsymbol{w})$
for $t = 1, 2, ..., T$ **do**
  Output $\boldsymbol{w}_t \in \mathcal{C}$
  Receive $f_t : \mathcal{C} \to \mathbb{R}$
  Update $\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^{t} f_t(\boldsymbol{w}) + \frac{1}{\eta}\psi(\boldsymbol{w})$
end for

Lemma 3.2.1 (*Regret Bound of FTRL*)**.** *Let $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ...\}$ be the decisions generated by FTRL, then for all $\boldsymbol{u} \in \mathcal{C}$, we have*

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \frac{1}{\eta}(\psi(\boldsymbol{u}) - \min_{\boldsymbol{w}\in\mathcal{C}} \psi(\boldsymbol{w})) + \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})).$$

Proof. Running FTRL on $\{f_1, f_2, ...\}$ is equivalent to running FTL on $\{f_0, f_1, ...\}$, where $f_0 = \frac{1}{\eta}\psi$. Therefore by Lemma 3.1.1 , we have for all $\boldsymbol{u} \in \mathcal{C}$,

$$\sum_{t=0}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})) \leq \sum_{t=0}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})),$$

$$f_0(\boldsymbol{w}_0) - f_0(\boldsymbol{u}) + \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u})) \leq f_0(\boldsymbol{w}_0) - f_0(\boldsymbol{w}_1) + \sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1})).$$

Then cancelling $f_0(\boldsymbol{w}_0)$, replacing f_0 by $\frac{1}{\eta}\psi$, $\sum_{t=1}^{T}(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}))$ by $\mathrm{Regret}_T(\boldsymbol{u})$, and $\boldsymbol{w}_1 \coloneqq \arg\min_{\boldsymbol{w}\in\mathcal{C}} \frac{1}{\eta}\psi(\boldsymbol{w})$ complete the proof. □

Theorem 3.2.1 (*Sublinear Regret of FTRL*)**.** *Let ψ be σ_ψ-strongly-convex w.r.t. $\|\cdot\|$, and f_t be L_t-Lipschitz w.r.t. $\|\cdot\|_*$ with $L = \max_t\{L_t\}$, then FTRL enjoys the following regret bound:*

$$\mathrm{Regret}_T \leq \frac{B_\psi}{\eta} + \sum_{t=1}^{T} \frac{\eta\|\nabla f_t(\boldsymbol{w}_t)\|_*^2}{\sigma_\psi} \leq \frac{B_\psi}{\eta} + \frac{\eta T L^2}{\sigma_\psi},$$

where $B_\psi = \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}} \psi(\boldsymbol{x}) - \psi(\boldsymbol{y})$. Setting $\eta = \sqrt{\frac{\sigma_\psi B_\psi}{TL^2}}$ optimally leads to

$$\mathrm{Regret}_T \in \mathcal{O}(\sqrt{\frac{L^2 B_\psi}{\sigma_\psi}T}).$$

Proof. The B_ψ/η part is trivial. We'll then prove $f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1}) \le \eta\|\nabla f_t(\boldsymbol{w}_t)\|_*^2/\sigma_\psi$. By convexity and the Hölder's inequality, it suffices to prove

$$\sigma_\psi\|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\| \le \eta\|\nabla f_t(\boldsymbol{w}_t)\|_*.$$

Let $F_t(\boldsymbol{w}) = \sum_{i=1}^t \eta\langle\nabla f_i(\boldsymbol{w}_i), \boldsymbol{w}\rangle + \psi(\boldsymbol{w})$, note that F_t is also σ_ψ-strongly convex, then we have

$$\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}\in\mathcal{C}} F_{t-1}(\boldsymbol{w}),\ \boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathcal{C}} F_t(\boldsymbol{w}).$$

By strong convexity,

$$F_{t-1}(\boldsymbol{w}_{t+1}) - F_{t-1}(\boldsymbol{w}_t) \ge \frac{\sigma_\psi}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\|^2,\ F_t(\boldsymbol{w}_t) - F_t(\boldsymbol{w}_{t+1}) \ge \frac{\sigma_\psi}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\|^2.$$

After summing up we yield

$$\eta\langle\nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}_t - \boldsymbol{w}_{t+1}\rangle \ge \sigma_\psi\|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\|^2,$$

and then after applying the Hölder's inequality to the r.h.s. we finish the proof. □

Remark 3.2.1. To understand the regret bound

$$\text{Regret}_T \le \underbrace{\frac{1}{\eta}(\max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}} \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}))}_{\text{penalty}} + \underbrace{\sum_{t=1}^T (f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1}))}_{\text{stability}},$$

- The extra penalty term is due to the imperfect mimic of the previous case: FTL with strongly convex loss. Although we have stability, we're now staying a little further from the best fixed decision for not directly solving $\arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^t f_t(\boldsymbol{w})$, which is known good at chasing the best fixed decision given stability. This distance essentially leads to the additionally inevitable penalty term in the regret bound, causing a $\mathcal{O}(\sqrt{T})$ bound instead of $\mathcal{O}(\log T)$.
- Stability: Since we're still using FTL-based algorithm, stability still matters.

Remark 3.2.2. Now we analyze how these assumptions, i.e. bounded B_ψ and ∇f_t potentially help to ensure stability and lead to a sub-linear regret intuitively:

- Bounded B_ψ: Informally equivalent to bounded domain. Restricting the range of $\boldsymbol{w}$ to fully bound the penalty term, and partially guaranteeing stability,
- Bounded ∇f_t: Restricting the range of $\boldsymbol{w}$ isn't enough, because it's possible that the gradient is so large that taking a small step from $\boldsymbol{w}_t$ to $\boldsymbol{w}_{t+1}$ leads to large $f(\boldsymbol{w}_t) - f(\boldsymbol{w}_{t+1})$.

Remark 3.2.3. Consider the following questions:

1. *Can we choose any strongly convex regularizer?*

   No, there's a trade-off here: if it's too strong, then the stability term will be small, while the penalty term will be large (small range of $\boldsymbol{w}$ will lead to large B_ψ); if it's too weak, the penalty term will be small, but it has less effect in stability. Besides, we should also choose the one that fully exploits the domain of the problem.

2. *What are the common choices of regularizer?*

   - Euclidean regularizer: $\psi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$ is 1-strongly convex w.r.t. $\|\cdot\|_2$ over $\mathbb{R}^d$
   - Negative Entropical regularizer: $\psi(\boldsymbol{w}) = \sum_{i=1}^d w_i \ln w_i$ is 1-strongly convex w.r.t. $\|\cdot\|_1$ over the d-dimensional probability simplex Δ^{d-1}.

3. *Why the abovementioned regularizers focus on the current decision only, i.e. $\psi(\boldsymbol{w})$, but not $\psi(\boldsymbol{w}_{t-1}, \boldsymbol{w}_t)$ since stability is crucial?*

   It's generally harder to solve those "time-coupling" cases, which is called the OCO with switching cost problem.

3.3 Online Gradient Descent

3.3.1 Unconstrained Domain

The *Online Gradient Descent* (*OGD*) algorithm is an instance of FTRL, instantiated by the Euclidean regularizer $\psi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$. Throughout this subsection, we consider an OLO with $f_t(\boldsymbol{w}) = \langle \boldsymbol{z}_t, \boldsymbol{w}\rangle$ and $\mathcal{C} = \mathbb{R}^n$, then

$$\boldsymbol{w}_{t+1} \coloneqq \arg\min_{\boldsymbol{w}} \sum_{i=1}^{t} \langle \boldsymbol{z}_t, \boldsymbol{w}\rangle + \frac{1}{2\eta}||\boldsymbol{w}||_2^2.$$

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:

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

Algorithm 3 Online Gradient Descent

Input: $\boldsymbol{w}_1 = \boldsymbol{0}$, $\eta = \frac{B}{L\sqrt{2T}}$
for $t = 1, 2, ..., T$ **do**
  Output $\boldsymbol{w}_t \in \mathbb{R}^d$
  Receive $f_t : \mathbb{R}^d \to \mathbb{R}$
  Update $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$
end for

Theorem 3.3.1 (*Sublinear Regret of OGD*)**.** *Assume f_t is L_t-Lipschitz w.r.t. $\|\cdot\|_2$ with $L = \max_t\{L_t\}$, and $B = \max_{\boldsymbol{w}} \|\boldsymbol{w}\|_2$, then OGD with $\eta = \frac{B}{L\sqrt{2T}}$ enjoys the following regret bound:*

$$\mathrm{Regret}_T \leq BL\sqrt{2T} \in \mathcal{O}(BL\sqrt{T}).$$

Proof. By Theorem 3.2.1,

$$\mathrm{Regret}_T \leq \frac{B_\psi}{\eta} + \frac{\eta T L^2}{\sigma_\psi} = \frac{1}{\eta}(\max_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{2}\|\boldsymbol{x}\|_2^2 - \frac{1}{2}\|\boldsymbol{y}\|_2^2) + \frac{\eta T L^2}{\sigma_\psi}$$

where $\sigma_\psi = 1$, since $1/2\|\boldsymbol{w}\|_2^2$ is 1-strongly convex. Then

$$\mathrm{Regret}_T \leq \frac{1}{2\eta}B^2 + \eta T L^2,$$

and plugging in η completes the proof. ☐

Remark 3.3.1. One interesting observation is that regardless of the adversarial f_{t+1}, fully trusting f_t can chase the best fixed decision pretty well.

One may argue that it's surprising because it seems that all history information is stored in f_t, but wouldn't fully trusting f_t be risky since f_{t+1} can be very different from f_t? Partially agreed with that, but note that $\boldsymbol{w}_t$ is based on $\{\boldsymbol{w}_i\}_{i=1}^{t-1}$ as well, which carries the information of $\{f_i\}_{i=1}^{t-1}$.

Hence, although OGD has a naive form, it depends on the history information as well, as FTL and FTRL do.

3.3.2 Constrained Domain

If $\mathcal{C}$ is constrained, we need to perform a projection $\boldsymbol{w}_t = \Pi_{\mathcal{C}}(\boldsymbol{u}_t) = \arg\min_{\boldsymbol{w}\in\mathcal{C}} \|\boldsymbol{w} - \boldsymbol{u}_t\|_2$ additionally. However, we cannot directly apply the bound in Theorem 3.2.1, since we don't know if the following holds

$$\arg\min_{\boldsymbol{w}\in\mathcal{C}} \sum_{i=1}^{t} f_t(\boldsymbol{w}) + \frac{1}{\eta}\psi(\boldsymbol{w}) = \Pi_{\mathcal{C}}(\arg\min_{\boldsymbol{w}} \sum_{i=1}^{t} f_t(\boldsymbol{w}) + \frac{1}{\eta}\psi(\boldsymbol{w})).$$

We shall see that it indeed holds, but we need to introduce the Bregman divergence first.

Algorithm 4 Online Gradient Descent

Input: $\boldsymbol{w}_1 \in \mathcal{C}$, $\eta = \frac{B}{L\sqrt{2T}}$
for $t = 1, 2, ..., T$ **do**
  Output $\boldsymbol{w}_t \in \mathcal{C}$
  Receive $f_t : \mathcal{C} \to \mathbb{R}$
  Compute $\boldsymbol{u}_{t+1} = \boldsymbol{w}_t - \eta \nabla f_t(\boldsymbol{w}_t)$
  Update $\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathcal{C}} \|\boldsymbol{w} - \boldsymbol{u}_{t+1}\|_2$
end for

Definition 3.3.1 (*Bregman Divergence*)**.** *Let $\psi : \mathcal{C} \to \mathbb{R}$ be strictly convex and differentiable. The Bregman divergence of ψ, $B_\psi : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ is a distance measure of two points w.r.t. ψ, where*

$$B_\psi(\boldsymbol{w}\|\boldsymbol{u}) = \psi(\boldsymbol{w}) - \underbrace{(\psi(\boldsymbol{u}) + \langle \nabla\psi(\boldsymbol{u}), \boldsymbol{w} - \boldsymbol{u}\rangle)}_{L_{\boldsymbol{u}}(\boldsymbol{w})}.$$

Remark 3.3.2. Euclidean Distance $\frac{1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|_2^2$ is an instance of Bregman distance, where $\psi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$.

Remark 3.3.3. We can see that extra affine functions won't affect the Bregman divergence. Consider $\phi(\boldsymbol{w}) = \psi(\boldsymbol{w}) + \langle \boldsymbol{z}, \boldsymbol{w}\rangle$, we have

$$\begin{aligned} B_\phi(\boldsymbol{w}\|\boldsymbol{u}) &= \phi(\boldsymbol{w}) - \phi(\boldsymbol{u}) - \langle \nabla\phi(\boldsymbol{u}), \boldsymbol{w} - \boldsymbol{u}\rangle \\ &= \psi(\boldsymbol{w}) + \langle \boldsymbol{z}, \boldsymbol{w}\rangle - \psi(\boldsymbol{u}) - \langle \boldsymbol{z}, \boldsymbol{u}\rangle - \langle \nabla\psi(\boldsymbol{u}) - \boldsymbol{z}, \boldsymbol{w} - \boldsymbol{u}\rangle = B_\psi(\boldsymbol{w}\|\boldsymbol{u}). \end{aligned}$$

Lemma 3.3.1. *To apply the bound in Theorem 3.2.1, we show that for strictly convex $f : \mathbb{R}^d \to \mathbb{R}$, first denote*

- $\boldsymbol{y} = \arg\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$,
- $\tilde{\boldsymbol{y}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$,
- $\boldsymbol{y}' = \arg\min_{\boldsymbol{x} \in \mathcal{C}} B_f(\boldsymbol{x}\|\tilde{\boldsymbol{y}})$.

then we have $\boldsymbol{y}' = \boldsymbol{y}$.

Proof. By definition, we have $f(\boldsymbol{y}) \leq f(\boldsymbol{y}')$, then we're left to show $f(\boldsymbol{y}') \leq f(\boldsymbol{y})$ by strict convexity. Note that $\nabla f(\tilde{\boldsymbol{y}}) = \boldsymbol{0}$, then we have

$$f(\boldsymbol{y}') - f(\tilde{\boldsymbol{y}}) = B_f(\boldsymbol{y}'\|\tilde{\boldsymbol{y}}) \leq B_f(\boldsymbol{y}\|\tilde{\boldsymbol{y}}) = f(\boldsymbol{y}) - f(\tilde{\boldsymbol{y}}),$$

and we finish the proof. If we assume linear loss, $f(\boldsymbol{w}) = \sum_{i=1}^t \langle \boldsymbol{z}_i, \boldsymbol{w}\rangle + \frac{1}{\eta}\psi(\boldsymbol{w})$, and therefore $B_f = B_\psi$. Now we can apply the bound in Theorem 3.2.1 directly. $\square$

3.4 Hedge

The *Hedge* Algorithm is another well-known instance of FTRL, instantiated by the negative entropy regularizer $\psi(\boldsymbol{w}) = \sum_{i=1}^d w(i) \ln w(i)$. We consider linear loss $f_t(\boldsymbol{w}) = \langle \boldsymbol{z}_t, \boldsymbol{w} \rangle$ and $\mathcal{C} = \Delta^{d-1}$, then

$$\boldsymbol{w}_{t+1} \coloneqq \arg\min_{\boldsymbol{w}} \sum_{i=1}^t \langle \boldsymbol{z}_t, \boldsymbol{w} \rangle + \frac{1}{\eta} \sum_{i=1}^d w_i \ln w_i.$$

Solving the unconstrained problem yields $w_{t+1,i} = \exp(-\eta \sum_{j=1}^t z_{t,i} - 1)$, and to make it feasible, we trivially set for all $i \in [d]$,

$$w_{t+1,i} = \frac{\exp(-\eta \sum_{j=1}^t z_{j,i})}{\sum_{i=1}^d \exp(-\eta \sum_{j=1}^t z_{j,i})} = \frac{w_{t,i} \cdot \exp(-\eta z_{t,i})}{\sum_{i=1}^d w_{t,i} \exp(-\eta z_{t,i})},$$

and this is called the Hedge algorithm. We can show that this is optimal within Δ^{d-1} using KKT conditions.

Algorithm 5 Hedge

Input: $\boldsymbol{w}_1 = (1/d, \ldots, 1/d) \in \Delta^{d-1}$, $\eta = \frac{B}{L\sqrt{2T}}$
for $t = 1, 2, ..., T$ **do**
  Output $\boldsymbol{w}_t \in \Delta^{d-1}$
  Receive $f_t : \Delta^{d-1} \to \mathbb{R}$
  Update $w_{t+1,i} = \frac{w_{t,i} \cdot \exp(-\eta z_{t,i})}{\sum_{i=1}^d w_{t,i} \exp(-\eta z_{t,i})}$, $\forall i \in [d]$
end for

Theorem 3.4.1 (*Sublinear Regret of Hedge*)**.** *Assume f_t is L_t-Lipschitz with $L = \max_t\{L_t\}$ w.r.t. $\|\cdot\|_\infty$, and $B = \max_{\boldsymbol{w}} \|\boldsymbol{w}\|_2$, then Hedge with $\eta = \sqrt{\frac{\ln d}{TL^2}}$ enjoys the following regret bound:*

$$\text{Regret}_T \leq L\sqrt{\ln d \cdot T} \in \mathcal{O}(L\sqrt{\ln d \cdot T}).$$

Proof. By Theorem 3.2.1,

$$\text{Regret}_T \leq \frac{B_\psi}{\eta} + \frac{\eta T L^2}{\sigma_\psi}$$

where $\sigma_\psi = 1$, and $B_\psi = \ln d$, incurred by $\arg\max_{\boldsymbol{w} \in \Delta^{d-1}} = (0, \ldots, 0, 1)$ and $\arg\min_{\boldsymbol{w} \in \Delta^{d-1}} (1/d, \ldots, 1/d)$. Then

$$\text{Regret}_T \leq \frac{1}{\eta} \ln d + \eta T L^2,$$

and plugging in η completes the proof. □

3.5 Doubling Trick

Among all algorithms that we've discussed and achieve $\mathcal{O}(\alpha\sqrt{T})$ regret bound for some constant α, the learning rate η requires prior knowledge of the time horizon T. Here we'll use a trick called *Doubling Trick* to remove the dependency:

Algorithm 6 Doubling Trick

```
Input: O(α√T) algorithm A whose parameters depend on T
for m = 0, 1, 2, ... do
    Run A as if T = 2^m on rounds 2^(m−1) + 1, 2^(m−1) + 2, ..., 2^m
end for
```

Lemma 3.5.1 (*$\mathcal{O}(\sqrt{T})$ Maintainer*)**.** *Assume algorithm $\mathcal{A}$ achieves $\mathcal{O}(\alpha\sqrt{T})$ regret bound but its parameters depend on T, then using the Doubling Trick, the dependency can be removed, and the $\mathcal{O}(\sqrt{T})$ regret bound is maintained.*

Proof. For all $T \in \mathbb{N}$, m is at most $\lceil \log_2 T \rceil$, then

$$\text{Regret}_T \leq \sum_{m=0}^{\lceil \log_2 T \rceil} \alpha\sqrt{2^m} = \alpha \sum_{m=0}^{\lceil \log_2 T \rceil} (\sqrt{2})^m = \alpha \frac{(\sqrt{2})^{\lceil \log_2 T \rceil + 1} - 1}{\sqrt{2} - 1} \leq \alpha \frac{(\sqrt{2})^{\log_2 T + 2} - 1}{\sqrt{2} - 1} \in \mathcal{O}(\alpha\sqrt{T}).$$

$\square$

Remark 3.5.1. We may notice that the constant doesn't have to be 2, i.e., doesn't have to double for each time the true T exceeds the guessed T, asymptotically speaking.

Remark 3.5.2. Doubling trick doesn't necessarily maintain any regret bound, e.g. for $\mathcal{O}(\log_2 T)$,

$$\text{Regret}_T = \sum_{m=0}^{\lceil \log_2 T \rceil} \log_2 2^m = \sum_{m=0}^{\lceil \log_2 T \rceil} m = \frac{\lceil \log_2 T \rceil (\lceil \log_2 T \rceil + 1)}{2} \notin \mathcal{O}(\log_2 T).$$

In order to use this "guess-based" trick to remove the dependency for $\mathcal{O}(\log_2 T)$, the followings must be satisfied:

- $\sum_{m=0}^{\lceil f^{-1}(T) \rceil} \log_2 f(m) \in \mathcal{O}(\log_2 T)$
- f is strictly increasing

3.6 Online Mirror Descent

3.6.1 Bregman Divergence Review

Definition 3.6.1 (*Bregman Divergence*)**.** *Let $\psi : \mathcal{C} \to \mathbb{R}$ be strictly convex and differentiable. The Bregman divergence of ψ, $B_\psi : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ is a distance measure of two points w.r.t. ψ, where*

$$B_\psi(\boldsymbol{w} \| \boldsymbol{u}) = \psi(\boldsymbol{w}) - \underbrace{(\psi(\boldsymbol{u}) + \langle \nabla\psi(\boldsymbol{u}), \boldsymbol{w} - \boldsymbol{u} \rangle)}_{L_{\boldsymbol{u}}(\boldsymbol{w})}.$$

Remark 3.6.1. Euclidean Distance $\frac{1}{2}\|\boldsymbol{w} - \boldsymbol{u}\|_2^2$ is an instance of Bregman distance, where $\psi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$.

3.6.2 Online Mirror Descent

Recall that the update method of OGD is $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\nabla f_t(\boldsymbol{w}_t)$, which is equivalent to the following equation:

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathcal{C}} f_t(\boldsymbol{w}_t) + \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w} - \boldsymbol{w}_t \rangle + \underbrace{\frac{1}{2\eta}\|\boldsymbol{w} - \boldsymbol{w}_t\|^2}_{\text{Euclidean Distance}}.$$

If we replace Euclidean distance by the more general Bregman divergence $B_\psi(\boldsymbol{w}\|\boldsymbol{w}_t)$:

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathcal{C}} f_t(\boldsymbol{w}_t) + \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w} - \boldsymbol{w}_t \rangle + \frac{1}{\eta} D_\psi(\boldsymbol{w}\|\boldsymbol{w}_t),$$

we yield the following formula, which is the update method for *Online Mirror Descent* (*OMD*)

$$\boldsymbol{w}_{t+1} = (\nabla\psi)^{-1}(\nabla\psi(\boldsymbol{w}_t) - \eta\nabla f_t(\boldsymbol{w}_t)),$$

where $\nabla\psi$ is known as the mirror map.

4 Experts Problem

4.1 Problem Setting

Now we introduce the well-known Experts problem. There're d experts in total, at each day t, each expert will make a prediction about the weather.

4.2 Hedge

The Hedge algorithm fits perfectly the problem setting, and we can get a $\mathcal{O}(L_\infty \sqrt{\ln d \cdot T})$ bound.

4.3 OGD

Nothing prevents us from using OGD, and we'll use Theorem 3.2.1. Firstly with B_ψ,

$$B_\psi(\boldsymbol{w}) = (1/2 \max_{\boldsymbol{w} \in \Delta^{d-1}} \|\boldsymbol{w}\|_2^2 - 1/2 \min_{\boldsymbol{w} \in \Delta^{d-1}} \|\boldsymbol{w}\|_2^2) = 1/2 \cdot (1 - 1/d),$$

where $\arg\max_{\boldsymbol{w} \in \Delta^{d-1}} = (0, 0, \ldots, 1)$ and $\arg\min_{\boldsymbol{w} \in \Delta^{d-1}} (1/d, \ldots, 1/d)$. Regarding the second part, we need to bound the gradient w.r.t. the same norm, say $\|\cdot\|_\infty$. By definition,

$$\|\boldsymbol{g}_t\|_2 \leq \sqrt{d} \|\boldsymbol{g}_t\|_\infty \leq \sqrt{d} L_\infty$$

5 Lower Bound of OCO

6 References