Regularized Algorithms for Online Optimization and Learning

CS245: Online Optimization and Learning

Xin Liu
SIST, ShanghaiTech University

Online Gradient Descent (OGD)

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.
For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \prod_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t))$.

The intuition of OGD is to solve "trust region optimization":

$$\min_{x \in \mathcal{K}} f_t(x_t) + \langle x - x_t, \nabla f_t(x_t) \rangle$$
$$\text{s.t. } \|x - x_t\| \leq \delta.$$

Review of Online Gradient Descent

Online Gradient Descent (OGD)

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.
For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \prod_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t))$.

The intuition of OGD is to minimize the first order approximation + regularization with ℓ_2 norm:

$$\hat{f}_{t+1}(x) = f_t(x_t) + \langle x - x_t, \nabla f_t(x_t) \rangle + \frac{1}{2\eta_t} \|x - x_t\|^2.$$

which is equavilent to

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \langle x, \nabla f_t(x_t) \rangle + \frac{1}{2\eta_t} \|x - x_t\|^2.$$

Bregman Divergence

Definition 1 (Bregman Divergence)

Let $\psi : X \to R$ be strictly convex and continuously differentiable function. The Bregman divergence w.r.t. ψ is B_ψ is defined as

$$B_\psi(x; y) = \psi(x) - \psi(y) - \langle x - y, \nabla\psi(y)\rangle.$$

If ψ is twice differentiable, and by Taylor theorem

$$B_\psi(x; y) = \langle x - y, \nabla^2\psi(z)(x - y)\rangle,$$

where z is a point between x and y.

Recall $\psi(\cdot)$ is α-strongly convex, we have a global property

$$B_\psi(x; y) \geq \frac{\alpha}{2}\|x - y\|^2.$$

Bregman Divergence - Examples

Let $\psi(x) = \frac{1}{2}\|x\|^2$, and the Bregman Divergence is

$$B_\psi(x;y) = \frac{1}{2}\|x - y\|^2$$

Let $\psi(x) = \sum_{i=1}^{d} x_i \log x_i$, with x being in a probability simplex, and the Bregman Divergence is

$$B_\psi(x;y) = \mathrm{KL}(x\|y).$$

Bregman Divergence - properties

The properties of Bregman divergence:

- Non-negative

$$B_\psi(x; y) \geq 0.$$

- "Non"-symmetric

$$B_\psi(x; y) \neq B_\psi(y; x).$$

- Three points identity:

$$B_\psi(z; x) + B_\psi(x; y) - B_\psi(z; y) = \langle \nabla\psi(y) - \nabla\psi(x), z - x \rangle.$$

Online Mirrored Descent

Online gradient descent is

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \langle x, \nabla f_t(x_t) \rangle + \frac{1}{2\eta_t}\|x - x_t\|^2.$$

Just change the “distance” metric to Bregman divergence w.r.t ψ, and we have

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \langle x, \nabla f_t(x_t) \rangle + \frac{1}{\eta_t} B_\psi(x; x_t).$$

If $\mathcal{K}$ is $\mathbb{R}^d$, let $\psi(x) = \frac{1}{2}\|x\|^2$ gives us online gradient descent algorithm.

If $\mathcal{K}$ is a probability simplex, let $\psi(x) = \sum_{i=1}^d x_i \log x_i$ gives us any algorithm?

Online Mirrored Descent (OMD)

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x \in \mathcal{K}} \; \langle x, \nabla f_t(x_t) \rangle + \frac{1}{\eta_t} B_\psi(x; x_t)$.

An alternative update is

$$
\begin{aligned}
y_{t+1} &= \arg\min_{x \in \mathbb{R}^d} \; \langle x, \nabla f_t(x_t) \rangle + \frac{1}{\eta_t} B_\psi(x; x_t) \\
x_{t+1} &= \arg\min_{x \in \mathcal{K}} B_\psi(x; y_{t+1})
\end{aligned}
$$

Online Mirrored Descent - Regret

Recall the regret of online gradient descent is $O(\sqrt{T})$. How about the regret of online mirrored descent?

Theorem 2

Let ψ be α-strongly convex function. Consider a fixed learning rate $\eta_t = \eta$. Online mirrored descent algorithm achieves

$$Regret(T) \leq \frac{B_\psi(x^*, x_1)}{\eta} + \frac{1}{2\alpha}\sum_{t=1}^{T} \eta \|\nabla f_t(x_t)\|^2.$$

OMD achieves $O(\sqrt{T})$ regret if:

- The feasible set and gradients are bounded.
- Learning rate is fixed with $O(1/\sqrt{T})$.
- Time varying learning rate $O(1/\sqrt{t})$ or adaptive learning rate also work (verify by yourself).

Online Mirrored Descent - Proof

We use a "potential/Lyapunov drift" style of analysis: define

$$\begin{aligned}\phi_t =& B_\psi(x^*; x_t)\\ =& \psi(x^*) - \psi(x_t) - \langle x^* - x_t, \nabla\psi(x_t)\rangle,\end{aligned}$$

and study the drift

$$\begin{aligned}\phi_{t+1} - \phi_t =& B_\psi(x^*; x_{t+1}) - B_\psi(x^*; x_t)\\ =& - B_\psi(x_{t+1}; x_t) + \langle \nabla\psi(x_t) - \nabla\psi(x_{t+1}), x^* - x_{t+1}\rangle\end{aligned}$$

Online Mirrored Descent - Proof

Online Mirrored Descent - An Alternative Proof

We have the following lemma that make our analysis simple[1]

Lemma 3 (A pushback lemma)

Suppose x_{t+1} minimizes the function $F(x)$ such that

$$F(x) := \langle x, \nabla f_t(x_t) \rangle + \frac{1}{\eta} B(x; x_t),$$

For any x, we have

$$F(x_{t+1}) \leq F(x) - \frac{1}{\eta} B(x; x_{t+1}).$$

[1] X. Wei, et al. Online Primal-Dual Mirror Descent under Stochastic Constraints. Sigmetrics 2020.

Online Mirrored Descent - An Alternative Proof

Why is called Mirrored descent?

Definition 4 (Fenchel Conjugate)

The Fenchel conjugate of a function f is

$$f^*(y) := \sup_{x \in \mathcal{K}} \langle y, x \rangle - f(x).$$

Theorem 5

The update of online mirrored descent

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \langle x, \nabla f_t(x_t) \rangle + \frac{1}{\eta_t} B_\psi(x; x_t)$$

is equivalent to

$$x_{t+1} = \nabla \psi^*_{\mathcal{K}}(\nabla \psi_{\mathcal{K}}(x_t) - \eta_t \nabla f_t(x_t)).$$

Let's consider the case of $\psi(x) = \frac{1}{2}\|x\|^2$, can we reduce it to online gradient descent?

Theorem 5 – Proof

By definition of online mirror descent, we have

$$\begin{aligned}
x_{t+1} &= \arg\min_{x \in \mathcal{K}} \ \langle x, \nabla f_t(x_t) \rangle + \frac{1}{\eta_t} B_\psi(x; x_t) \\
&= \arg\min_{x \in \mathcal{K}} \ \eta_t \langle x, \nabla f_t(x_t) \rangle + B_\psi(x; x_t) \\
&= \arg\min_{x \in \mathcal{K}} \ \eta_t \langle x, \nabla f_t(x_t) \rangle + \psi(x) - \langle x, \nabla \psi(x_t) \rangle \\
&= \arg\min_{x \in \mathcal{K}} \ \langle x, \eta_t \nabla f_t(x_t) - \nabla \psi(x_t) \rangle + \psi(x) \\
&= \arg\max_{x \in \mathcal{K}} \ \langle x, \nabla \psi(x_t) - \eta_t \nabla f_t(x_t) \rangle - \psi(x)
\end{aligned}$$

Let's define $y = \nabla \psi(x_t) - \eta_t \nabla f_t(x_t)$, and we have

$$x_{t+1} = \arg\max_{x \in \mathcal{K}} \ \langle x, y \rangle - \psi(x).$$

Let's first consider $\mathcal{K} = \mathbb{R}^d$. Note x_{t+1} is maximizing

$$\langle x, y\rangle - \psi(x),$$

we have

$$\begin{aligned}\nabla\psi^*(y) =& \frac{\partial\left(\max_x \langle x, y\rangle - \psi(x)\right)}{\partial y}, \\ =& \frac{\partial\left(\langle x_{t+1}, y\rangle - \psi(x_{t+1})\right)}{\partial y} \\ =& x_{t+1},\end{aligned}$$

which means

$$x_{t+1} = \nabla\psi^*(y) = \nabla\psi^*(\nabla\psi(x_t) - \eta_t \nabla f_t(x_t)).$$

We are done. Please verify the case of the general $\mathcal{K}$.

Why is called Mirrored descent?

Let's understand online mirrored descent ($\mathcal{K} = \mathbb{R}^d$)

$$x_{t+1} = \nabla\psi^*(\nabla\psi(x_t) - \eta_t \nabla f_t(x_t))$$

in three steps:

- Mirror x_t from primal space to dual $\theta_t = \nabla\psi(x_t)$.
- Take gradient descent in dual space
$$\theta_{t+1} = \theta_t - \eta_t \nabla f_t(x_t).$$
- Mirror θ_{t+1} back to $\nabla\psi^*(\theta_{t+1})$.

Review of Expert problem

Expert problem:

Initialization: N experts/models.

For each day $t = 1, \cdots, T$:

- **Learner:** Obtain predictions from N experts/models and sample an expert i from a probability simplex x_t.
- **Environment:** Observe the loss of each model $\ell_t \in [0,1]^N$.

Objective: Find the best expert in hindsight, which is equivalent to minimize regret:

$$\mathcal{R}(T) := \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i) - \sum_{t=1}^{T} \ell_t(i^*)\right] = \sum_{t=1}^{T} \langle x_t, \ell_t \rangle - \sum_{t=1}^{T} \langle x^*, \ell_t \rangle$$

Expert problem: Hedge

Hedge - "weighted" version:

Initialization: $w_1(i) = 1, \forall i \in [N]$.

For each day $t = 1, \cdots, T$:

- **Learner:** Sample an expert i : $p_t(i) = w_t(i) / \sum_i w_t(i)$.
- **Environment:** Observe the error $\ell_t \in [0,1]^N$.
- **Update:** $w_{t+1} = w_t \cdot e^{-\eta \ell_t(i)}, \forall i \in [N]$.

Hedge - "prob" version:

Initialization: $x_1 = [1/d, \cdots, 1/d]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** Sample an expert i according to x_t.
- **Environment:** Observe the error $\ell_t \in [0,1]^N$.
- **Update:** $x_{t+1,i} = x_{t,i} e^{-\eta \ell_t(i)} / \sum_{i=1}^{d} x_{t,i} e^{-\eta \ell_t(i)}, \forall i \in [N]$.

Exponentiated Gradient – Hedge

Exponentiated Gradient:

Initialization: $x_1 = [1/d, \cdots, 1/d]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the loss $f_t(\cdot)$.
- **Update:** $x_{t+1,i} = x_{t,i} e^{-\eta \nabla f_{t,i}(x_t)} / \sum_{i=1}^{d} x_{t,i} e^{-\eta \nabla f_{t,i}(x_t)}$.

How it is related to Hedge - "prob" version?

- No sampling operator from x_t.
- The loss is $f_t(x_t) = \langle x_t, \ell_t \rangle$.
- Regret is equivalent to the "expected" regret of Hedge!

Online Mirrored Descent:

Initialization: $x_1 = [1/d, \cdots, 1/d]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** submit x_t.
- **Environment:** Observe the loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{\mathcal{K}} \ \langle x, \nabla f_t(x_t)\rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

Since x in the prob simplex, can we try $\psi(x) = \sum_{i=1}^d x_i \log x_i$ in the Bregman divergence and show x_{t+1} is equivalent to that in Exponentiated Gradient?

Online Mirrored Descent:

Initialization: $x_1 = [1/d, \cdots, 1/d]$ and η.
For each day $t = 1, \cdots, T$:

- **Learner:** submit x_t.
- **Environment:** Observe the loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{\mathcal{K}} \ \langle x, \nabla f_t(x_t)\rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

Since x in the prob simplex, can we try $\psi(x) = \sum_{i=1}^d x_i \log x_i$ in the Bregman divergence and show x_{t+1} is equivalent to that in the Exponentiated Gradient:

$$x_{t+1,i} = \frac{x_{t,i} e^{-\eta \nabla f_{t,i}(x_t)}}{\sum_{i=1}^d x_{t,i} e^{-\eta \nabla f_{t,i}(x_t)}}.$$

Exponentiated Gradient as Online Mirrored Descent

The update of Bragman divergence

$$\min_{x \in \mathcal{K}} \eta \langle x, \nabla f_t(x_t) \rangle + \sum_{i=1}^{d} x_i \log \frac{x_i}{x_{t,i}}$$

$$\text{s.t. } \sum_{i=1}^{d} x_i = 1, \quad x_i \geq 0.$$

Let's consider (partial) Lagrangian function:

$$L(x, \lambda) = \eta \langle x, \nabla f_t(x_t) \rangle + \sum_{i=1}^{d} x_i \log \frac{x_i}{x_{t,i}} + \lambda (1 - \sum_{i=1}^{d} x_i)$$

Exponentiated Gradient as Online Mirrored Descent

Hedge as Online Mirrored Descent

Hedge as Online Mirrored Descent:

Initialization: $x_1 = [1/d, \cdots, 1/d]$ and η_t.
For each day $t = 1, \cdots, T$:

- **Learner:** Sample an expert i from x_t.
- **Environment:** Observe the error $\ell_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{\mathcal{K}} \ \langle x, \ell_t \rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

Hedge $\longrightarrow$ Exponentiated Gradient $\longrightarrow$ OMD!

OMD is a strong and general framework to design online algorithms!

Hedge as Online Mirrored Descent – Regret

Theorem 6 (Restate Theorem 2)

Let ψ be α-strongly convex function in B_ψ. Let fixed learning rate $\eta_t = \eta$. Online mirrored descent algorithm achieves

$$Regret(T) \leq \frac{B_\psi(x^*, x_1)}{\eta} + \frac{\eta}{2\alpha} \sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2.$$

In Hedge, we have

- $\psi(x) = \sum_{i=1}^{d} x_i \log x_i$ is 1-strongly convex,
- $B_\psi(x^*, x_1) = \sum_{i=1}^{d} x_i^* \log \frac{x_i^*}{x_{1,i}} \leq \log N$,

which implies the regret of Hedge is

$$\text{Regret}(T) = O(\sqrt{T \log N}).$$

Consider a linear function

$$f_t(x) = \langle \ell_t, x \rangle.$$

Online Learning with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Learner:** Given a prediction $\hat{\ell}_t$ and submit x_t.
- **Environment:** Observe the cost ℓ_t.

Online Learning with Prediction

Consider a linear function

$$f_t(x) = \langle \ell_t, x \rangle.$$

Online Learning with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Learner:** Given a prediction $\hat{\ell}_t$ and submit x_t.
- **Environment:** Observe the cost ℓ_t.

LI predict weather tomorrow in shanghai

As an AI language model, I cannot predict the weather with certainty. However, based on historical weather data and current trends, I can suggest that the weather in Shanghai tomorrow is likely to be partly cloudy with a chance of scattered showers. The temperature is expected to range from around 10 to 18 degrees Celsius, with a moderate breeze from the east-northeast. However, I would recommend checking the latest weather forecast for Shanghai to get the most accurate information.

LI list three top papers on this topic with just titles

Sure, here are three top papers on online learning and optimization with just their titles:

1. "Online Learning and Online Convex Optimization" by Elad Hazan.
2. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization" by John Duchi, Elad Hazan, and Yoram Singer.
3. "Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization" by Peter Bartlett, Elad Hazan, and Alexander Rakhlin.

Consider a linear function

$$f_t(x) = \langle \ell_t, x \rangle.$$

Online Learning with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Learner:** Given a prediction $\hat{\ell}_t$ and submit x_t.
- **Environment:** Observe the cost ℓ_t.

How to utilize the prediction to improve the online learning algorithms?

- For perfect predictions $\hat{\ell}_t = \ell_t$, the regret is smaller than $O(\sqrt{T})$?
- For bad predictions, the regret should not be worse than $O(\sqrt{T})$!

Online Learning with Prediction

Online Learning with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta\}$.
For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the cost ℓ_t.
- **Prediction:** The cost $\hat{\ell}_{t+1}$.
- **Update:** $x_{t+1} = \mathrm{Alg}(x_1, \cdots, x_t, \ell_1, \cdots, \ell_t, \hat{\ell}_{t+1})$.

$\mathrm{Alg}(x_1, \cdots, x_t, \ell_1, \cdots, \ell_t, \hat{\ell}_{t+1})$ could be $\mathrm{Alg}(x_t, \ell_t, \hat{\ell}_{t+1})$ like online gradient/mirrored descent:

$$x_{t+1} = \arg\min_{x \in \mathbb{R}^d} \langle x, \ell_t \rangle + \frac{1}{\eta} B_{\psi}(x; x_t)$$

How to incorporate the prediction $\hat{\ell}_{t+1}$?

Online Learning with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta\}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the cost ℓ_t.
- **Prediction:** The cost $\hat{\ell}_{t+1}$.
- **Update:** $x_{t+1} = \mathrm{Alg}(x_1, \cdots, x_t, \ell_1, \cdots, \ell_t, \hat{\ell}_{t+1})$.

Online gradient/mirrored descent:

$$y_{t+1} = \arg\min_{y \in \mathbb{R}^d} \ \langle y, \ell_t \rangle + \frac{1}{\eta} \ B_\psi(y; y_t)$$

How to incorporate the prediction $\hat{\ell}_{t+1}$?

Online Mirrored Descent with Prediction

Online Learning with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta\}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the cost ℓ_t.
- **Prediction:** The cost $\hat{\ell}_{t+1}$.
- **Update:** $x_{t+1} = \text{Alg}(x_1, \cdots, x_t, \ell_1, \cdots, \ell_t, \hat{\ell}_{t+1})$.

Online gradient/mirrored descent with prediction:

$$y_{t+1} = \arg\min_{y \in \mathbb{R}^d} \ \langle y, \ell_t \rangle + \frac{1}{\eta} \ B_\psi(y; y_t)$$

$$x_{t+1} = \arg\min_{x \in \mathbb{R}^d} \ \langle x, \hat{\ell}_{t+1} \rangle + \frac{1}{\eta} \ B_\psi(x; y_{t+1})$$

Online Mirrored Descent with Prediction

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta\}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the loss ℓ_t.
- **Prediction:** The cost $\hat{\ell}_{t+1}$.
- **Update:** $y_{t+1} = \arg\min_{y \in \mathbb{R}^d} \ \langle y, \ell_t \rangle + \frac{1}{\eta} \ B_\psi(y; y_t)$

  $x_{t+1} = \arg\min_{x \in \mathbb{R}^d} \ \langle x, \hat{\ell}_{t+1} \rangle + \frac{1}{\eta} \ B_\psi(x; y_{t+1})$

Intuition:

- Online mirrored descent guarantees "not too bad" even with unreliable predictions.
- Decrease the cost further if $\hat{\ell}_{t+1}$ is reliable.

Online Mirrored Descent with Prediction – Regret

The regret of OMD with prediction is as follows. [2]

Theorem 7

Let ψ be 1-strongly convex function in B_ψ. Let fixed learning rate $\eta_t = \eta$. Given a prediction sequence of $\{\hat{\ell}_t\}$, online mirrored descent achieves

$$Regret(T) \leq \frac{B(x^*, x_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\hat{\ell}_t - \ell_t\|^2.$$

"Almost" the best of two worlds:

- If the predictions are "perfect", the regret is constant!
- If the predictions are "bad", the regret can be $O(\sqrt{T})$.
- If the predictions are "good", the regret can be $o(\sqrt{T})$.

[2] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. COLT, 2013

Online Mirrored Descent with Prediction – Proof

According to the pushback lemma, suppose x_{t+1} minimizes the function $F(x)$ such that

$$F(x) := \langle x, \ell_t \rangle + \frac{1}{\eta} B(x; x_t).$$

For any x, we have

$$F(x_{t+1}) \leq F(x) - \frac{1}{\eta} B(x; x_{t+1}).$$

Therefore, we have

$$\eta \langle x_{t+1}, \ell_t \rangle + B(x_{t+1}; x_t) \leq \eta \langle x^*, \ell_t \rangle + B(x^*; x_t) - B(x^*; x_{t+1}).$$

which implies

$$\eta \langle x_t - x^*, \ell_t \rangle + \eta \langle x_{t+1} - x_t, \ell_t \rangle + B(x_{t+1}; x_t) \leq B(x^*; x_t) - B(x^*; x_{t+1}).$$

Online Mirrored Descent with Prediction – Proof

Step one:

$$y_{t+1} = \arg\min_{y \in \mathbb{R}^d} \ \langle y, \ell_t \rangle + \frac{1}{\eta} \ B_\psi(y; y_t).$$

By pushback lemma, we have

$$\eta\langle y_{t+1}, \ell_t \rangle + B(y_{t+1}; y_t) \le \eta\langle x^*, \ell_t \rangle + B(x^*; y_t) - B(x^*; y_{t+1}).$$

Step two:

$$x_t = \arg\min_{x \in \mathbb{R}^d} \ \langle x, \hat{\ell}_t \rangle + \frac{1}{\eta} \ B_\psi(x; y_t).$$

By pushback lemma, we have

$$\eta\langle x_t, \hat{\ell}_t \rangle + B(x_t; y_t) \le \eta\langle x, \hat{\ell}_t \rangle + B(x; y_t) - B(x; x_t).$$

Online Mirrored Descent with Prediction – Proof

Why Online Gradient/Mirrored Descent?

Online Learning Algorithm

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the cost ℓ_t.
- **Update:** $x_{t+1} = \mathrm{Alg}(x_1, \cdots, x_t, \ell_1, \cdots, \ell_t)$.

We design online learning algorithms to achieve small regret:

- Online gradient/mirrored descent is based on the current x_t and ℓ_t as
$$\mathrm{Alg}(x_t, \ell_t).$$
- Can we use all information to design online algorithms?
$$x_{t+1} = \mathrm{Alg}(x_1, \cdots, x_t, \ell_1, \cdots, \ell_t).$$

Follow-The-Leader (FTL) Algorithm

Follow-The-Leader (FTL) Algorithm

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \sum_{s=1}^{t} f_s(x)$.

Intuition of Follow-The-Leader (FTL) algorithm:

- A batch/offline learning problem to use all history info.
- Minimize the "regret" for the next round

$$\sum_{s=1}^{t} f_s(x_{t+1}) \leq \sum_{s=1}^{t} f_s(x^*).$$

Follow-The-Leader (FTL) Algorithm

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \sum_{s=1}^{t} f_s(x)$.

Follow-The-Leader (FTL) algorithm seems to work!?

What is the regret of FTL algorithms?

$$\mathcal{R}(T) := \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x).$$

Follow-The-Leader (FTL) Algorithm – Regret

Theorem 8

Under Follow-The-Leader algorithm, we have the sequence of actions $\{x_t\}$ which satisfies

$$\begin{aligned}\mathcal{R}(T) &:= \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \\ &\leq \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x_{t+1}).\end{aligned}$$

Intuitively, we have a small regret if it is "stable":

$$x_t \text{ is close to } x_{t+1}.$$

Follow-The-Leader (FTL) Algorithm – Proof

Follow-The-Leader (FTL) Algorithm – Caveat

Follow-The-Leader (FTL) Algorithm

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x \in \mathcal{K}} \sum_{s=1}^{t} f_s(x)$.

Let's consider a counter example as follows

$$\mathcal{K} = [-1, 1],$$
$$\{f_1, f_2, f_3, f_4, f_5, \cdots, f_T\} = \{0.5x, -x, x, -x, x, \cdots, x\}.$$

What is the regret of FTL algorithms?

Follow-The-Leader (FTL) Algorithm – Caveat

Follow-The-Regularized-Leader (FTRL) Algorithm

We need to make FTL algorithm stable:

$$\text{FTL} + \text{Regularization} = \text{FTRL}.$$

Follow-The-Regularized-Leader (FTRL) Algorithm

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \sum_{s=1}^{t} f_s(x) + R_{t+1}(x)$.

Intuition of Follow-The-Regularized-Leader:

- The regularization term $R_{t+1}(x)$ prevents x_{t+1} going too far from x_t.
- FTRL is FTL with the initial regularization $f_0(x) = R(x)$.

FTRL Algorithm – Regret

Let's consider the linear costs and the quadratic regularizar:

$$f_t(x) = \langle \ell_t, x \rangle, \forall t, \quad R(x) = \frac{1}{2\eta}\|x\|^2.$$

Theorem 9 (linear losses and quadratic regularizar)

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ $\|\nabla f_t(x)\| \leq G, \forall x \in \mathcal{K}$. *Under Follow-The-Regularized-Leader algorithm, we have the sequence of actions* $\{x_t\}$ *which satisfies*

$$\mathcal{R}(T) \leq DG\sqrt{2T}.$$

We recover the good result of $O(\sqrt{T})$, which is similar as online gradient descent.

We can also get similar result for a convex loss and other types of regulazizar.

FTRL Algorithm – Proof

FTRL and OMD Algorithms

Since FTRL and OMD both have regularization terms, any connection between these two algorithms?

- FTRL is

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \sum_{s=1}^{t} f_s(x) + R(x).$$

- OMD is

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \langle x, \nabla f_t(x) \rangle + \frac{1}{\eta} B_{\psi}(x; x_t).$$

Let's consider two examples corresponding to two type of gradient algorithms:

- Online gradient descent.
- Exponentiated gradient.

FTRL and OMD Algorithms

Let's consider the linear costs and the quadratic regularizar:

$$f_t(x) = \langle \ell_t, x \rangle, \forall t, \quad R(x) = \frac{1}{2\eta}\|x\|^2.$$

FTRL and OMD Algorithms

Let's consider the expert problem with linear costs and the negative entropy regularizar:

$$f_t(x) = \langle \ell_t, x \rangle, \forall t, \quad R(x) = \frac{1}{\eta} \sum_i x_i \log x_i.$$

FTRL and OMD Algorithms

FTRL with the linear losses and adaptive regularization are

$$\begin{aligned} x_{t+1} &= \arg\min_{x \in \mathcal{K}} \ \sum_{s=1}^{t} f_s(x) + R_{t+1}(x) \\ &= \arg\min_{x \in \mathcal{K}} \ \langle \sum_{s=1}^{t} \ell_s, x \rangle + R_{t+1}(x) \\ &= \arg\max_{x \in \mathcal{K}} \ \langle -\sum_{s=1}^{t} \ell_s, x \rangle - R_{t+1}(x) \end{aligned}$$

Recall the conjugate definition $f^*(y) = \sup_x \ \langle y, x \rangle - f(x)$.
Therefore, we have

$$x_{t+1} = \nabla R_{t+1}^* \left(-\sum_{s=1}^{t} \ell_s \right)$$

FTRL and OMD Algorithms

Let's define $\theta_{t+1} = -\sum_{s=1}^{t} \ell_s$ and $\theta_{t+1} = \theta_t - \ell_t$.
FTRL updates as

$$\begin{aligned} \theta_{t+1} &= \theta_t - \ell_t \\ x_{t+1} &= \nabla R^*_{t+1}\left(\theta_{t+1}\right) \end{aligned}$$

Recall OMD updates as

$$\begin{aligned} \theta_{t+1} &= \nabla \psi(x_t) - \eta_t \ell_t \\ x_{t+1} &= \nabla \psi^*\left(\theta_{t+1}\right) \end{aligned}$$

FTRL v.s. OMD:

- FTRL takes "gradient" directly in dual space. Unlike in OMD, it first "mirrors" from x_t to $\theta_t = \nabla\psi(x_t)$.
- FTRL treats losses equally & OMD weights losses by η_t.

Follow-The-Regularized-Leader Algorithm

Follow-The-Regularized-Leader (FTRL) Algorithm

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \sum_{s=1}^{t} f_s(x) + R_{t+1}(x)$.

We have already got the intuition on how the regularization helps stabilize the algorithm.

FTRL is a powerful framework to design online algorithms and the adaptive regulazier plays an important role.

- $R_t(x) = \sqrt{t}\|x\|^2$.
- $R_t(x) = \sqrt{t}\sum_i x_i \log x_i$.

FTRL Algorithm – Regret

Let's consider the convex costs $f_t(x)$ and the adaptive regularizar $R_t(x)$ that is "increasing" as time t and α_t-strongly convex.

Theorem 10 (convex losses and adaptive regularizar)

Assume $\|x - y\| \le D, \forall x, y \in \mathcal{K}$ $\|\nabla f_t(x)\| \le G, \forall x \in \mathcal{K}$. *Under Follow-The-Regularized-Leader algorithm, we have the sequence of actions* $\{x_t\}$ *which satisfies*

$$\mathcal{R}(T) \le R_{T+1}(x^*) - \min R_1(x) + \sum_{t=1}^{T} \frac{\|\nabla f_t\|^2}{2\alpha_t}.$$

We recover the good result of $O(\sqrt{T})$ (e.g., the regularizar $R_t(x) = \sqrt{t}\|x\|^2$). It is similar as FTRL with the fixed regularizar.

FTRL Algorithm – Proof

We want to study

$$\mathcal{R}(T) = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*).$$

Denote $F_t(x) = \sum_{s=1}^{t-1} f_s(x) + R_t(x)$ and we have

$$F_{T+1}(x^*) = \sum_{s=1}^{T} f_s(x^*) + R_{T+1}(x^*).$$

Therefore, we have

$$\mathcal{R}(T) = \sum_{t=1}^{T} f_t(x_t) - F_{T+1}(x^*) + R_{T+1}(x^*).$$

We need to connect $f_t(x_t)$ with $F_t(x_t)$.

FTRL Algorithm – Proof

We have

$$
\begin{aligned}
\mathcal{R}(T) &= \sum_{t=1}^{T} f_t(x_t) - F_{T+1}(x^*) + R_{T+1}(x^*) \\
&= \sum_{t=1}^{T} \left(F_t(x_t) - F_{t+1}(x_{t+1}) + f_t(x_t)\right) \\
&\quad + F_{T+1}(x_{T+1}) - F_1(x_1) - F_{T+1}(x^*) + R_{T+1}(x^*) \\
&= \sum_{t=1}^{T} \left(F_t(x_t) - F_{t+1}(x_{t+1}) + f_t(x_t)\right) \\
&\quad + F_{T+1}(x_{T+1}) - F_{T+1}(x^*) + R_{T+1}(x^*) - \min R_1(x)
\end{aligned}
$$

The key is to quantify $F_t(x_t) - F_{t+1}(x_{t+1}) + f_t(x_t)$.

FTRL Algorithm – Proof

Lemma 11 (One-step difference)

Let F_t be α_t-strongly convex function, FTRL algorithm has

$$F_t(x_t) - F_{t+1}(x_{t+1}) + f_t(x_t) \leq \frac{\|\nabla f_t\|^2}{2\alpha_t} + R_t(x_{t+1}) - R_{t+1}(x_{t+1}).$$

FTRL Algorithm – Proof

Optimistic FTRL

Optimistic Follow-The-Regularized-Leader (FTRL)

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Prediction:** The cost $\hat{f}_{t+1}(\cdot)$.
- **Update:**
  $x_{t+1} = \arg\min_{x \in \mathcal{K}} \sum_{s=1}^{t} f_s(x) + \hat{f}_{t+1}(x) + R_{t+1}(x)$.

Intuition:

- FTRL guarantees "not too bad" even with unreliable predictions.
- Decrease the cost further if $\hat{f}_{t+1}(\cdot)$ is reliable.

Optimistic FTRL – Regret

Theorem 12 (Optimistic FTRL)

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ $\|\nabla f_t(x)\| \leq G, \forall x \in \mathcal{K}$.
$R_t(x)$ *that is "increasing" as time* t *and* α_t*-strongly convex.*
Under Optimistic Follow-The-Regularized-Leader algorithm, we have the sequence of actions $\{x_t\}$ *which satisfies*

$$\mathcal{R}(T) \leq R_{T+1}(x^*) - \min R_1(x) + \sum_{t=1}^{T} \frac{\|\nabla f_t - \nabla \hat{f}_t\|^2}{2\alpha_t}.$$

As in OMD with prediction, we have a few observations:

- If the predictions are "perfect", the regret is constant!
- If the predictions are "bad", the regret can be $O(\sqrt{T})$.
- If the predictions are "good", the regret can be $o(\sqrt{T})$.

Optimistic FTRL – Proof

Online Learning with Delayed Feedback

Online Learning with Delayed Feedback

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_{t-d}(\cdot)$.
- **Update:** $x_{t+1} = \mathrm{Alg}(f_1, f_2, \cdots, f_{t-d})$.

A few examples:

- Subseasonal prediction: the prediction correct or not will be known in 2$\sim$6 weeks.
- Medical treatment: the treatment effective or not will be observed a few days or weeks.
- Dynamic pricing: the promotion working or not will be revealed a few days or weeks.

FTRL with Delayed Feedback

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_{t-d}(\cdot)$.
- **Update:** $x_{t+1} = \arg\min_{x\in\mathcal{K}} \ \sum_{s=1}^{t-d} f_s(x) + R_{t+1}(x)$.

Observations of FTRL with delayed feedback:

- Use all revealed feedback seen at time t.
- Large delay degrades the performance because of missing feedback $\sum_{s=t-d+1}^{t} f_s(x)$.

What is the regret of the algorithms?

Delay as Optimism in FTRL

Delay is "optimistism"!!!

Delay as Optimism in FTRL

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Prediction:** The cost $\hat{f}_{t+1}(\cdot) = -\sum_{s=t-d+1}^{t} f_s(x)$.
- **Update:**
  $x_{t+1} = \arg\min_{x \in \mathcal{K}} \ \sum_{s=1}^{t} f_s(x) + \hat{f}_{t+1}(x) + R_{t+1}(x)$.

Delayed FTRL $\longrightarrow$ Optimistic FTRL.

Optimistic FTRL is a powerful framework that can handle the prediction and delay!

Delayed FTRL – Regret

Theorem 13 (Delayed FTRL)

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ $\|\nabla f_t(x)\| \leq G, \forall x \in \mathcal{K}$.
$R_t(x)$ *that is "increasing" as time* t *and* α_t*-strongly convex.*
Under Follow-The-Regularized-Leader algorithm, we have the sequence of actions $\{x_t\}$ *which satisfies*

$$\mathcal{R}(T) \leq R_{T+1}(x^*) - \min R_1(x) + \sum_{t=1}^{T} \frac{\|\nabla f_t - \nabla \hat{f}_t\|^2}{2\alpha_t},$$

where $\nabla \hat{f}_t = -\sum_{s=t-d+1}^{t} \nabla f_s$.

The effect caused by the delay:

$$\|\nabla f_t\|^2 \longrightarrow \|\nabla f_t + \sum_{s=t-d+1}^{t} \nabla f_s\|^2.$$

Let $\alpha_t = O(1/\sqrt{(d+1)T})$. Delayed FTRL achieves the regret of $O(\sqrt{(d+1)T})$, where the delay hurts the regret!