

# Regularized Algorithms for Online Optimization and Learning

CS245: Online Optimization and Learning

Xin Liu  
SIST, ShanghaiTech University

# 2024 ACM A.M. Turing Award



Andrew Barto



Rich Sutton

Alan Turing: What we want is a machine that can learn from experience?

Answer: Reinforcement learning!

Keywords: Temporal difference learning, policy gradient methods,  
RL book, AlphaGo, ChatGPT, RLHF

# 2024 ACM A.M. Turing Award



Richard S. Sutton

FOLLOWING

Keen, Amii, and University of Alberta

Verified email at richsutton.com - [Homepage](#)

[artificial intelligence](#) [reinforcement learning](#) [machine learning](#) [cognitive science](#)  
[computer science](#)

| TITLE   | CITED BY | YEAR |
|---|----------|------|
| <a href="#">Reinforcement learning: An introduction</a><br>RS Sutton, AG Barto<br>MIT press 1 (1), 9-11   | 79694    | 1998 |
| <a href="#">Policy gradient methods for reinforcement learning with function approximation</a><br>RS Sutton, D McAllester, S Singh, Y Mansour<br>Advances in neural information processing systems 12             | 9288     | 1999 |
| <a href="#">Learning to predict by the methods of temporal differences</a><br>RS Sutton<br>Machine learning 3, 9-44   | 8383     | 1988 |
| <a href="#">Neuronlike adaptive elements that can solve difficult learning control problems</a><br>AG Barto, RS Sutton, CW Anderson<br>IEEE transactions on systems, man, and cybernetics 13 (5), 834-846         | 5301     | 1983 |
| <a href="#">Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning</a><br>RS Sutton, D Precup, S Singh<br>Artificial intelligence 112 (1-2), 181-211                          | 4731     | 1999 |
| <a href="#">Integrated architectures for learning, planning, and reacting based on approximating dynamic programming</a><br>RS Sutton<br>Proceedings of the International Conference on Machine Learning, 216-224 | 2326     | 1990 |

# 2024 ACM A.M. Turing Award



Richard S. Sutton

Keen, Amii, and University of Alberta  
Verified email at richsutton.com - [Homepage](#)

[artificial intelligence](#) [reinforcement learning](#) [machine learning](#) [cognitive science](#)  
[computer science](#)



| TITLE  | CITED BY | YEAR |
|--|----------|------|
| <a href="#">Asynchronous Stochastic Approximation and Average-Reward Reinforcement Learning</a><br>H Yu, Y Wan, RS Sutton<br>arXiv preprint arXiv:2409.03915                             |          | 2024 |
| <a href="#">On Convergence of Average-Reward Q-Learning in Weakly Communicating Markov Decision Processes</a><br>Y Wan, H Yu, RS Sutton<br>arXiv preprint arXiv:2408.16262               | 1        | 2024 |
| <a href="#">Loss of plasticity in deep continual learning</a><br>S Dohare, JF Hernandez-Garcia, Q Lan, P Rahman, AR Mahmood, ...<br>Nature 632 (8026), 768-774                           | 60       | 2024 |
| <a href="#">An Idiosyncrasy of Time-discretization in Reinforcement Learning</a><br>K De Asis, RS Sutton<br>arXiv preprint arXiv:2406.14951  | 1        | 2024 |
| <a href="#">Off-Policy Prediction Learning: An Empirical Study of Online Algorithms</a><br>S Ghiassian, B Rafiee, RS Sutton<br>IEEE Transactions on Neural Networks and Learning Systems | 1        | 2024 |
| <a href="#">Reward centering</a><br>A Naik, Y Wan, M Tomar, RS Sutton<br>arXiv preprint arXiv:2405.09999   | 9        | 2024 |

---

## Online Gradient Descent (OGD)

---

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

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

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

---

## Online Gradient Descent (OGD)

---

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

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

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

---

The intuition of OGD is to solve “trust region optimization”:

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

---

## Online Gradient Descent (OGD)

---

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

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

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

The intuition of OGD is to minimize the first order approximation + regularization with  $\ell_2$  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 equivalent 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  $\psi : X \rightarrow R$  be strongly convex and continuously differentiable function. The Bregman divergence w.r.t.  $\psi$  is  $B_\psi$  is defined as

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

If  $\psi$  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  $\alpha$ -strongly convex, we have a global property

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



# 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

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

# 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  $\psi$ , and we have

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

# 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  $\psi$ , 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  $t = 1, \dots, T$ :

- **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

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

$$x_{t+1} = \arg \min_{x \in \mathcal{K}} B_\psi(x; y_{t+1})$$

# 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  $\psi$  be  $\alpha$ -strongly convex function. Consider a fixed learning rate  $\eta_t = \eta$ . Online mirrored descent algorithm achieves*

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

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

# 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  $\psi$  be  $\alpha$ -strongly convex function. Consider a fixed learning rate  $\eta_t = \eta$ . Online mirrored descent algorithm achieves*

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

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

- The feasible set and gradients are bounded.

# 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  $\psi$  be  $\alpha$ -strongly convex function. Consider a fixed learning rate  $\eta_t = \eta$ . Online mirrored descent algorithm achieves*

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

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

- The feasible set and gradients are bounded.
- Learning rate is fixed with  $O(1/\sqrt{T})$ .



# 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  $\psi$  be  $\alpha$ -strongly convex function. Consider a fixed learning rate  $\eta_t = \eta$ . Online mirrored descent algorithm achieves*

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

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<sup>1</sup>

## 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*

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

---

<sup>1</sup>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

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

## Theorem 5 – Proof

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

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

we have

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

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  $\theta_{t+1}$  back to  $\nabla\psi^*(\theta_{t+1})$ .

# Review of Expert problem

---

## Expert problem:

---

**Initialization:**  $N$  experts/models.

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

- **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  $t = 1, \dots, T$ :

- **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:**  $x_1 = [1/d, \dots, 1/d]$  and  $\eta$ .

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

- **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:

---

**Initialization:**  $x_1 = [1/d, \dots, 1/d]$  and  $\eta$ .

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

- **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:**  $x_1 = [1/d, \dots, 1/d]$  and  $\eta$ .

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

- **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?

# Exponentiated Gradient – Online Mirrored Descent

---

## Online Mirrored Descent:

---

**Initialization:**  $x_1 = [1/d, \dots, 1/d]$  and  $\eta$ .

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

- **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:

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

# Exponentiated Gradient as Online Mirrored Descent

The update of Bragman divergence

$$\begin{aligned} \min_{x \in \mathcal{K}} \quad & \eta \langle x, \nabla f_t(x_t) \rangle + \sum_{i=1}^d x_i \log \frac{x_i}{x_{t,i}} \\ \text{s.t.} \quad & \sum_{i=1}^d x_i = 1, \quad x_i \geq 0. \end{aligned}$$

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:**  $x_1 = [1/d, \dots, 1/d]$  and  $\eta_t$ .

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

- **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!

## Theorem 6 (Restate Theorem 2)

Let  $\psi$  be  $\alpha$ -strongly convex function in  $B_\psi$ . Let fixed learning rate  $\eta_t = \eta$ . Online mirrored descent algorithm achieves

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

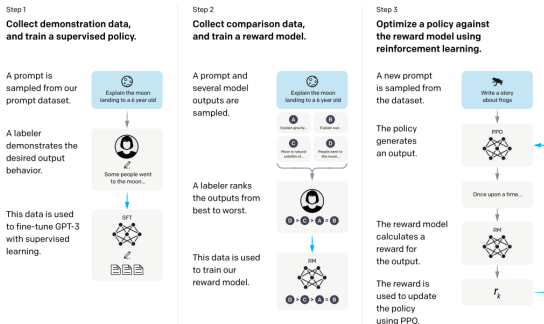
In Hedge, we have

- $\psi(x) = \sum_{i=1}^d x_i \log x_i$  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

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

# LLM Finetuning



“Reinforcement learning from human feedback” for LLM finetuning<sup>2</sup>:

- Learn a base/ref policy from supervise learning
- Learn reward models via human/AI feedback
- Policy optimization based on reward models

<sup>2</sup>Long Ouyang, etc., “Training language models to follow instructions with human feedback”. NeurIPS, 2022.

# RLHF Finetuning as (Online) Mirrored Descent

---

## RLHF<sup>3</sup> as (Online) Mirrored Descent:

---

**Initialization:** A pretrained reward model  $r$  and policy  $\pi_{ref}$ .

**FineTuning:** Find a policy such that

$$\pi_{\theta}^* = \arg \max \mathbb{E}_{c \sim \mathcal{D}, a \sim \pi_{\theta}(a|c)} [r(c, a)] - \frac{1}{\eta} \text{KL}(\pi_{\theta}; \pi_{ref}).$$

---

---

<sup>3</sup>OpenAI. Training language models to follow instructions with human feedback. NeurIPS 2022.

# 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  $t = 1, \dots, T$ :

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

# 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  $t = 1, \dots, T$ :

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



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.



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.

# 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  $t = 1, \dots, T$ :

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

 **You**  
predict weather tomorrow in shanghai

 **ChatGPT**  
I'm sorry for any confusion, but as an AI, I don't have access to real-time data or the ability to predict the weather. For accurate and up-to-date weather forecasts for Shanghai or any other location, I recommend checking a reliable weather forecasting website or app, or tuning in to a local weather report. These sources typically provide the most accurate and timely information about weather conditions.

 **You**  
list three top papers on this topic with just titles

 **ChatGPT**

1. "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems" by Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer.
2. "Follow the Regularized Leader and Mirror Descent: Equivalence Theorems and L1 Regularization" by Elad Hazan, Amit Agarwal, and Satyen Kale.
3. "Online Learning: A Survey" by Shai Shalev-Shwartz.

# 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  $t = 1, \dots, T$ :

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

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

---

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

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

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

$\text{Alg}(x_1, \dots, x_t, \ell_1, \dots, \ell_t, \hat{\ell}_{t+1})$  could be  $\text{Alg}(x_t, \ell_t, \hat{\ell}_{t+1})$  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  $t = 1, \dots, T$ :

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

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 Learning with Prediction

---

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

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

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

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  $t = 1, \dots, T$  :

- **Learner:** Submit  $x_t$ .
- **Environment:** Observe the loss  $\ell_t$ .
- **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<sup>4</sup> is as follows.

## Theorem 7

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

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

“Almost” the best of two worlds:

---

<sup>4</sup>Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. COLT, 2013

# Online Mirrored Descent with Prediction – Regret

The regret of OMD with prediction<sup>4</sup> is as follows.

## Theorem 7

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

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

“Almost” the best of two worlds:

- If the predictions are “perfect”, the regret is constant!

---

<sup>4</sup>Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. COLT, 2013

# Online Mirrored Descent with Prediction – Regret

The regret of OMD with prediction<sup>4</sup> is as follows.

## Theorem 7

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

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

“Almost” the best of two worlds:

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

---

<sup>4</sup>Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. COLT, 2013

# Online Mirrored Descent with Prediction – Regret

The regret of OMD with prediction<sup>4</sup> is as follows.

## Theorem 7

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

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

“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  $o(\sqrt{T})$ .

---

<sup>4</sup>Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. COLT, 2013



# Review: OMD without 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

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

# OMD 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

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

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 **for any  $x$**

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

# OMD with Prediction – Proof

# Why Online Gradient/Mirrored Descent?

---

## Online Learning Algorithm

---

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

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

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

We design online learning algorithms to achieve small regret:

- Online gradient/mirrored descent is based on the current  $x_t$  and  $\ell_t$  as

$$\text{Alg}(x_t, \ell_t).$$

# Why Online Gradient/Mirrored Descent?

---

## Online Learning Algorithm

---

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

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

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

We design online learning algorithms to achieve small regret:

- Online gradient/mirrored descent is based on the current  $x_t$  and  $\ell_t$  as

$$\text{Alg}(x_t, \ell_t).$$

- Can we use all information to design online algorithms?

$$x_{t+1} = \text{Alg}(x_1, \dots, x_t, \ell_1, \dots, \ell_t).$$

# Follow-The-Leader (FTL) Algorithm

---

## Follow-The-Leader (FTL) Algorithm

---

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

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

- **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

---

## Follow-The-Leader (FTL) Algorithm

---

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

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

- **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”:

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



# Follow-The-Leader (FTL) Algorithm – Proof

---

## Follow-The-Leader (FTL) Algorithm

---

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

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

- **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

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

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  $t = 1, \dots, T$ :

- **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 regularizer:

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

## Theorem 9 (linear losses and quadratic regularizer)

*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 regularizer.

# 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 regularizer:

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



# FTRL and OMD Algorithms

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

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

# FTRL and OMD Algorithms

FTRL with the linear losses and adaptive regularization are

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

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

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

Recall OMD updates as

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

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  $\eta_t$ .

# Follow-The-Regularized-Leader Algorithm

---

## Follow-The-Regularized-Leader (FTRL) Algorithm

---

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

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

- **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 regularizer 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 regularizer  $R_t(x)$  that is “increasing” as time  $t$  and  $\alpha_t$ -strongly convex.

## Theorem 10 (convex losses and adaptive regularizer)

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

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

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

# 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

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

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  $\alpha_t$ -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 Follow-The-Regularized-Leader (FTRL)

---

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

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

- **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.

## 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  $\alpha_t$ -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

---

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

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

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

A few examples:

- Subseasonal prediction: the prediction correct or not will be known in 2~6 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  $t = 1, \dots, T$ :

- **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 “optimism” !!!

---

## Delay as Optimism in FTRL

---

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

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

- **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!

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