Optimistic Learning for Resource Management in Communication Systems

Naram Mhaisen, George Iosifidis

ELLIS, Delft, 2024



Table of Contents

1. Background & Motivation
2. Preliminaries & Convex OL
3. Non-convex OL
4. OL & Non-stochastic Control

Table of Contents

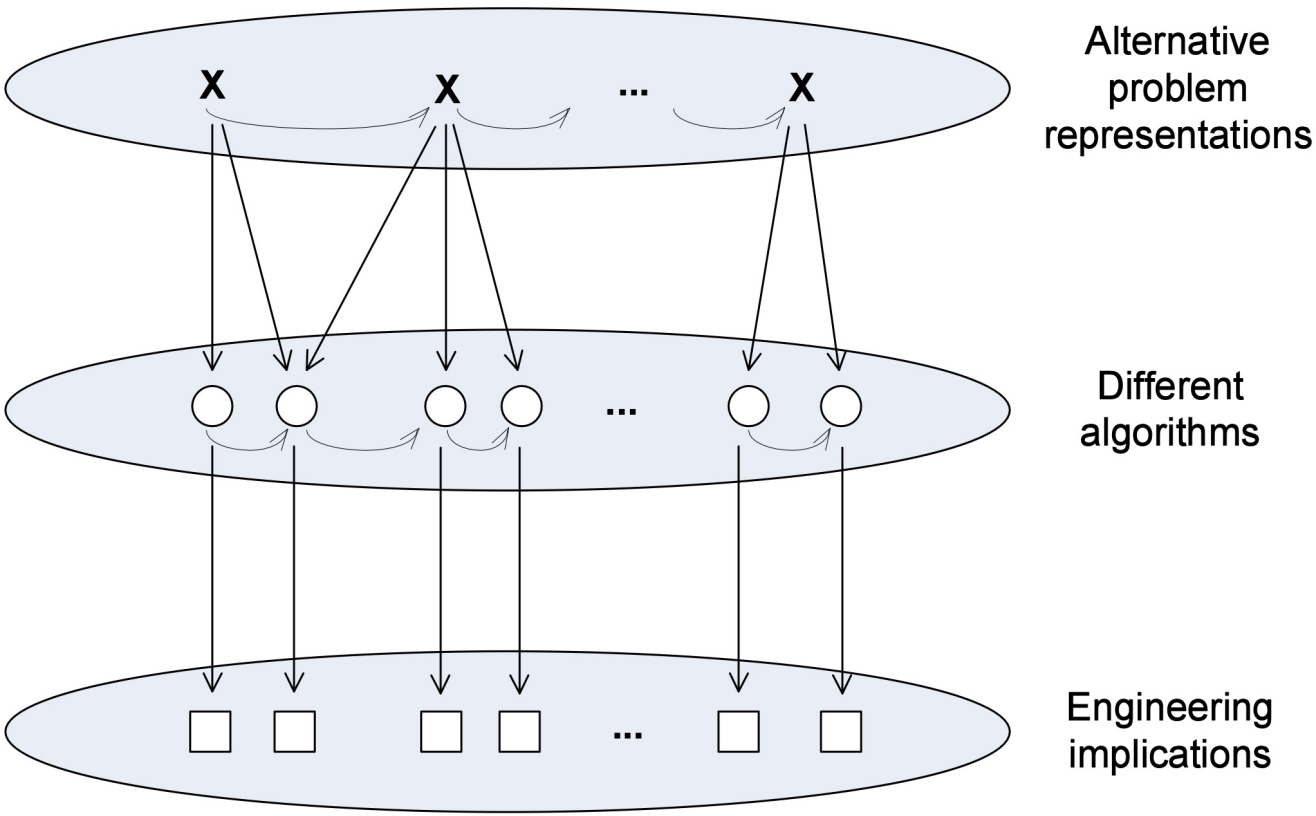
1 Background & Motivation

2 Preliminaries & Convex OL

3 Non-convex OL

4 OL & Non-stochastic Control

Network Control



The Optimization Perspective (credits: M. Chiang et al.)

Facets of Network Control

Resource Management via **Static** Optimization

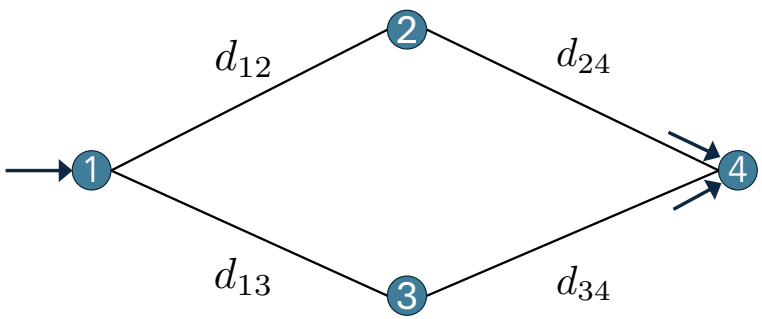
- The network controller (NC) has prior access to user demands (e.g., popularity of files) and network parameters (e.g., link delays).
- Formulates & solves an offline optimization problem.
- Applies the solution for the rest of the network operation (one-off).

M. Chiang et al, A Mathematical Theory of Network Architectures, Proceedings of IEEE, 2007.

Facets of Network Control

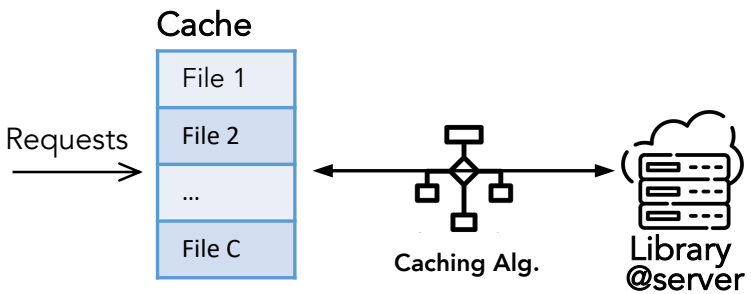
Resource Management via **Static** Optimization

- The network controller (NC) has prior access to user demands (e.g., popularity of files) and network parameters (e.g., link delays).
- Formulates & solves an offline optimization problem.
- Applies the solution for the rest of the network operation (one-off).



Selection of Routing Path

Which path has the smallest delay?



Selection of Cached Files

Which file receives most requests?

M. Chiang et al, A Mathematical Theory of Network Architectures, Proceedings of IEEE, 2007.

Facets of Network Control

Resource Management via **Stochastic** Optimization

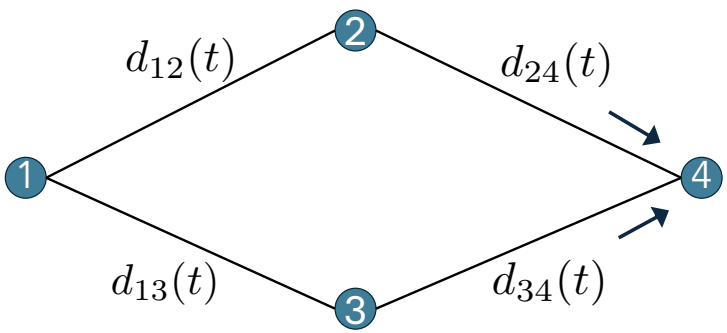
- The system operation is time-slotted.
- NC observes at the beginning of each slot the demands, environment and network state (e.g., unfinished workloads = queues).
- Solves a problem to find the best decision for the current slot.

M. Neely, Stochastic Network Optimization with Applications to Communication & Queueing Systems, Synthesis Lectures on Communication Networks, 2012.

Facets of Network Control

Resource Management via **Stochastic** Optimization

- The system operation is time-slotted.
- NC observes at the beginning of each slot the demands, environment and network state (e.g., unfinished workloads = queues).
- Solves a problem to find the best decision for the current slot.



$\forall t$: observe delays & queues; select max-weight path

- **Max-weight** policy provably converges to benchmark.
- **Benchmark**: the best path w.r.t. the delays over all slots, $t = 1, \ldots, T$.
- Perturbations are benign (i.i.d.); performance functions are known.

M. Neely, Stochastic Network Optimization with Applications to Communication & Queueing Systems, Synthesis Lectures on Communication Networks, 2012.

Facets of Network Control

Solve (one-off or over time) problems with i.i.d.-perturbed parameters and fixed/known performance functions.

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) - g(\boldsymbol{y}) \\ \text{Subject to} \quad & \boldsymbol{q}(\boldsymbol{x}) \leq \boldsymbol{Q} \\ & \boldsymbol{h}(\boldsymbol{y}) \leq \boldsymbol{H} \\ & \textit{Network Stability} \end{aligned}$$

Facets of Network Control

Solve (one-off or over time) problems with i.i.d.-perturbed parameters and fixed/known performance functions.

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) - g(\boldsymbol{y}) \\ \text{Subject to} \quad & \boldsymbol{q}(\boldsymbol{x}) \leq \boldsymbol{Q} \\ & \boldsymbol{h}(\boldsymbol{y}) \leq \boldsymbol{H} \\ & \textit{Network Stability} \end{aligned}$$

What if:

- The parameters (e.g., link delays) change arbitrarily?

Facets of Network Control

Solve (one-off or over time) problems with i.i.d.-perturbed parameters and fixed/known performance functions.

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) - g(\boldsymbol{y}) \\ \text{Subject to} \quad & \boldsymbol{q}(\boldsymbol{x}) \leq \boldsymbol{Q} \\ & \boldsymbol{h}(\boldsymbol{y}) \leq \boldsymbol{H} \\ & \textit{Network Stability} \end{aligned}$$

What if:

- The parameters (e.g., link delays) change arbitrarily?
- The performance functions are time-varying?

Facets of Network Control

Solve (one-off or over time) problems with i.i.d.-perturbed parameters and fixed/known performance functions.

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) - g(\boldsymbol{y}) \\ \text{Subject to} \quad & \boldsymbol{q}(\boldsymbol{x}) \leq \boldsymbol{Q} \\ & \boldsymbol{h}(\boldsymbol{y}) \leq \boldsymbol{H} \\ & \textit{Network Stability} \end{aligned}$$

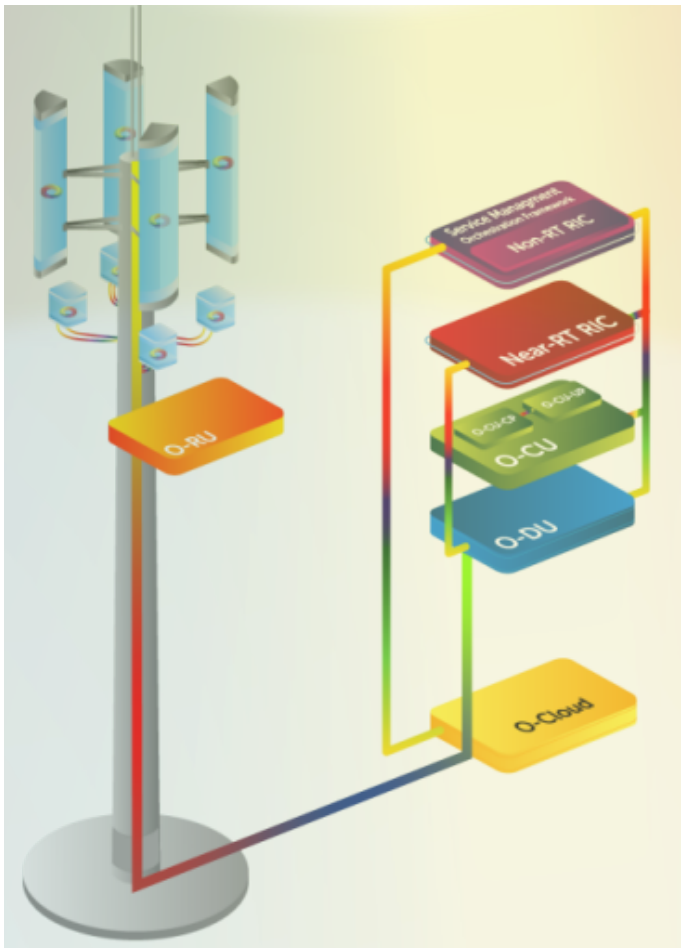
What if:

- The parameters (e.g., link delays) change arbitrarily?
- The performance functions are time-varying?
- Parameters/functions are revealed to NC only at the end of each slot?

Virtualization & Softwarization



GSM Base Stations (2G), 1990s

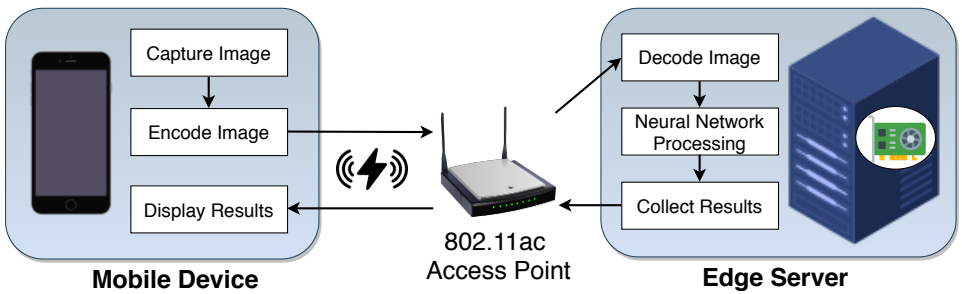


5G Virtualized BSs

New Type of Services



Seeing AI: Talking Camera for the Blind

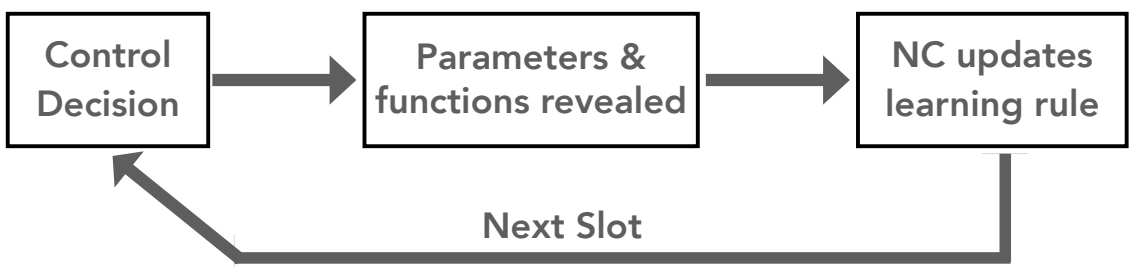


End-to-End Control of Mobile Analytics

Online Convex Optimization Approach (1/2)

The system operation is modeled as an online learning problem. The NC:

1. keeps track of past decisions, perturbations and performance functions;
2. commits its decision at the start of each slot;
3. observes the outcome (function, perturbations) at the end of the slot;
4. and updates its decisions.

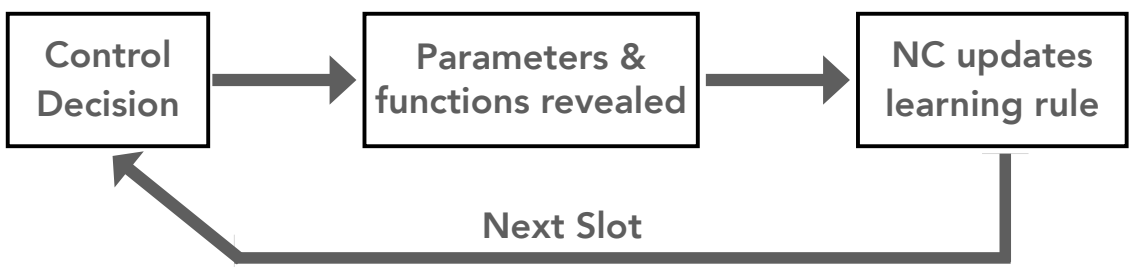


M. Zinkevich, Online Convex Programming and Generalized Infinitesimal Gradient Ascent, ICML 2003.
E. Hazan, Introduction to Online Convex Optimization. Found. Trends Optim. 2(3-4), 2016.

Online Convex Optimization Approach (1/2)

The system operation is modeled as an online learning problem. The NC:

1. keeps track of past decisions, perturbations and performance functions;
2. commits its decision at the start of each slot;
3. observes the outcome (function, perturbations) at the end of the slot;
4. and updates its decisions.



M. Zinkevich, Online Convex Programming and Generalized Infinitesimal Gradient Ascent, ICML 2003.
E. Hazan, Introduction to Online Convex Optimization. Found. Trends Optim. 2(3-4), 2016.

Online Convex Optimization Approach (1/2)

The system operation is modeled as an online learning problem. The NC:

1. keeps track of past decisions, perturbations and performance functions;
2. commits its decision at the start of each slot;
3. observes the outcome (function, perturbations) at the end of the slot;
4. and updates its decisions.

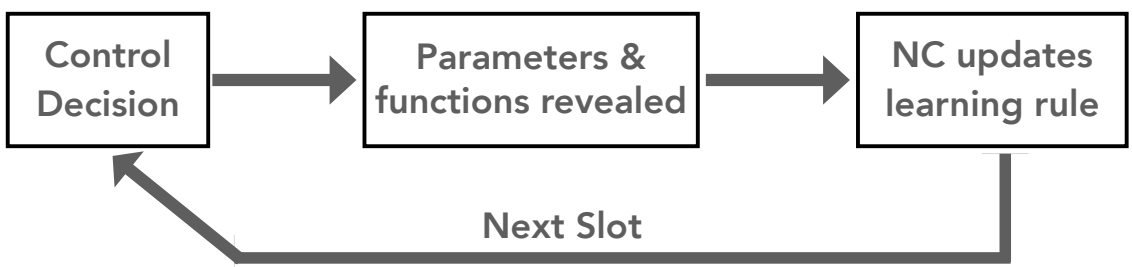


M. Zinkevich, Online Convex Programming and Generalized Infinitesimal Gradient Ascent, ICML 2003.
E. Hazan, Introduction to Online Convex Optimization. Found. Trends Optim. 2(3-4), 2016.

Online Convex Optimization Approach (1/2)

The system operation is modeled as an online learning problem. The NC:

1. keeps track of past decisions, perturbations and performance functions;
2. commits its decision at the start of each slot;
3. observes the outcome (function, perturbations) at the end of the slot;
4. and updates its decisions.



M. Zinkevich, Online Convex Programming and Generalized Infinitesimal Gradient Ascent, ICML 2003.
E. Hazan, Introduction to Online Convex Optimization. Found. Trends Optim. 2(3-4), 2016.

Online Convex Optimization Approach (2/2)

- Fewer assumptions & richer modeling tool
  - Perturbations (e.g., delays) can change even adversarially.
  - No need to know the performance functions in advance.
  - Guaranteed convergence to an oracle benchmark.
- Robustness through cautiousness: decisions tailored to worst scenario.
- In practice, sometimes:
  - Perturbations may be stationary.
  - Performance functions may be predictable.
  - There is often abundance of training data.

T. Chen, et al., Learning and Management for Internet of Things: Accounting for Adaptivity and Scalability. Proc. IEEE, 2019.

Online Convex Optimization Approach (2/2)

- Fewer assumptions & richer modeling tool
  - Perturbations (e.g., delays) can change even adversarially.
  - No need to know the performance functions in advance.
  - Guaranteed convergence to an oracle benchmark.
- Robustness through cautiousness: decisions tailored to worst scenario.
- In practice, sometimes:
  - Perturbations may be stationary.
  - Performance functions may be predictable.
  - There is often abundance of training data.

T. Chen, et al., Learning and Management for Internet of Things: Accounting for Adaptivity and Scalability. Proc. IEEE, 2019.

Online Convex Optimization Approach (2/2)

- Fewer assumptions & richer modeling tool
  - Perturbations (e.g., delays) can change even adversarially.
  - No need to know the performance functions in advance.
  - Guaranteed convergence to an oracle benchmark.
- Robustness through cautiousness: decisions tailored to worst scenario.
- In practice, sometimes:
  - Perturbations may be stationary.
  - Performance functions may be predictable.
  - There is often abundance of training data.

T. Chen, et al., Learning and Management for Internet of Things: Accounting for Adaptivity and Scalability. Proc. IEEE, 2019.

Offline or Online ?

Offline Learning

+ Abundance of training data
  - path delays, file requests, etc.
+ NNs predict perturb. or actions
  - Fast & efficient solution
– Inaccurate under distribution shifts
  - Need data & offline training

C. Zhang, P. Patras, H. Haddadi: Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Commun. Surv. Tutorials 21(3), 2019.

Offline or Online ?

Offline Learning

+ Abundance of training data
  - path delays, file requests, etc.
+ NNs predict perturb. or actions
  - Fast & efficient solution
– Inaccurate under distribution shifts
  - Need data & offline training

Online Learning

+ No prior training
  - Adapts at runtime
+ Works with general settings
  - Minimal assumptions
  - Guaranteed & interpretable performance
– Unnecessary slow when:
  - perturbations are benign
  - functions are observable

C. Zhang, P. Patras, H. Haddadi: Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Commun. Surv. Tutorials 21(3), 2019.

Optimistic Learning

How to leverage offline trained predictors in online learning?

Optimistic Learning

How to leverage offline trained predictors in online learning?

Goal:

- Achieve faster convergence when predictions are proved accurate;
- Achieve the performance of legacy OCO when predictions are inaccurate.

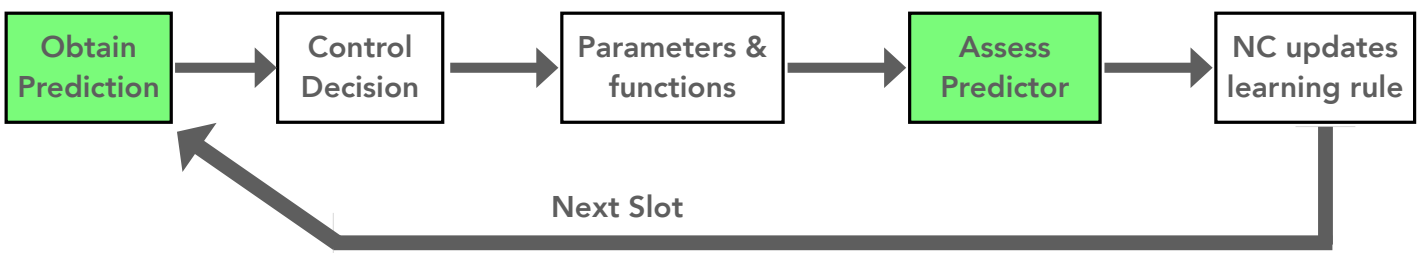


Table of Contents

1 Background & Motivation

2 Preliminaries & Convex OL

3 Non-convex OL

4 OL & Non-stochastic Control

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

$$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum\nolimits_n x_n \leq C \right\}$$

2. Request is submitted

$$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

$$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum_n x_n \leq C \right\}$$

2. Request is submitted

$$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

$$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum\nolimits_n x_n \leq C \right\}$$

2. Request is submitted

$$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

   $$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum_n x_n \leq C \right\}$$

2. Request is submitted

   $$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

   $$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

   $$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum\nolimits_n x_n \leq C \right\}$$

2. Request is submitted

   $$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

   $$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

   $$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum\nolimits_n x_n \leq C \right\}$$

2. Request is submitted

   $$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

   $$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

   $$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum_n x_n \leq C \right\}$$

2. Request is submitted

   $$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

   $$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO Template

For $t = 1, 2, \ldots, T$ do:

1. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
2. Nature reveals utility $f_t(\cdot)$
3. Learner gains $f_t(\boldsymbol{x}_t)$

Target: The best in hindsight

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

E.g.: Online Caching (Cache C out of N files)

1. Decide the cached files

   $$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \mid \sum_n x_n \leq C \right\}$$

2. Request is submitted

   $$\boldsymbol{q}_t \in \{0,1\}^N$$

3. Cache hit or miss

   $$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

Target: Top C requested files

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle$$

G. S. Paschos, A. Destounis, G. Iosifidis: Online Convex Optimization for Caching Networks. IEEE/ACM Trans. Netw. 28(2): 625-638 (2020).

OCO with Optimism

For $t = 1, 2, \ldots, T$ do:

1. Obtain $\tilde{f}_t(\cdot)$.
2. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
1. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

$$\epsilon_t = \|\nabla f_t(\boldsymbol{x}_t) - \nabla \tilde{f}_t(\boldsymbol{x}_t)\|^2$$

Online Caching (Cache C out of N files)

1. Obtain request prediction $\tilde{\boldsymbol{q}}_t$
2. Decide the cached files $\boldsymbol{x}_t$
3. Request $\boldsymbol{q}_t$ is submitted
4. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

5. Prediction error

$$\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$$

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching with Optimistic Learning. IFIP Networking 2022.

OCO with Optimism

For $t = 1, 2, \ldots, T$ do:

1. Obtain $\tilde{f}_t(\cdot)$.
2. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
1. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

$$\epsilon_t = \|\nabla f_t(\boldsymbol{x}_t) - \nabla \tilde{f}_t(\boldsymbol{x}_t)\|^2$$

Online Caching (Cache C out of N files)

1. Obtain request prediction $\tilde{\boldsymbol{q}}_t$
2. Decide the cached files $\boldsymbol{x}_t$
3. Request $\boldsymbol{q}_t$ is submitted
4. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

5. Prediction error

$$\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$$

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching with Optimistic Learning. IFIP Networking 2022.

OCO with Optimism

For $t = 1, 2, \ldots, T$ do:

1. Obtain $\tilde{f}_t(\cdot)$.
2. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
1. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

$$\epsilon_t = \|\nabla f_t(\boldsymbol{x}_t) - \nabla \tilde{f}_t(\boldsymbol{x}_t)\|^2$$

Online Caching (Cache C out of N files)

1. Obtain request prediction $\tilde{\boldsymbol{q}}_t$
2. Decide the cached files $\boldsymbol{x}_t$
3. Request $\boldsymbol{q}_t$ is submitted
4. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

5. Prediction error

$$\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$$

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching with Optimistic Learning. IFIP Networking 2022.

OCO with Optimism

For $t = 1, 2, \ldots, T$ do:

1. Obtain $\tilde{f}_t(\cdot)$.
2. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
1. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

$$\epsilon_t = \|\nabla f_t(\boldsymbol{x}_t) - \nabla \tilde{f}_t(\boldsymbol{x}_t)\|^2$$

Online Caching (Cache C out of N files)

1. Obtain request prediction $\tilde{\boldsymbol{q}}_t$
2. Decide the cached files $\boldsymbol{x}_t$
3. Request $\boldsymbol{q}_t$ is submitted
4. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

5. Prediction error

$$\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$$

↪ How can we predict file requests in caching problems?

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching with Optimistic Learning. IFIP Networking 2022.

OCO with Optimism

For $t = 1, 2, \ldots, T$ do:

1. Obtain $\tilde{f}_t(\cdot)$.
2. Learner decides $\boldsymbol{x}_t \in \mathcal{X}$
1. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

$$\epsilon_t = \|\nabla f_t(\boldsymbol{x}_t) - \nabla \tilde{f}_t(\boldsymbol{x}_t)\|^2$$

Online Caching (Cache C out of N files)

1. Obtain request prediction $\tilde{\boldsymbol{q}}_t$
2. Decide the cached files $\boldsymbol{x}_t$
3. Request $\boldsymbol{q}_t$ is submitted
4. Cache hit or miss

$$f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$$

5. Prediction error

$$\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$$

↪ How can we predict file requests in caching problems?
↪ Why do we measure the prediction error this way?

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching with Optimistic Learning. IFIP Networking 2022.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t)\Big)$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $\boldsymbol{x}_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t) \Big)$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $\boldsymbol{x}_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sup_{\{f_t\}_T} \left[\sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t)\Big)\right]$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $\boldsymbol{x}_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sup_{\{f_t\}_T} \left[\sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t)\Big)\right]$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $\boldsymbol{x}_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sup_{\{f_t\}_T} \left[\sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t)\Big)\right]$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $\boldsymbol{x}_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sup_{\{f_t\}_T} \left[\sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t)\Big)\right]$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $x_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (1/3)

- An online algorithm $\mathcal{A}$ produces $\{\boldsymbol{x}_t\}_T \doteq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$

Definition (Regret Metric)

$$\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \sup_{\{f_t\}_T} \left[\sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t)\Big)\right]$$

- In caching: $f_t(\boldsymbol{x_t}) = \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$; $\mathcal{R}_T\big(\{\boldsymbol{x}_t\}_T\big) = \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{q_t}, \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle}_{\text{Cache-hit gap}}$.
- $\mathcal{A}$ is no-regret when $\lim_{T\to\infty} \mathcal{R}_T/T = 0$.
- I.e., it performs as good as $\boldsymbol{x}^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$
- E.g., $\boldsymbol{x}_{t+1} = \arg\max \langle q_{1:t}, x \rangle$, *or* $\arg\max \langle \tilde{q}_{t+1}, x \rangle$, *or* $\arg\max \langle q_{1:t} + \tilde{q}_{t+1}, x \rangle$ *or* $\arg\max \langle \alpha q_{1:t} + (1-\alpha)\tilde{q}_{t+1}, x \rangle$ all lead to $\mathcal{O}(T)$ (linear) regret.

Metric, Algorithms & Basic Results (2/3)

Follow **T**he **R**egularized **L**eader (FTRL):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$$

$r_t(\boldsymbol{x}) = \sigma_t \|\boldsymbol{x}\|^2$ (OGD)
or $\sigma_t \sum_i x_i \log x_i$ (EG)

- $\sqrt{T}$ (Optimal) regret *for any* $\{\boldsymbol{q}_t\}_T$
- Works well for alternating functions.

Be **T**he **L**eader (BTL):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_t, \boldsymbol{x} \rangle \right\}$$

- 0-regret when $\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_{t+1}$
- Works well for fixed / slowly varying functions (by setting $\tilde{\boldsymbol{q}}_{t+1} = \boldsymbol{q}_t$)

Optimistic FTRL: Regs. depend on prediction errors, e.g.,$r_t(\boldsymbol{x}) = \sigma(\epsilon_{1:t})\|\boldsymbol{x}\|^2$

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$$

B. McMahan: A Survey of Algorithms and Analysis for Adaptive Online Learning. J. Mach. Learn., 2017; N. Mhaisen, G. Iosifidis, et. al., : Online Caching With no Regret. IEEE Trans. Mob. Comput, 2024.

Metric, Algorithms & Basic Results (2/3)

Follow **T**he **R**egularized **L**eader (FTRL):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$$

$r_t(\boldsymbol{x}) = \sigma_t \|\boldsymbol{x}\|^2$ (OGD)
or $\sigma_t \sum_i x_i \log x_i$ (EG)

- $\sqrt{T}$ (Optimal) regret *for any* $\{\boldsymbol{q}_t\}_T$
- Works well for alternating functions.

Be **T**he **L**eader (BTL):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_t, \boldsymbol{x} \rangle \right\}$$

- 0-regret when $\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_{t+1}$
- Works well for fixed / slowly varying functions (by setting $\tilde{\boldsymbol{q}}_{t+1} = \boldsymbol{q}_t$)

Optimistic FTRL: Regs. depend on prediction errors, e.g., $r_t(\boldsymbol{x}) = \sigma(\epsilon_{1:t})\|\boldsymbol{x}\|^2$

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$$

B. McMahan: A Survey of Algorithms and Analysis for Adaptive Online Learning. J. Mach. Learn., 2017; N. Mhaisen, G. Iosifidis, et. al., : Online Caching With no Regret. IEEE Trans. Mob. Comput, 2024.

Metric, Algorithms & Basic Results (2/3)

Follow **T**he **R**egularized **L**eader (FTRL):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$$

$r_t(\boldsymbol{x}) = \sigma_t \|\boldsymbol{x}\|^2$ (OGD)
or $\sigma_t \sum_i x_i \log x_i$ (EG)

- $\sqrt{T}$ (Optimal) regret *for any* $\{\boldsymbol{q}_t\}_T$
- Works well for alternating functions.

Be **T**he **L**eader (BTL):

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_t, \boldsymbol{x} \rangle \right\}$$

- 0-regret when $\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_{t+1}$
- Works well for fixed / slowly varying functions (by setting $\tilde{\boldsymbol{q}}_{t+1} = \boldsymbol{q}_t$)

Optimistic FTRL: Regs. depend on prediction errors, e.g.,$r_t(\boldsymbol{x}) = \sigma(\epsilon_{1:t})\|\boldsymbol{x}\|^2$

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$$

BTL's $\boldsymbol{x}_{t+1}$ When $\epsilon_t = 0, \forall t$ ⟷ $\approx$ FTRL's $\boldsymbol{x}_{t+1}$ When $\epsilon_t > 0$

B. McMahan: A Survey of Algorithms and Analysis for Adaptive Online Learning. J. Mach. Learn., 2017; N. Mhaisen, G. Iosifidis, et. al., : Online Caching With no Regret. IEEE Trans. Mob. Comput, 2024.

Metric, Algorithms & Basic Results (3/3)

Regret Bound

Using $r_t(\cdot)$ that are σ_t-strongly convex w.r.t. $\|\cdot\|$, where σ_t is non-decreasing, the FTRL iteration $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$ produces decisions (e.g., caching vectors) $\{\boldsymbol{x}_t\}_T$ such that:

$$\mathcal{R}_T \;\leq\; \sigma_{T-1}\|\boldsymbol{x}^\star\| + \sum_{t=1}^{T} \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|}{\sigma_{t-1}} \;\leq\; \sigma_{T+1}$$

M. Mohri, S. Yang: Accelerating Online Convex Optimization via Adaptive Prediction. AISTATS 2016.

Metric, Algorithms & Basic Results (3/3)

Regret Bound

Using $r_t(\cdot)$ that are σ_t-strongly convex w.r.t. $\|\cdot\|$, where σ_t is non-decreasing, the FTRL iteration $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x}) \right\}$ produces decisions (e.g., caching vectors) $\{\boldsymbol{x}_t\}_T$ such that:

$$\mathcal{R}_T \;\leq\; \sigma_{T-1}\|\boldsymbol{x}^\star\| + \sum_{t=1}^{T} \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|}{\sigma_{t-1}} \;\leq\; \sigma_{T+1}$$

Optimistic Regs ($\forall \tilde{\boldsymbol{q}}_t$)

$$\sigma_t \approx \sqrt{\textstyle\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|^2}$$

$$\mathcal{R}_T \leq G\sqrt{\textstyle\sum_{t=1}^{T} \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2}$$

M. Mohri, S. Yang: Accelerating Online Convex Optimization via Adaptive Prediction. AISTATS 2016.

OL for Caching

Algorithm: Optimistic (Bipartite) Caching

Input: $x_1 \in \mathcal{X}$; $\sigma = 2/D_{\mathcal{X}}$. **Output**: $\boldsymbol{x}_t = (\boldsymbol{y}_t, \boldsymbol{z}_t)$, $\forall t$.
for $t = 1, 2, \ldots$ **do**
  Receive prediction $\tilde{\boldsymbol{q}}_{t+1}$
  Calculate the new policy $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + \sigma_t r(\boldsymbol{x})$
  Serve request $\boldsymbol{q}_t$ according to $\boldsymbol{x}_t$
  Incur a hit/miss $\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$ & measure prediction error
  Update the regularization $\sigma_{t+1} \approx \sqrt{\sum_{i=1}^{t} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|^2}$
end

- $\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2}\right)$, worse by $\sqrt{2}$ w.r.t. non-optimistic FTRL.

$$\mathcal{R}_t \leq$$

0 When $\tilde{\boldsymbol{q}}_t = \boldsymbol{q}_t, \forall t$ — $\sqrt{\#\text{ mistakes}}$ — $4C\sqrt{T}$ When $\tilde{\boldsymbol{q}}_t \neq \boldsymbol{q}_t, \forall t$

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching With no Regret: Optimistic Learning via Recommendations. IEEE Trans. Mob. Comput. 23(5), 2024.

OL for Caching

Algorithm: Optimistic (Bipartite) Caching

Input: $x_1 \in \mathcal{X}$; $\sigma = 2/D_{\mathcal{X}}$. **Output**: $\boldsymbol{x}_t = (\boldsymbol{y}_t, \boldsymbol{z}_t)$, $\forall t$.
for $t = 1, 2, \ldots$ **do**
  Receive prediction $\tilde{\boldsymbol{q}}_{t+1}$
  Calculate the new policy $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x} \rangle + \sigma_t r(\boldsymbol{x})$
  Serve request $\boldsymbol{q}_t$ according to $\boldsymbol{x}_t$
  Incur a hit/miss $\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$ & measure prediction error
  Update the regularization $\sigma_{t+1} \approx \sqrt{\sum_{i=1}^{t} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|^2}$
end

- $\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2}\right)$, worse by $\sqrt{2}$ w.r.t. non-optimistic FTRL.

$$\mathcal{R}_t \leq$$

0 When $\tilde{\boldsymbol{q}}_t = \boldsymbol{q}_t, \forall t$ — $\sqrt{\#\text{ mistakes}}$ — $4C\sqrt{T}$ When $\tilde{\boldsymbol{q}}_t \neq \boldsymbol{q}_t, \forall t$

N. Mhaisen, G. Iosifidis, D. J. Leith: Online Caching With no Regret: Optimistic Learning via Recommendations. IEEE Trans. Mob. Comput. 23(5), 2024.

Implementation

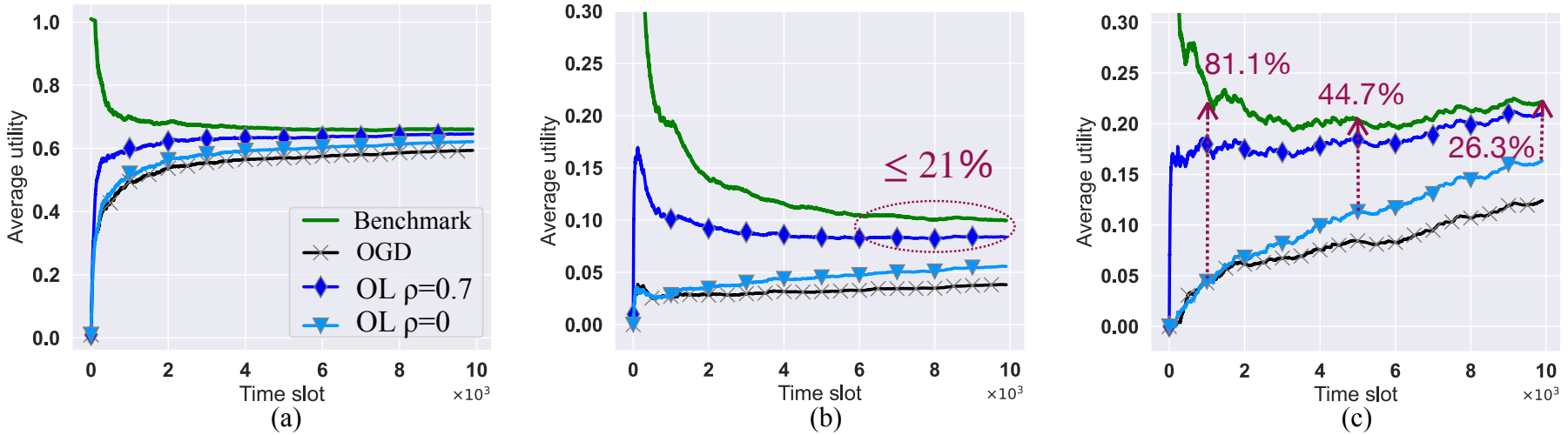


Figure: Cache utility with predictors of different accuracy ($\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_t$ with probability ρ) and request traces from: (a) Zipf distribution; (b) YouTube dataset; (c) MovieLens datasets.

Implementation

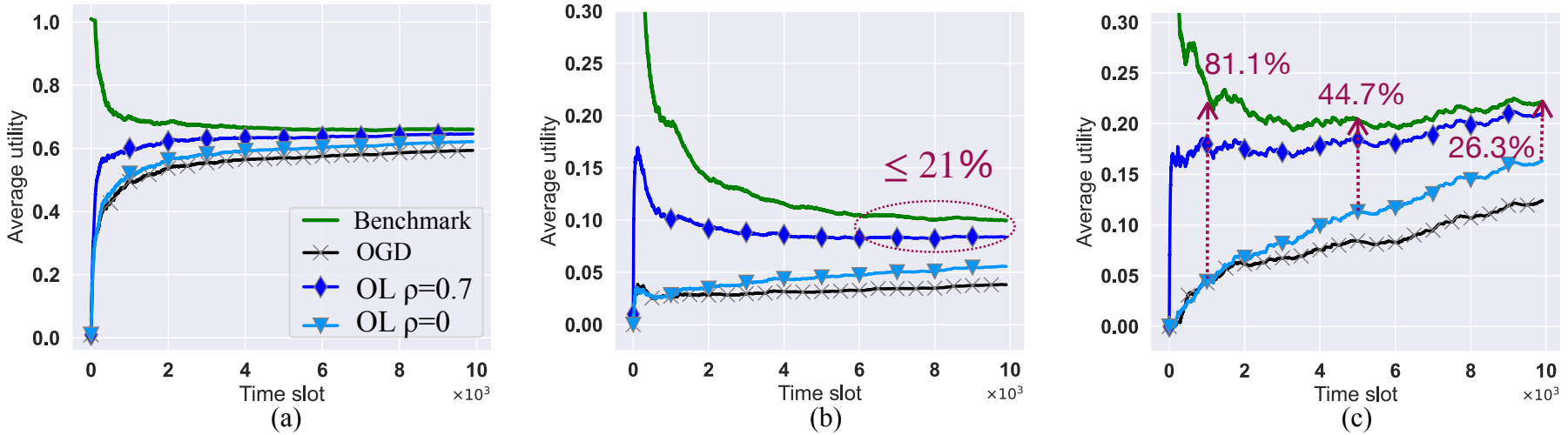


Figure: Cache utility with predictors of different accuracy ($\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_t$ with probability ρ) and request traces from: (a) Zipf distribution; (b) YouTube dataset; (c) MovieLens datasets.

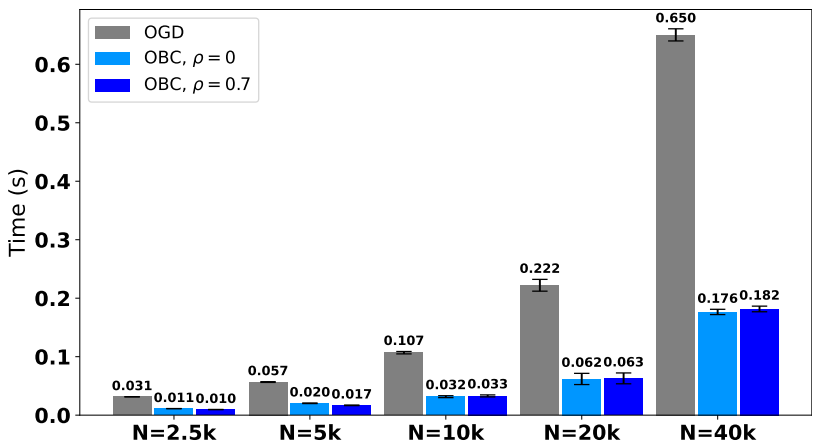


Figure: Average time consumed per decision step for different library size N and horizon $T = 10k$).

OL for: Time-average Constraints

- Optimizing performance s.t. average resource consumption, costs, QoS, etc.
  - Constraints: fixed convex set $\mathcal{X}$ and time-varying functions $h_t(\boldsymbol{x}),\ \forall t$.
  - Need to bound both the Regret and the Constraint Violation:

$$\mathcal{R}_T = \sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t) \Big), \qquad \mathcal{V}_T = \Big\| \Big[\sum_{t=1}^{T} h_t(\boldsymbol{x}_t) \Big] \Big\|$$

- Impossibility result: cannot minimize both rates at the same time.
  - Predictions can improve significantly the factors of both terms.

$$\boldsymbol{x}_t = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{\tau=1}^{t-1} \mathcal{L}_\tau(\boldsymbol{x}, \boldsymbol{\lambda}_\tau) + \langle \tilde{\boldsymbol{q}}_t, \boldsymbol{x} \rangle + \langle \boldsymbol{\lambda}_t, \tilde{h}_t(\boldsymbol{x}) \rangle$$

$$\boldsymbol{\lambda}_{t+1} = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^n_+} \sum_{\tau=1}^{t} \mathcal{L}_\tau(\boldsymbol{x}_\tau, \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \tilde{h}_{t+1}(\boldsymbol{x}) \rangle$$

D. Anderson, G. Iosifidis, D. J. Leith: Lazy Lagrangians for Optimistic Learning With Budget Constraints. IEEE/ACM Trans. Netw. 31(5), 2023.

OL for: Time-average Constraints

- Optimizing performance s.t. average resource consumption, costs, QoS, etc.
  - Constraints: fixed convex set $\mathcal{X}$ and time-varying functions $h_t(\boldsymbol{x}),\ \forall t$.
  - Need to bound both the Regret and the Constraint Violation:

$$\mathcal{R}_T = \sum_{t=1}^{T} \Big(f_t(\boldsymbol{x}^\star) - f_t(\boldsymbol{x}_t) \Big), \qquad \mathcal{V}_T = \Big\| \Big[\sum_{t=1}^{T} h_t(\boldsymbol{x}_t) \Big] \Big\|$$

- Impossibility result: cannot minimize both rates at the same time.
  - Predictions can improve significantly the factors of both terms.

$$\boldsymbol{x}_t = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{\tau=1}^{t-1} \mathcal{L}_\tau(\boldsymbol{x}, \boldsymbol{\lambda}_\tau) + \langle \tilde{\boldsymbol{q}}_t, \boldsymbol{x} \rangle + \langle \boldsymbol{\lambda}_t, \tilde{h}_t(\boldsymbol{x}) \rangle$$

$$\boldsymbol{\lambda}_{t+1} = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^n_+} \sum_{\tau=1}^{t} \mathcal{L}_\tau(\boldsymbol{x}_\tau, \boldsymbol{\lambda}) + \langle \boldsymbol{\lambda}, \tilde{h}_{t+1}(\boldsymbol{x}) \rangle$$

D. Anderson, G. Iosifidis, D. J. Leith: Lazy Lagrangians for Optimistic Learning With Budget Constraints. IEEE/ACM Trans. Netw. 31(5), 2023.

OL with Multiple Predictors

- How to incorporate **multiple** (untrusted) predictors?

N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

OL with Multiple Predictors

- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.

N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

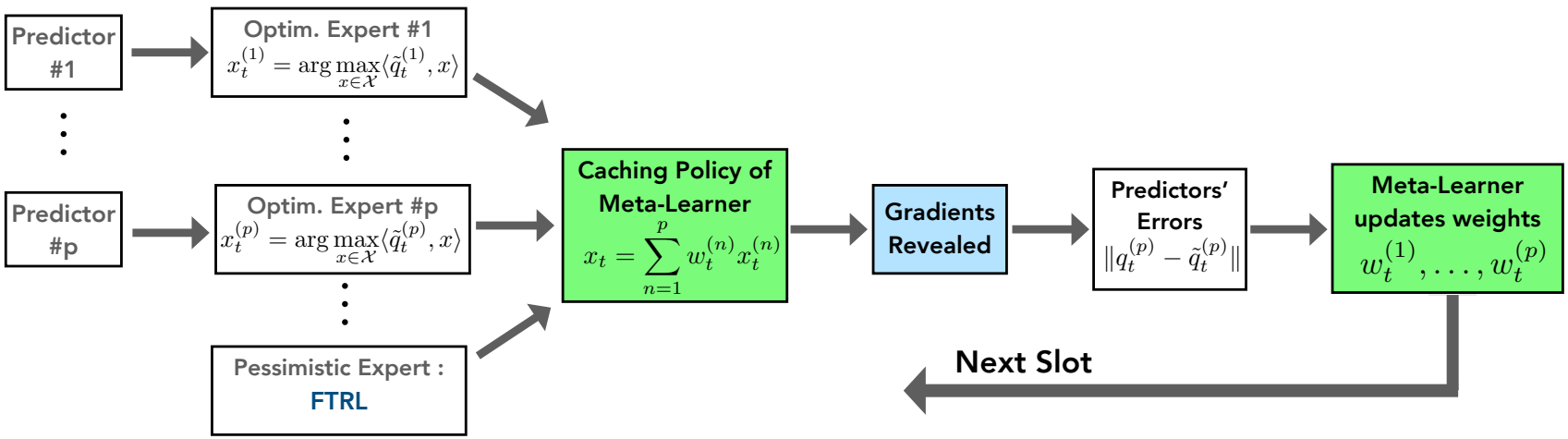
OL with Multiple Predictors

- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.
  - Solution: a meta-learner adapts trusting weights for each predictor.

N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

OL with Multiple Predictors

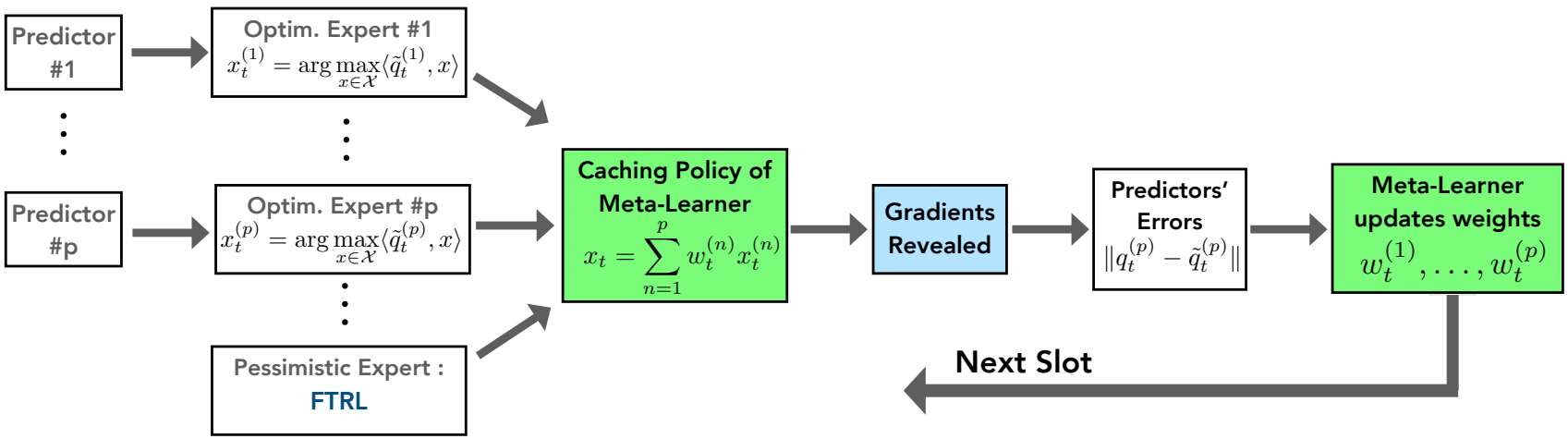
- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.
  - Solution: a meta-learner adapts trusting weights for each predictor.



N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

OL with Multiple Predictors

- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.
  - Solution: a meta-learner adapts trusting weights for each predictor.

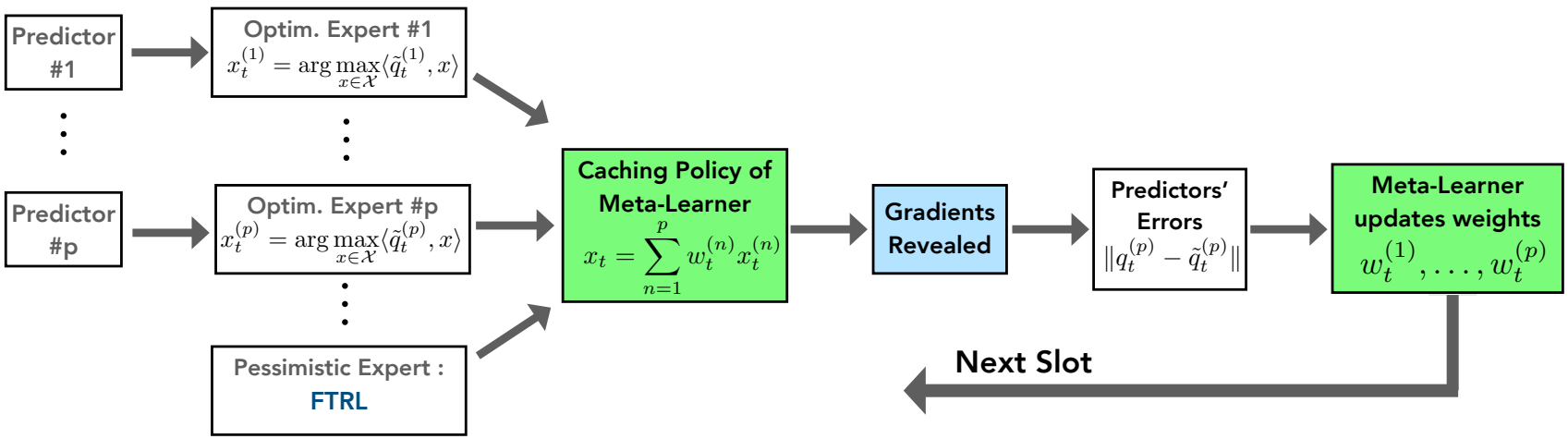


$$\mathcal{R}_T \leq \underbrace{\mathcal{R}_T^{(w)}}_{\text{find best predictor}} + \underbrace{\min_p \mathcal{R}_T^{(p)}}_{\text{regret of best predictor}}$$

N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

OL with Multiple Predictors

- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.
  - Solution: a meta-learner adapts trusting weights for each predictor.

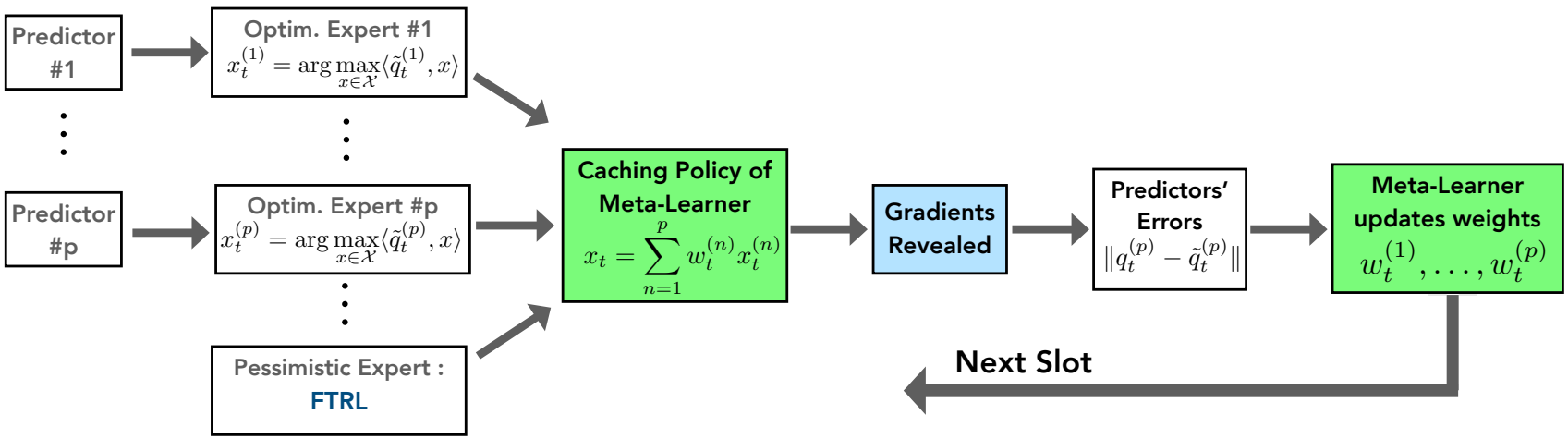


$$\mathcal{R}_T \leq \underbrace{\sqrt{\log P}\sqrt{T}}_{\text{find best predictor}} + \underbrace{\min_p \mathcal{R}_T^{(p)}}_{\text{max accuracy} = \mathcal{O}\sqrt{T}}$$

N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

OL with Multiple Predictors

- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.
  - Solution: a meta-learner adapts trusting weights for each predictor.



$$\mathcal{R}_T \leq \underbrace{\sqrt{\text{meta-mistakes}}}_{\text{find best predictor}} + \underbrace{\min_p \mathcal{R}_T^{(p)}}_{\text{max accuracy} = \mathcal{O}\sqrt{T}}$$

N. Mhaisen, G. Iosifidis, D. Leith: Online Caching With no Regret[...], IEEE Trans. Mob. Comp., 2024.
A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.

Prior Work: Slow-varying functions

- **[A1]** FTRL can be improved if gradients stay close to their average:

$$\mathcal{R}_T \leq \mathcal{O}\left(\sqrt{\sum_t \|\boldsymbol{q}_t - \bar{\boldsymbol{q}}_T\|^2}\right), \quad \text{where} \quad \bar{\boldsymbol{q}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{q}_t$$

  ↪ The algorithm needs a twist for the experts problem.

- **[A2]** If functions have small variation, a new algorithm achieves:

$$\mathcal{R}_T \leq \mathcal{O}\left(\sqrt{D}\right), \quad D = \sum_t \max_{x \in \mathcal{X}} \|\nabla f_t(\boldsymbol{x}) - \nabla f_{t-1}(\boldsymbol{x})\|^2$$

  ↪ Main idea: perform an additional gradient iteration in each round

$$\boldsymbol{x}_{t+1} = \text{Proj}_{\mathcal{X}}\Big(\boldsymbol{x}_t - \eta \nabla f_t(\boldsymbol{x}_t)\Big), \quad \hat{\boldsymbol{x}}_{t+1} = \text{Proj}_{\mathcal{X}}\Big(\boldsymbol{x}_{t+1} - \eta \nabla f_t(\boldsymbol{x}_t)\Big)$$

[A1] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. Machine learning, 80(2):165–188, 2010.

[A2] C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. COLT, 2012.

Prior Work: Slow-varying functions

- **[A1]** FTRL can be improved if gradients stay close to their average:

$$\mathcal{R}_T \leq \mathcal{O}\left(\sqrt{\sum_t \|\boldsymbol{q}_t - \bar{\boldsymbol{q}}_T\|^2}\right), \quad \text{where} \quad \bar{\boldsymbol{q}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{q}_t$$

  $\hookrightarrow$ The algorithm needs a twist for the experts problem.

- **[A2]** If functions have small variation, a new algorithm achieves:

$$\mathcal{R}_T \leq \mathcal{O}\left(\sqrt{D}\right), \quad D = \sum_t \max_{x\in\mathcal{X}} \|\nabla f_t(\boldsymbol{x}) - \nabla f_{t-1}(\boldsymbol{x})\|^2$$

  $\hookrightarrow$ Main idea: perform an additional gradient iteration in each round

$$\boldsymbol{x}_{t+1} = \text{Proj}_{\mathcal{X}}\Big(\boldsymbol{x}_t - \eta\nabla f_t(\boldsymbol{x}_t)\Big), \quad \hat{\boldsymbol{x}}_{t+1} = \text{Proj}_{\mathcal{X}}\Big(\boldsymbol{x}_{t+1} - \eta\nabla f_t(\boldsymbol{x}_t)\Big)$$

[A1] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. Machine learning, 80(2):165–188, 2010.

[A2] C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. COLT, 2012.

Prior Work: Introducing Optimism

- Generalization via predictions for next gradient
  - FTRL with self-concordant barrier and predictions.

$$\hat{\boldsymbol{x}}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_t + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x}\rangle + \underbrace{r(\boldsymbol{x})}_{\text{Self-Conc}} \right\}$$

$\hookrightarrow$ No need to have slow-varying environment.

$$\mathcal{R}_T \leq \sqrt{4 \max_{\boldsymbol{x}\in\mathcal{X}} r(\boldsymbol{x}) \left[\sum_{t=1}^{T} \underbrace{\left\| \boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t \right\|_{t,\star}^2}_{\text{dual norm wrt } r(\boldsymbol{x}_t)} + 1 \right]}$$

- Similar bound when using optimistic OMD.

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_t, \boldsymbol{x}\rangle + \underbrace{D_r(\boldsymbol{x}, \boldsymbol{x}_t)}_{\text{Bregman Diverg.}}, \quad \hat{\boldsymbol{x}}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \langle \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x}\rangle + D_r(\boldsymbol{x}, \boldsymbol{x}_{t+1})$$

[B1] A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.
[B2] Rakhlin, Sridharan: Optimization, Learning, and Games with Predictable Sequences. NIPS 2013

Prior Work: Introducing Optimism

- Generalization via predictions for next gradient
  - FTRL with self-concordant barrier and predictions.

$$\hat{\boldsymbol{x}}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \left\{ \langle \boldsymbol{q}_t + \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x}\rangle + \underbrace{r(\boldsymbol{x})}_{\text{Self-Conc}} \right\}$$

  $\hookrightarrow$ No need to have slow-varying environment.

$$\mathcal{R}_T \leq \sqrt{4 \max_{\boldsymbol{x}\in\mathcal{X}} r(\boldsymbol{x}) \left[\sum_{t=1}^{T} \underbrace{\left\| \boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t \right\|_{t,\star}^2}_{\text{dual norm wrt } r(\boldsymbol{x}_t)} + 1 \right]}$$

  - Similar bound when using optimistic OMD.

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{q}_t, \boldsymbol{x}\rangle + \underbrace{D_r(\boldsymbol{x}, \boldsymbol{x}_t)}_{\text{Bregman Diverg.}}, \quad \hat{\boldsymbol{x}}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \langle \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{x}\rangle + D_r(\boldsymbol{x}, \boldsymbol{x}_{t+1})$$

[B1] A. Rakhlin, K. Sridharan: Online Learning with Predictable Sequences. COLT 2013.
[B2] Rakhlin, Sridharan: Optimization, Learning, and Games with Predictable Sequences. NIPS 2013

Prior Work: Different Prediction Metrics

We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$

- What if we know a coordinate, e.g., $q_{t+1,1}$, of the next gradient $\boldsymbol{q}_{t+1} \in \mathbb{R}^n$?

$$\mathcal{R}_T \leq \mathcal{O}\left(n^2 \log T\right)$$

- What if we have a directional-only prediction? (easier to obtain, perhaps)

$$\langle \tilde{\boldsymbol{q}}_t, \boldsymbol{q}_t \rangle \geq \alpha \|\boldsymbol{q}_t\| \quad \rightarrow \quad \mathcal{R}_T \leq \mathcal{O}\left(\frac{1}{\alpha} \log T\right)$$

- What if we are interested in a sequence of comparators $\{\boldsymbol{x}_t\}_{t+1}^T$

$$\mathcal{R}_T = \mathcal{O}((1 + C_T)\sqrt{1 + D_T});\ C_T \propto \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|$$

[D1] E. Hazan et al. Online learning with prior knowledge. COLT, 2007.
[D2] O. Dekel, et al., Online Learning with a Hint. NIPS 2017.
[D3] A. Bhaskara, et al., Online Learning with Imperfect Hints. ICML 2020.
[D4] PZ Scroccaro, et., al., Adaptive composite online optimization: Predictions in static and dynamic environments., IEEE TAC 2024

Prior Work: Different Prediction Metrics

We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$

- What if we know a coordinate, e.g., $q_{t+1,1}$, of the next gradient $\boldsymbol{q}_{t+1} \in \mathbb{R}^n$?

$$\mathcal{R}_T \leq \mathcal{O}\left(n^2 \log T\right)$$

- What if we have a directional-only prediction? (easier to obtain, perhaps)

$$\langle \tilde{\boldsymbol{q}}_t, \boldsymbol{q}_t \rangle \geq \alpha \|\boldsymbol{q}_t\| \quad \rightarrow \quad \mathcal{R}_T \leq \mathcal{O}\left(\frac{1}{\alpha} \log T\right)$$

- What if we are interested in a sequence of comparators $\{\boldsymbol{x}_t\}_{t+1}^T$

$$\mathcal{R}_T = \mathcal{O}((1 + C_T)\sqrt{1 + D_T});\ C_T \propto \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|$$

[D1] E. Hazan et al. Online learning with prior knowledge. COLT, 2007.
[D2] O. Dekel, et al., Online Learning with a Hint. NIPS 2017.
[D3] A. Bhaskara, et al., Online Learning with Imperfect Hints. ICML 2020.
[D4] PZ Scroccaro, et., al., Adaptive composite online optimization: Predictions in static and dynamic environments., IEEE TAC 2024

Prior Work: Different Prediction Metrics

We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$

- What if we know a coordinate, e.g., $q_{t+1,1}$, of the next gradient $\boldsymbol{q}_{t+1} \in \mathbb{R}^n$?

$$\mathcal{R}_T \leq \mathcal{O}\left(n^2 \log T\right)$$

- What if we have a directional-only prediction? (easier to obtain, perhaps)

$$\langle \tilde{\boldsymbol{q}}_t, \boldsymbol{q}_t \rangle \geq \alpha \|\boldsymbol{q}_t\| \quad \rightarrow \quad \mathcal{R}_T \leq \mathcal{O}\left(\frac{1}{\alpha} \log T\right)$$

- What if we are interested in a sequence of comparators $\{\boldsymbol{x}_t\}_{t+1}^T$

$$\mathcal{R}_T = \mathcal{O}((1 + C_T)\sqrt{1 + D_T});\ C_T \propto \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|$$

[D1] E. Hazan et al. Online learning with prior knowledge. COLT, 2007.
[D2] O. Dekel, et al., Online Learning with a Hint. NIPS 2017.
[D3] A. Bhaskara, et al., Online Learning with Imperfect Hints. ICML 2020.
[D4] PZ Scroccaro, et., al., Adaptive composite online optimization: Predictions in static and dynamic environments., IEEE TAC 2024

Prior Work: Different Prediction Metrics

We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2$

- What if we know a coordinate, e.g., $q_{t+1,1}$, of the next gradient $\boldsymbol{q}_{t+1} \in \mathbb{R}^n$?

$$\mathcal{R}_T \leq \mathcal{O}\left(n^2 \log T\right)$$

- What if we have a directional-only prediction? (easier to obtain, perhaps)

$$\langle \tilde{\boldsymbol{q}}_t, \boldsymbol{q}_t \rangle \geq \alpha \|\boldsymbol{q}_t\| \quad \rightarrow \quad \mathcal{R}_T \leq \mathcal{O}\left(\frac{1}{\alpha} \log T\right)$$

- What if we are interested in a sequence of comparators $\{\boldsymbol{x}_t\}_{t+1}^T$

$$\mathcal{R}_T = \mathcal{O}((1 + C_T)\sqrt{1 + D_T});\ C_T \propto \sum_{t=1}^{T} \|\boldsymbol{x}_t^\star - \boldsymbol{x}_{t-1}^\star\|$$

[D1] E. Hazan et al. Online learning with prior knowledge. COLT, 2007.
[D2] O. Dekel, et al., Online Learning with a Hint. NIPS 2017.
[D3] A. Bhaskara, et al., Online Learning with Imperfect Hints. ICML 2020.
[D4] PZ Scroccaro, et., al., Adaptive composite online optimization: Predictions in static and dynamic environments., IEEE TAC 2024

Prior Work: Applications

- Portfolio management, Rental Strategies, Bidding in Auctions,...
  - C. Tsai, et al, Data-Dependent Bounds for Online Portfolio Selection[...], NeurIPS'23.
  - A. Bhattacharya, et al, ML advised algorithms for the ski rental problem with a discount, Theor. Comput. Sci.'22.
  - W. Zhang, et al, Leveraging The Hints: Adaptive Bidding in Repeated First-Price Auctions, NeurIPS 2022.
- Training of Neural Networks
  - C. Daskalakis, et al., Training GANs with Optimism. ICLR, 2018.
  - P. Mertikopoulos, et al, Optimistic mirror descent in saddle-point problems, ICLR'19.
- Caching, Slice Reservation, Network Control,...
  - T. Lykouris, et al., Competitive Caching with ML Advice, J. ACM 68(4), 2021.
  - F. Faticanti, et al., Optimistic Online Caching for Batched Requests, Comp. Net.'24.
  - J. Monteil, et al, Reservation of Virtualized Resources with OL, ICC'23.

Prior Work: Applications

- Portfolio management, Rental Strategies, Bidding in Auctions,...
  - C. Tsai, et al, Data-Dependent Bounds for Online Portfolio Selection[...], NeurIPS'23.
  - A. Bhattacharya, et al, ML advised algorithms for the ski rental problem with a discount, Theor. Comput. Sci.'22.
  - W. Zhang, et al, Leveraging The Hints: Adaptive Bidding in Repeated First-Price Auctions, NeurIPS 2022.
- Training of Neural Networks
  - C. Daskalakis, et al., Training GANs with Optimism. ICLR, 2018.
  - P. Mertikopoulos, et al, Optimistic mirror descent in saddle-point problems, ICLR'19.
- Caching, Slice Reservation, Network Control,...
  - T. Lykouris, et al., Competitive Caching with ML Advice, J. ACM 68(4), 2021.
  - F. Faticanti, et al., Optimistic Online Caching for Batched Requests, Comp. Net.'24.
  - J. Monteil, et al, Reservation of Virtualized Resources with OL, ICC'23.

Prior Work: Applications

- Portfolio management, Rental Strategies, Bidding in Auctions,...
  - C. Tsai, et al, Data-Dependent Bounds for Online Portfolio Selection[...], NeurIPS'23.
  - A. Bhattacharya, et al, ML advised algorithms for the ski rental problem with a discount, Theor. Comput. Sci.'22.
  - W. Zhang, et al, Leveraging The Hints: Adaptive Bidding in Repeated First-Price Auctions, NeurIPS 2022.
- Training of Neural Networks
  - C. Daskalakis, et al., Training GANs with Optimism. ICLR, 2018.
  - P. Mertikopoulos, et al, Optimistic mirror descent in saddle-point problems, ICLR'19.
- Caching, Slice Reservation, Network Control,...
  - T. Lykouris, et al., Competitive Caching with ML Advice, J. ACM 68(4), 2021.
  - F. Faticanti, et al., Optimistic Online Caching for Batched Requests, Comp. Net.'24.
  - J. Monteil, et al, Reservation of Virtualized Resources with OL, ICC'23.

Table of Contents

1 Background & Motivation

2 Preliminaries & Convex OL

3 Non-convex OL

4 OL & Non-stochastic Control

Optimistic Discrete Caching

Recall: OL for caching

- Obtain request prediction $\tilde{\boldsymbol{q}}_t$
- Commit to caching policy $\hat{\boldsymbol{x}}_t \in \hat{\mathcal{X}} \doteq \left\{ [0,1]^N \;\middle|\; \sum_n y_n \leq C \right\}$.
- User request is revealed $\boldsymbol{q}_t \in \{0,1\}^N$,
- Cache hit $f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$, Prediction error $\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|$

$$\boldsymbol{x}^\star = \arg\max_{x \in \mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle. \text{ (Already Discrete)}$$

Optimistic Discrete Caching

OL for **Discrete** caching

- Obtain request prediction $\tilde{\boldsymbol{q}}_t$
- ~~Commit to caching policy $\hat{\boldsymbol{x}}_t \in \hat{\mathcal{X}} \doteq \{[0,1]^N \mid \sum_n y_n \leq C\}$.~~
- Commit to caching policy $\boldsymbol{x}_t \in \mathcal{X} \doteq \{\{0,1\}^N \mid \sum_n y_n \leq C\}$.
- User request is revealed $\boldsymbol{q}_t \in \{0,1\}^N$,
- Cache hit $f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$, Prediction error $\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|$

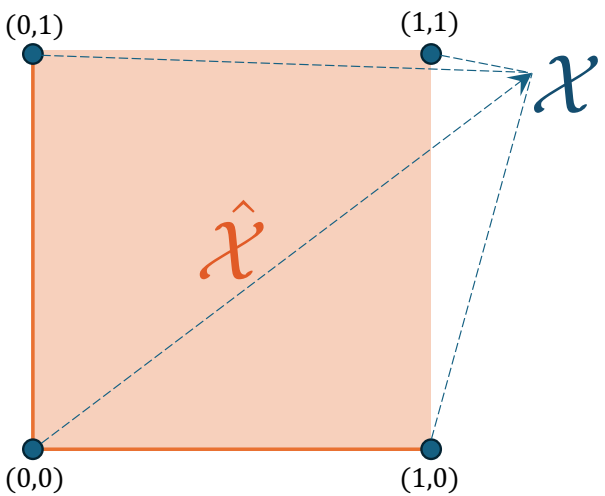
$$\boldsymbol{x}^\star = \arg\max_{x \in \mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle. \text{ (Already Discrete)}$$

Optimistic Discrete Caching

OL for **Discrete** caching

- Obtain request prediction $\tilde{\boldsymbol{q}}_t$
- ~~Commit to caching policy $\hat{\boldsymbol{x}}_t \in \hat{\mathcal{X}} \doteq \{[0,1]^N \mid \sum_n y_n \leq C\}$.~~
- Commit to caching policy $\boldsymbol{x}_t \in \mathcal{X} \doteq \{\{0,1\}^N \mid \sum_n y_n \leq C\}$.
- User request is revealed $\boldsymbol{q}_t \in \{0,1\}^N$,
- Cache hit $f_t(\boldsymbol{x}_t) \doteq \langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$, Prediction error $\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|$

$$\boldsymbol{x}^\star = \arg\max_{x \in \mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle. \text{ (Already Discrete)}$$



OFTPL for Discrete Actions

- Another way to regularize FTL update: Perturbing.
- FTPerturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \gamma_t, \boldsymbol{x} \rangle$, $\gamma_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Discrete solution. Efficient Quantile finding problem.

FTPL Regret Bound

The iterates

$$\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1} \eta_t \gamma_t, \boldsymbol{x} \rangle$$

achieve the regret bound:

$$\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \eta_T C \sqrt{\ln(N/C)} + \sum_{t=1}^T \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2}{eta_t} \leq \eta_{T+1}$$

- $\eta_t \propto \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|_1^2}$, $\rightarrow \mathcal{R}_T = \mathcal{O}(\ln(N)\sqrt{C}\sqrt{\sum_{t=1}^T \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2})$

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15.

N. Mhaisen, A. Sinha, G. Paschos, G. Iosifidis, Optimistic No-regret Algorithms for Discrete Caching, ACM SIGMETRICS '22.

OFTPL for Discrete Actions

- Another way to regularize FTL update: Perturbing.
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \boldsymbol{\gamma}_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Discrete solution. Efficient Quantile finding problem.

FTPL Regret Bound

The iterates

$$\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}\eta_t\gamma_t, \boldsymbol{x}\rangle$$

achieve the regret bound:

$$\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \eta_T C\sqrt{\ln(N/C)} + \sum_{t=1}^T \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2}{eta_t} \leq \eta_{T+1}$$

- $\eta_t \propto \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|_1^2}$, $\rightarrow \mathcal{R}_T = \mathcal{O}(\ln(N)\sqrt{C}\sqrt{\sum_{t=1}^T \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2})$

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15.

N. Mhaisen, A. Sinha, G. Paschos, G. Iosifidis, Optimistic No-regret Algorithms for Discrete Caching, ACM SIGMETRICS '22.

OFTPL for Discrete Actions

- Another way to regularize FTL update: Perturbing.
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \boldsymbol{\gamma}_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Discrete solution. Efficient Quantile finding problem.

FTPL Regret Bound

The iterates

$$\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}\eta_t\gamma_t, \boldsymbol{x}\rangle$$

achieve the regret bound:

$$\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \eta_T C\sqrt{\ln(N/C)} + \sum_{t=1}^T \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2}{eta_t} \leq \eta_{T+1}$$

- $\eta_t \propto \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|_1^2}$, $\rightarrow \mathcal{R}_T = \mathcal{O}(\ln(N)\sqrt{C}\sqrt{\sum_{t=1}^T \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2})$

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15.

N. Mhaisen, A. Sinha, G. Paschos, G. Iosifidis, Optimistic No-regret Algorithms for Discrete Caching, ACM SIGMETRICS '22.

OFTPL for Discrete Actions

- Another way to regularize FTL update: Perturbing.
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \boldsymbol{\gamma}_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Discrete solution. Efficient Quantile finding problem.

FTPL Regret Bound

The iterates

$$\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}\eta_t\gamma_t, \boldsymbol{x}\rangle$$

achieve the regret bound:

$$\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \eta_T C\sqrt{\ln(N/C)} + \sum_{t=1}^T \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2}{eta_t} \leq \eta_{T+1}$$

- $\eta_t \propto \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|_1^2}$, $\rightarrow \mathcal{R}_T = \mathcal{O}(\ln(N)\sqrt{C}\sqrt{\sum_{t=1}^T \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2})$

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15.

N. Mhaisen, A. Sinha, G. Paschos, G. Iosifidis, Optimistic No-regret Algorithms for Discrete Caching, ACM SIGMETRICS '22.

OFTPL for Discrete Actions

- Another way to regularize FTL update: Perturbing.
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \boldsymbol{\gamma}_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Discrete solution. Efficient Quantile finding problem.

FTPL Regret Bound

The iterates

$$\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}\eta_t\gamma_t, \boldsymbol{x}\rangle$$

achieve the regret bound:

$$\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \eta_T C\sqrt{\ln(N/C)} + \sum_{t=1}^T \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2}{eta_t} \leq \eta_{T+1}$$

- $\eta_t \propto \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|_1^2}$, $\rightarrow \mathcal{R}_T = \mathcal{O}(\ln(N)\sqrt{C}\sqrt{\sum_{t=1}^T \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2})$

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15.

N. Mhaisen, A. Sinha, G. Paschos, G. Iosifidis, Optimistic No-regret Algorithms for Discrete Caching, ACM SIGMETRICS '22.

OFTPL for Discrete Actions

- Another way to regularize FTL update: Perturbing.
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \boldsymbol{\gamma}_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Discrete solution. Efficient Quantile finding problem.

FTPL Regret Bound

The iterates

$$\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \tilde{\boldsymbol{q}}_{t+1}\eta_t\gamma_t, \boldsymbol{x}\rangle$$

achieve the regret bound:

$$\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \eta_T C\sqrt{\ln(N/C)} + \sum_{t=1}^T \frac{\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2}{eta_t} \leq \eta_{T+1}$$

- $\eta_t \propto \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|_1^2}$, $\rightarrow \mathcal{R}_T = \mathcal{O}(\ln(N)\sqrt{C}\sqrt{\sum_{t=1}^T \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|_1^2})$

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15.

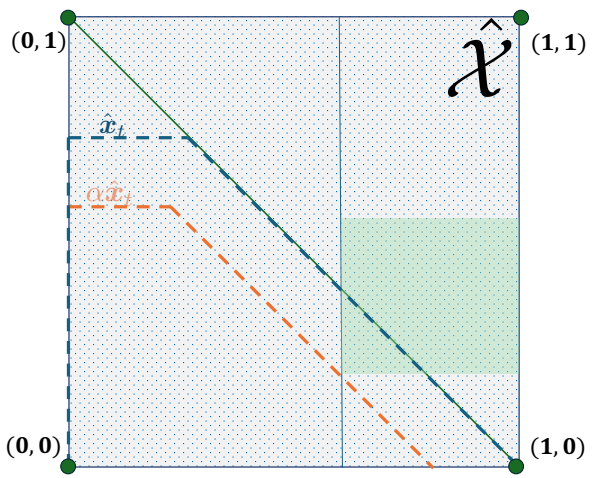
N. Mhaisen, A. Sinha, G. Paschos, G. Iosifidis, Optimistic No-regret Algorithms for Discrete Caching, ACM SIGMETRICS '22.

OFTR/PL with Discrete Actions + Capacity Constraints

$$\left.\begin{array}{ll} \text{Maximize} & \langle \boldsymbol{p}, \boldsymbol{x} \rangle \\ \text{Subject to} & \boldsymbol{x} \in \mathcal{X}, \quad \langle \boldsymbol{s}, \boldsymbol{x} \rangle \leq B \end{array}\right\} \text{Knapsack}$$

(Optimistic) Online Learning of (NP-Hard) problems: Two elements

- Approximate benchmark: $\mathcal{R}_T^{\alpha}(\{\boldsymbol{x}_t\}_T) = \alpha \langle \boldsymbol{q}_{1:T}, \boldsymbol{x}^\star \rangle - \langle \boldsymbol{q}_{1:T}, \boldsymbol{x}_t \rangle$, $\alpha < 1$
- Approximate rounding: $\mathbb{E}[\boldsymbol{x}_t] \geq \alpha \hat{\boldsymbol{x}}_t$ s.t. $\Pr[\langle \boldsymbol{s}, \boldsymbol{x}_t \rangle \leq B] = 1$.



Disc. to Cont. Reduction

$$\underbrace{\mathcal{R}_T(\{\hat{\boldsymbol{x}}_t\}_T)}_{\text{Possibly Optimistic}} \leq A \rightarrow \mathbb{E}[\mathcal{R}_T^{\alpha}(\{\boldsymbol{x}_t\}_T)] \leq A$$

Table of Contents

1 Background & Motivation

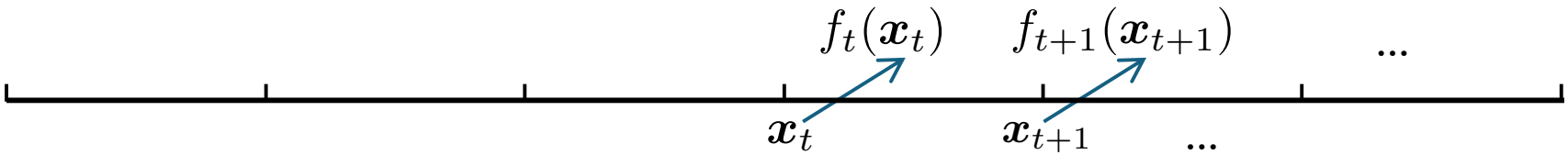
2 Preliminaries & Convex OL

3 Non-convex OL

4 OL & Non-stochastic Control

OCO with Memory

- Regular OCO: At each t, the learner pays the cost of $\boldsymbol{x}_t$:



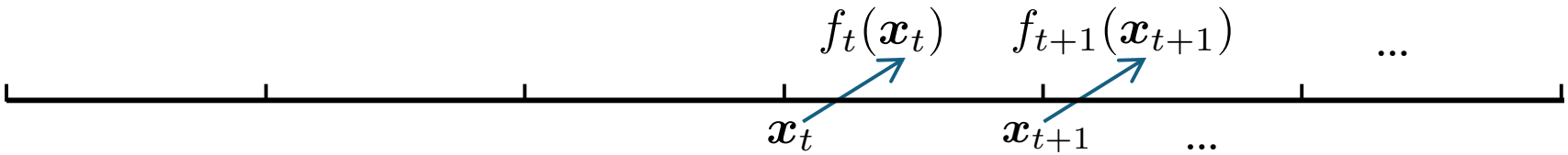
- OCO with memory (OCO-M): Pays for the past m decisions, $\boldsymbol{x}_t$:

- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

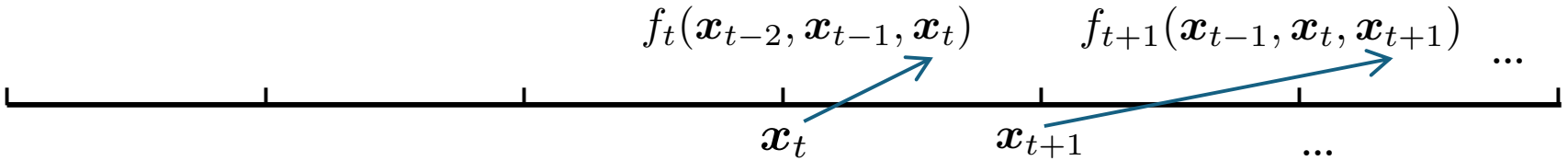
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

OCO with Memory

- Regular OCO: At each t, the learner pays the cost of $\boldsymbol{x}_t$:



- OCO with memory (OCO-M): Pays for the past m decisions, $\boldsymbol{x}_t$:

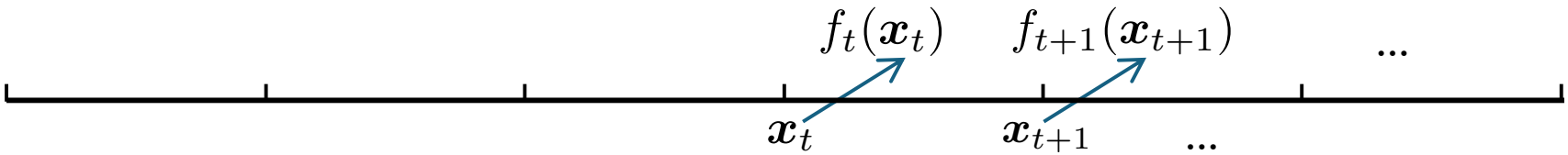


- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

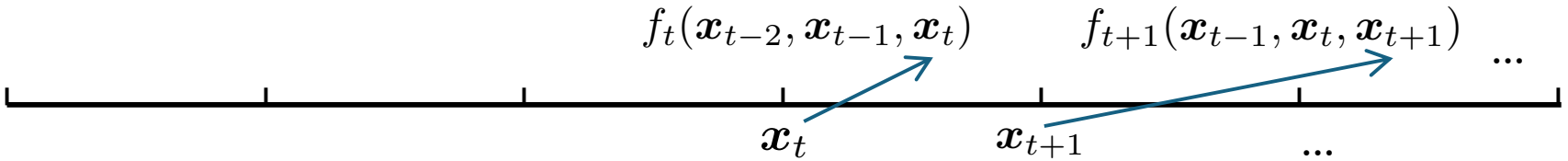
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

OCO with Memory

- Regular OCO: At each t, the learner pays the cost of $\boldsymbol{x}_t$:



- OCO with memory (OCO-M): Pays for the past m decisions, $\boldsymbol{x}_t$:

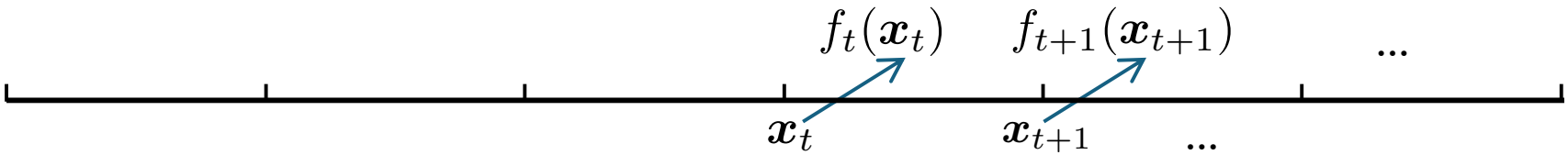


- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

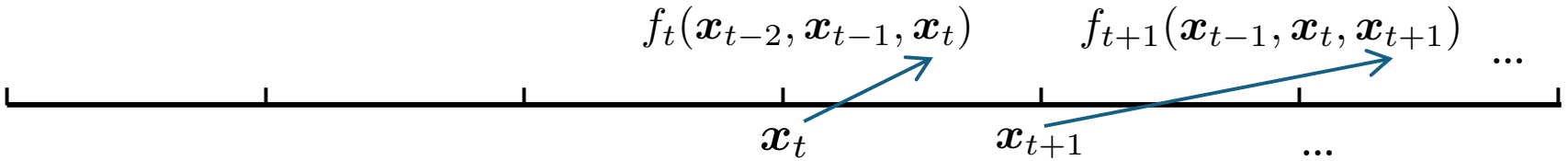
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

OCO with Memory

- Regular OCO: At each t, the learner pays the cost of $\boldsymbol{x}_t$:



- OCO with memory (OCO-M): Pays for the past m decisions, $\boldsymbol{x}_t$:

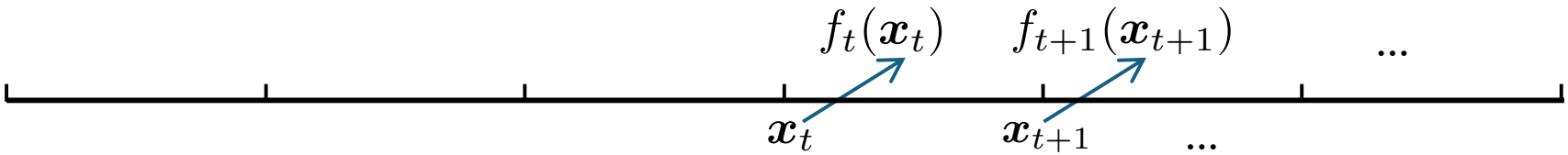


- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

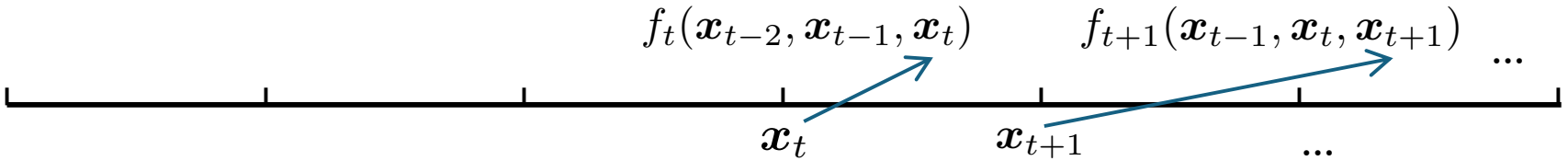
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

OCO with Memory

- Regular OCO: At each t, the learner pays the cost of $\boldsymbol{x}_t$:



- OCO with memory (OCO-M): Pays for the past m decisions, $\boldsymbol{x}_t$:



- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

Non-stochastic Control (NSC) 1/2

NSC Problem: start at some state $\boldsymbol{x}_t$

- Commit to action $\boldsymbol{u}_t$
- Transition to a state $\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{w}_t$
- Incur a cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$

$$\text{Policy Regret:} \quad \mathcal{R}_T^{\text{pol}}(\mathcal{A}) \doteq \left[c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \min_{\pi \in \Pi} c_t(\boldsymbol{x}_t^{\pi}, \boldsymbol{u}_t^{\pi}) \right]$$

- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 1/2

NSC Problem: start at some state $\boldsymbol{x}_t$

- Commit to action $\boldsymbol{u}_t$
- Transition to a state $\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{w}_t$
- Incur a cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Policy Regret: $$\mathcal{R}_T^{\text{pol}}(\mathcal{A}) \doteq \left[c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \min_{\pi \in \Pi} c_t(\boldsymbol{x}_t^{\pi}, \boldsymbol{u}_t^{\pi}) \right]$$

- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 1/2

NSC Problem: start at some state $\boldsymbol{x}_t$

- Commit to action $\boldsymbol{u}_t$
- Transition to a state $\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{w}_t$
- Incur a cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$

Policy Regret: $$\mathcal{R}_T^{\text{pol}}(\mathcal{A}) \doteq \left[c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \min_{\pi \in \Pi} c_t(\boldsymbol{x}_t^{\pi}, \boldsymbol{u}_t^{\pi}) \right]$$

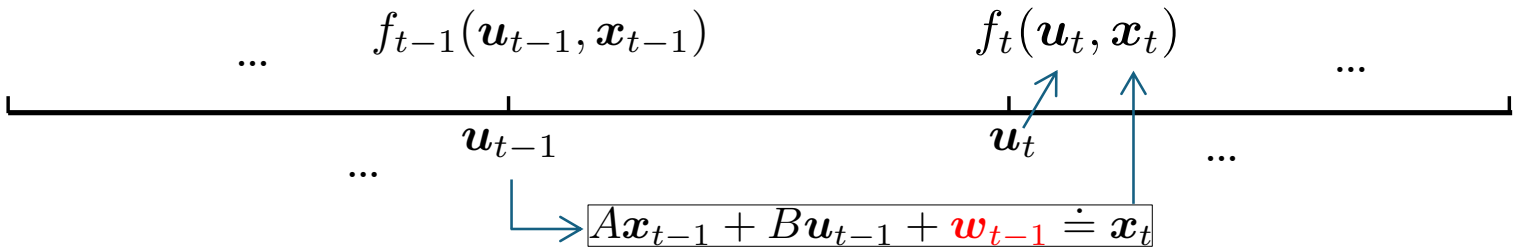
- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 1/2

NSC Problem: start at some state $\boldsymbol{x}_t$

- Commit to action $\boldsymbol{u}_t$
- Transition to a state $\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{w}_t$
- Incur a cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$



Policy Regret: $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) \doteq \left[c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \min_{\pi \in \Pi} c_t(\boldsymbol{x}_t^\pi, \boldsymbol{u}_t^\pi) \right]$

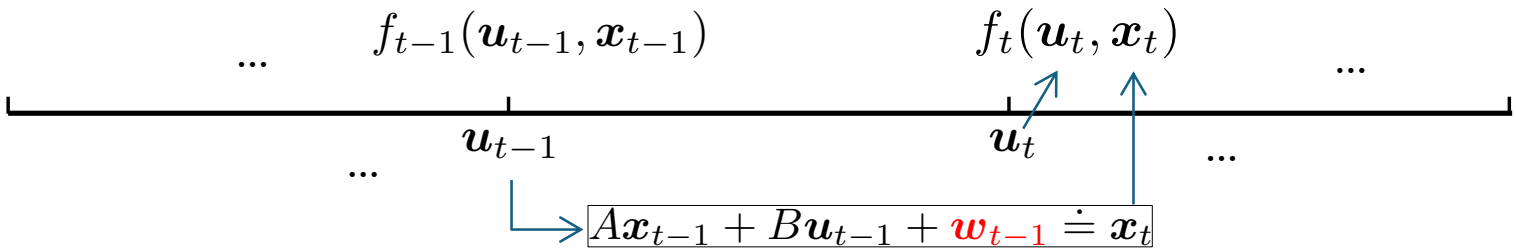
- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 1/2

NSC Problem: start at some state $\boldsymbol{x}_t$

- Commit to action $\boldsymbol{u}_t$
- Transition to a state $\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \underline{\boldsymbol{w}_t}$ *adversarial noise*
- Incur a cost $\underline{c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)}$ *adversarial cost*



Policy Regret: $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) \doteq \left[c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \min_{\pi \in \Pi} c_t(\boldsymbol{x}_t^\pi, \boldsymbol{u}_t^\pi) \right]$

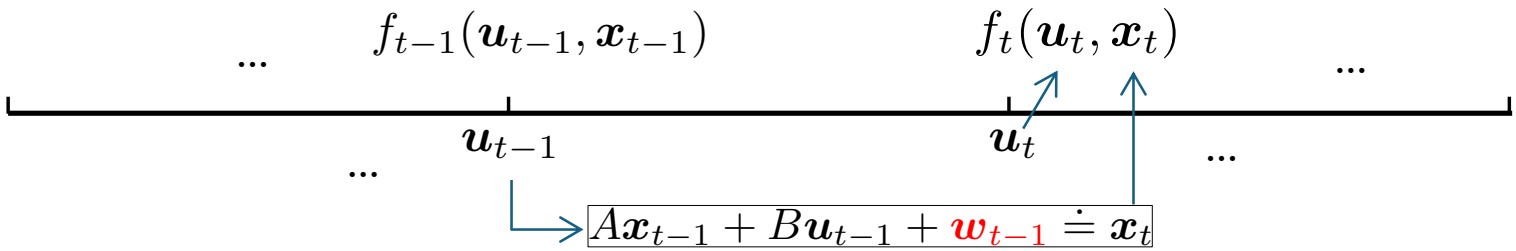
- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 1/2

NSC Problem: start at some state $\boldsymbol{x}_t$

- Commit to action $\boldsymbol{u}_t$
- Transition to a state $\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \underline{\boldsymbol{w}_t}$ *adversarial noise*
- Incur a cost $\underline{c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)}$ *adversarial cost*



$$\text{Policy Regret:} \quad \mathcal{R}_T^{\text{pol}}(\mathcal{A}) \doteq \left[c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \min_{\pi \in \Pi} c_t(\boldsymbol{x}_t^\pi, \boldsymbol{u}_t^\pi) \right]$$

- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 2/2

- Competitor policy class: Disturbance Action Controllers
  $\Pi : \boldsymbol{u}_t = K\boldsymbol{x}_t + \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- DAC can approximate all linear state feedback controller. $\rightarrow$
- Under the DAC class $f_t(M) = c_t(\boldsymbol{x}(\pi), \boldsymbol{x}(\pi))$ is convex in M

Algorithm 1: GPC

Input: An intrinsically stable LTI system (A, B), $\eta \approx 1/g\sqrt{T}$
Output: $\{\boldsymbol{u}_t\}_{t=1}^{T}$

1 **for** $t = 1, 2, \ldots, T$ **do**
2 Use action $\boldsymbol{u}_t = \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
3 Observe cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient $G_t(M_t)$
4 Observe new state $\boldsymbol{x}_{t+1}$
5 Record the disturbance $\boldsymbol{w}_t = \boldsymbol{x}_{t+1} - A\boldsymbol{x}_t - B\boldsymbol{u}_t$
6 Calculate M_{t+1} via OGD: $M_{t+1} = \Pi_{\mathcal{M}}(M_t + \eta G_t)$
end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{T}$ (i.e., converges to the best DAC policy)

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 2/2

- Competitor policy class: Disturbance Action Controllers
  $\Pi : \boldsymbol{u}_t = K\boldsymbol{x}_t + \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- DAC can approximate all linear state feedback controller. $\rightarrow$
- Under the DAC class $f_t(M) = c_t(\boldsymbol{x}(\pi), \boldsymbol{x}(\pi))$ is convex in M

Algorithm 1: GPC

Input: An intrinsically stable LTI system (A, B), $\eta \approx {}^{1}/_{g\sqrt{T}}$
Output: $\{\boldsymbol{u}_t\}_{t=1}^{T}$

for $t = 1, 2, \ldots, T$ **do**
- Use action $\boldsymbol{u}_t = \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- Observe cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient $G_t(M_t)$
- Observe new state $\boldsymbol{x}_{t+1}$
- Record the disturbance $\boldsymbol{w}_t = \boldsymbol{x}_{t+1} - A\boldsymbol{x}_t - B\boldsymbol{u}_t$
- Calculate M_{t+1} via OGD: $M_{t+1} = \Pi_{\mathcal{M}}(M_t + \eta G_t)$

end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{T}$ (i.e., converges to the best DAC policy)

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 2/2

- Competitor policy class: Disturbance Action Controllers
  $\Pi : \boldsymbol{u}_t = K\boldsymbol{x}_t + \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- DAC can approximate all linear state feedback controller. $\rightarrow$
- Under the DAC class $f_t(M) = c_t(\boldsymbol{x}(\pi), \boldsymbol{x}(\pi))$ is convex in M

Algorithm 1: GPC

Input: An intrinsically stable LTI system (A, B), $\eta \approx {}^{1}/_{g\sqrt{T}}$
Output: $\{\boldsymbol{u}_t\}_{t=1}^{T}$

```
for t = 1, 2, ..., T do
    Use action u_t = Σ_{j=1}^{p} M_t^{[j]} w_{t-j}
    Observe cost c_t(x_t, u_t) and record the gradient G_t(M_t)
    Observe new state x_{t+1}
    Record the disturbance w_t = x_{t+1} − A x_t − B u_t
    Calculate M_{t+1} via OGD: M_{t+1} = Π_M(M_t + η G_t)
end
```

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{T}$ (i.e., converges to the best DAC policy)

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 2/2

- Competitor policy class: Disturbance Action Controllers
  $\Pi : \boldsymbol{u}_t = K\boldsymbol{x}_t + \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- DAC can approximate all linear state feedback controller. $\rightarrow$
- Under the DAC class $f_t(M) = c_t(\boldsymbol{x}(\pi), \boldsymbol{x}(\pi))$ is convex in M

Algorithm 1: GPC

Input: An intrinsically stable LTI system (A, B), $\eta \approx 1/g\sqrt{T}$
Output: $\{\boldsymbol{u}_t\}_{t=1}^{T}$

for $t = 1, 2, \ldots, T$ **do**
- Use action $\boldsymbol{u}_t = \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- Observe cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient $G_t(M_t)$
- Observe new state $\boldsymbol{x}_{t+1}$
- Record the disturbance $\boldsymbol{w}_t = \boldsymbol{x}_{t+1} - A\boldsymbol{x}_t - B\boldsymbol{u}_t$
- Calculate M_{t+1} via OGD: $M_{t+1} = \Pi_{\mathcal{M}}(M_t + \eta G_t)$

end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{T}$ (i.e., converges to the best DAC policy)

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Non-stochastic Control (NSC) 2/2

- Competitor policy class: Disturbance Action Controllers
  $\Pi : \boldsymbol{u}_t = K\boldsymbol{x}_t + \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- DAC can approximate all linear state feedback controller. $\rightarrow$
- Under the DAC class $f_t(M) = c_t(\boldsymbol{x}(\pi), \boldsymbol{x}(\pi))$ is convex in M

Algorithm 1: GPC

Input: An intrinsically stable LTI system (A, B), $\eta \approx 1/g\sqrt{T}$
Output: $\{\boldsymbol{u}_t\}_{t=1}^{T}$

for $t = 1, 2, \ldots, T$ **do**
- Use action $\boldsymbol{u}_t = \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}$
- Observe cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient $G_t(M_t)$
- Observe new state $\boldsymbol{x}_{t+1}$
- Record the disturbance $\boldsymbol{w}_t = \boldsymbol{x}_{t+1} - A\boldsymbol{x}_t - B\boldsymbol{u}_t$
- Calculate M_{t+1} via OGD: $M_{t+1} = \Pi_{\mathcal{M}}(M_t + \eta G_t)$

end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{T}$ (i.e., converges to the best DAC policy)

Agarwal et. al., Online control with adversarial disturbances , ICML '19

Optimistic NSC (1/3)

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable

Flaspohler et. al., Online Learning with Optimism and Delay, ICML '21

Optimistic NSC (1/3)

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable

Flaspohler et. al., Online Learning with Optimism and Delay, ICML '21

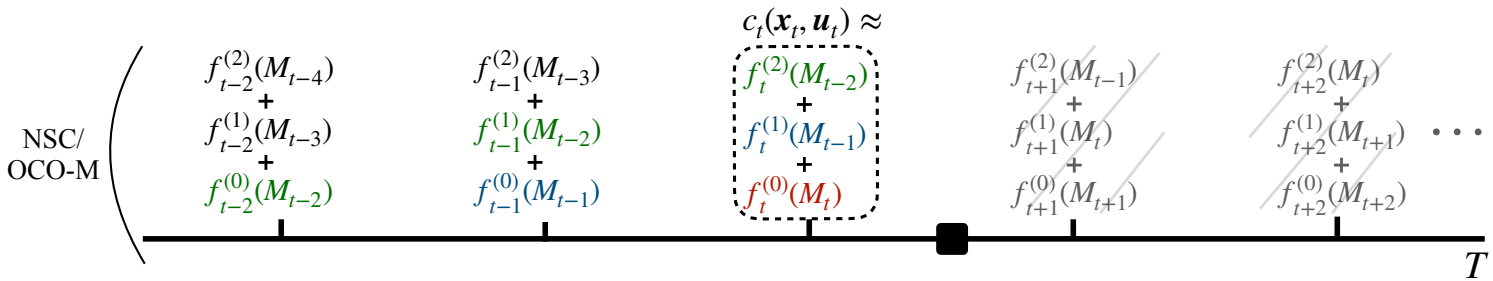
Optimistic NSC (1/3)

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable

Flaspohler et. al., Online Learning with Optimism and Delay, ICML '21

Optimistic NSC (1/3)

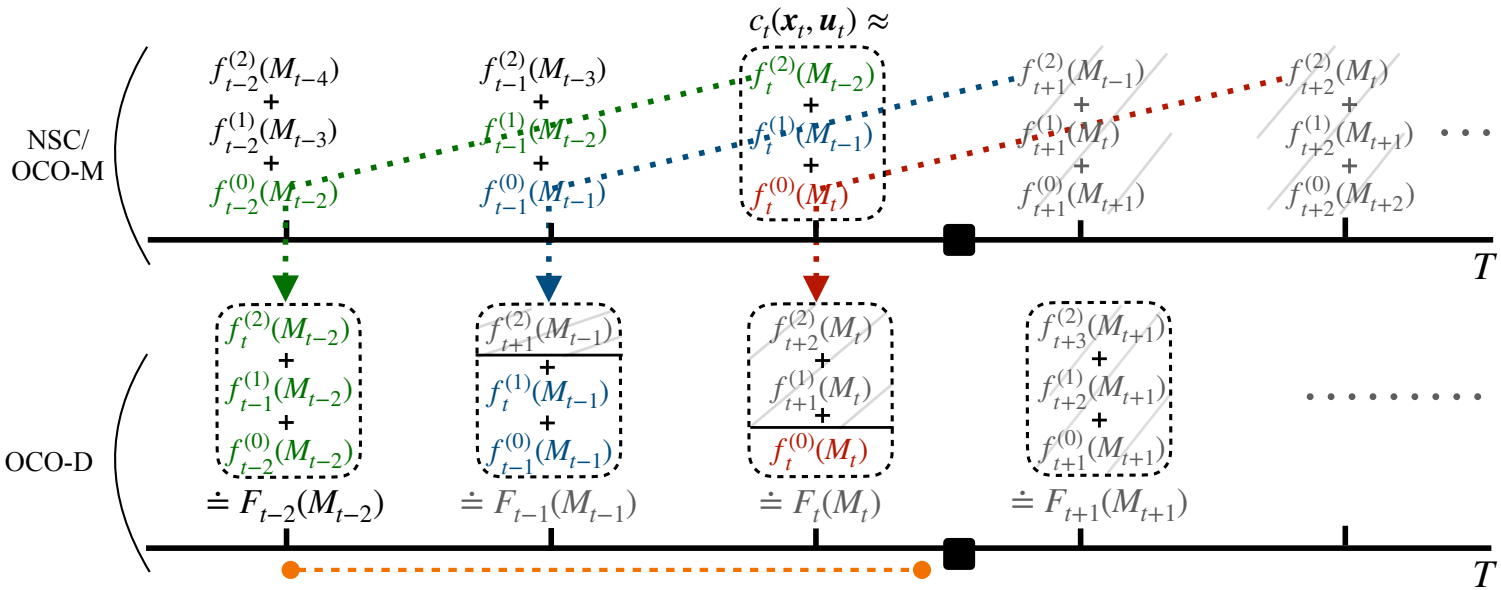
- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable



Flaspohler et. al., Online Learning with Optimism and Delay, ICML '21

Optimistic NSC (1/3)

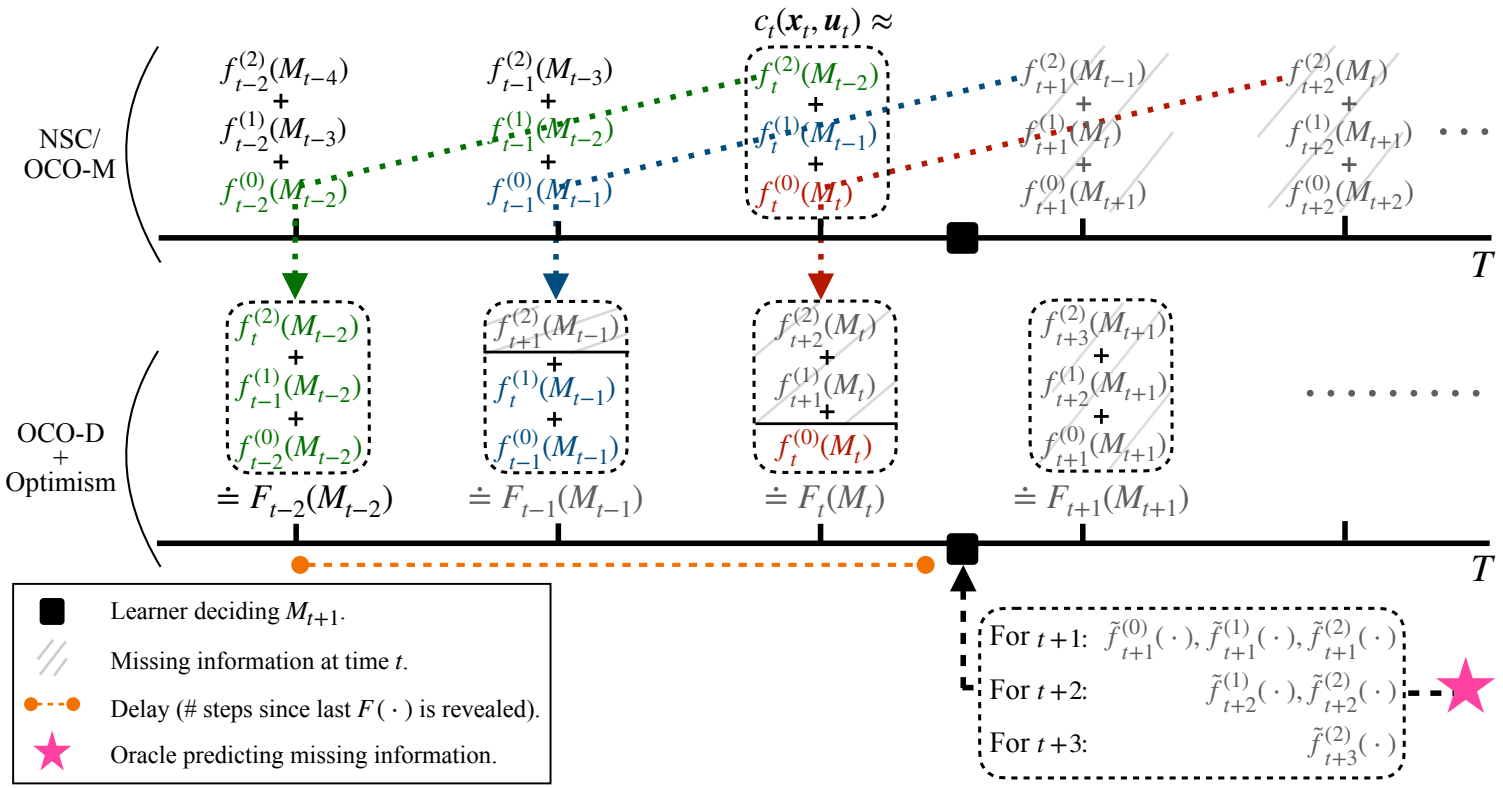
- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable



Flaspohler et. al., Online Learning with Optimism and Delay, ICML '21

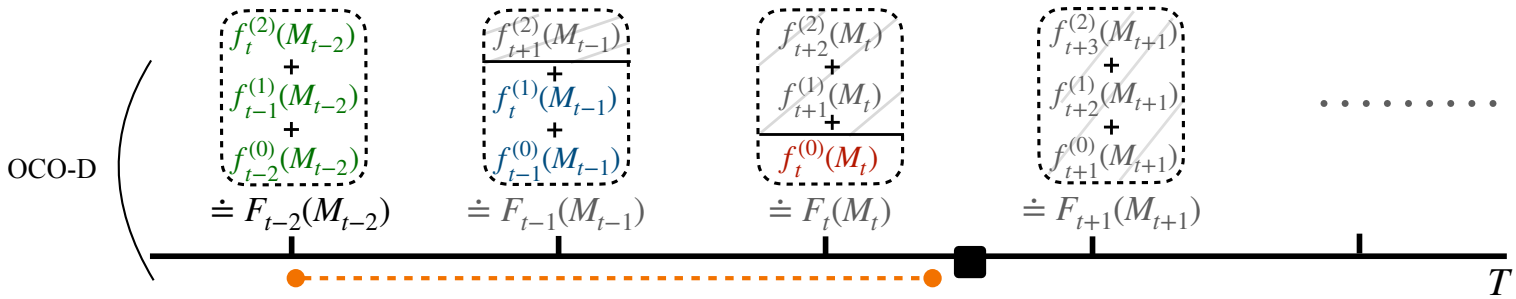
Optimistic NSC (1/3)

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable



Flaspohler et. al., Online Learning with Optimism and Delay, ICML '21

Optimistic NSC: Predictions (2/3)



Algorithm 2: `OptFTRL-C`

Input: System (A, B), parameter d, DAC parameters κ_M, p.
Output: $\{\boldsymbol{u}_t\}_{t=1}^T$.
for *each time slot* $t = 1, \ldots, T$ **do**

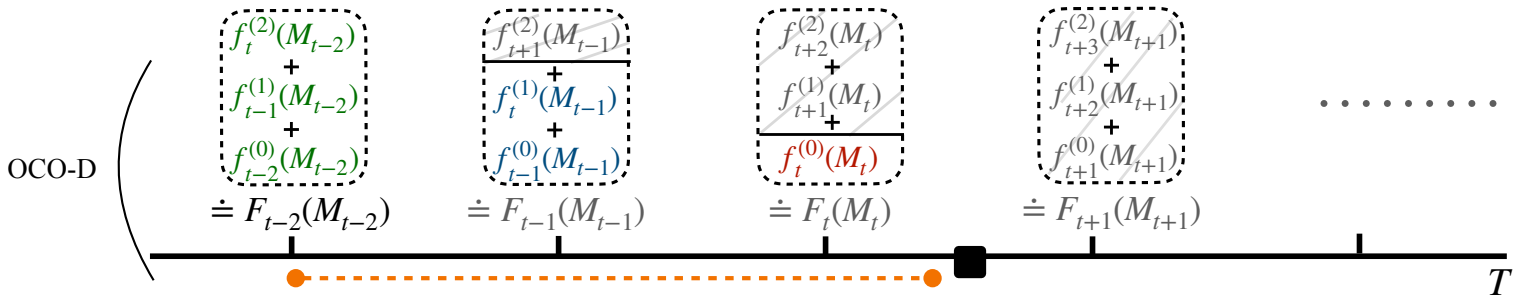
1. Use action $\boldsymbol{u}_t = \sum_{j=1}^p M_t^{[j]} \boldsymbol{w}_{t-j}$
2. Observe $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient G_{t-d}
3. Observe new state $\boldsymbol{x}_{t+1}$ and record $\boldsymbol{w}_t$
4. Update s.c. $\sigma_t \approx \sqrt{\sum_{i=0}^{t-1} \epsilon_i^2} + \max_{j \le t-d-2} \epsilon_{j-d+1:j}$
5. Receive future predictions $\{c_\tau(\cdot,\cdot)\}_{\tau=t}^{t+d}$
6. $M_{t+1} = \arg\min_{M \in \mathcal{M}} \{\langle G_{1:t-d} + H_{t+1}, M \rangle + r_t(M)\}$

end

- OptFTRL-C = Delayed OFTRL.

N. Mhaisen, G. Iosifidis, Optimistic online non-stochastic control via FTRL, IEEE CDC, 2024.

Optimistic NSC: Predictions (2/3)



Algorithm 2: `OptFTRL-C`

Input: System (A, B), parameter d, DAC parameters κ_M, p.
Output: $\{\boldsymbol{u}_t\}_{t=1}^T$.
for *each time slot* $t = 1, \ldots, T$ **do**

1. Use action $\boldsymbol{u}_t = \sum_{j=1}^p M_t^{[j]} \boldsymbol{w}_{t-j}$
2. Observe $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient G_{t-d}
3. Observe new state $\boldsymbol{x}_{t+1}$ and record $\boldsymbol{w}_t$
4. Update s.c. $\sigma_t \approx \sqrt{\sum_{i=0}^{t-1} \epsilon_i^2} + \max_{j \le t-d-2} \epsilon_{j-d+1:j}$
5. Receive future predictions $\{c_\tau(\cdot,\cdot)\}_{\tau=t}^{t+d}$
6. $M_{t+1} = \arg\min_{M \in \mathcal{M}} \{\langle G_{1:t-d} + H_{t+1}, M \rangle + r_t(M)\}$

end

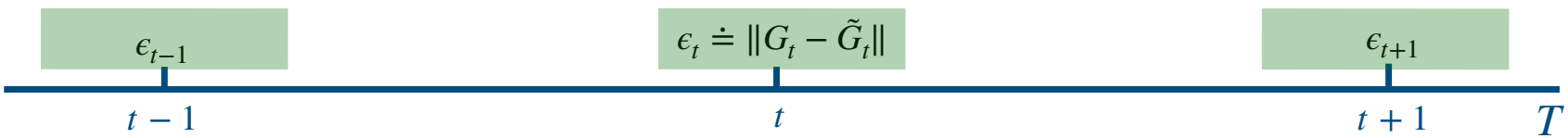
- OptFTRL-C = Delayed OFTRL.

N. Mhaisen, G. Iosifidis, Optimistic online non-stochastic control via FTRL, IEEE CDC, 2024.

Optimistic NSC: Prediction Oracle (3/3)

- Recall: Optimistic bounds accumulate the prediction errors sublinearly:

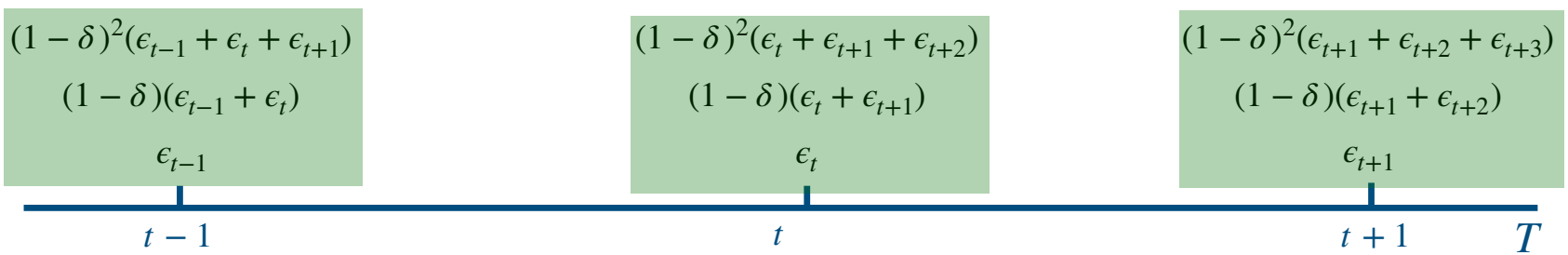
$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \epsilon_t^2}\right).$$



Optimistic NSC: Prediction Oracle (3/3)

- Optimistic Statefull bounds: accumulate the (truncated) sum of prediction errors.

$$\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T}\left(\sum_{i=0}^{d}\sum_{j=i}^{d}\epsilon_{t+i}^{(d-j+i)}\right)^2}\right).$$



Recent Advances in the Literature

Robustness and consistency tradeoff:

- Li et. al."Robustness and consistency in linear quadratic control with untrusted predictions" SIGMETRICS '23
- Li et. al. "Beyond Black-Box Advice: Learning-Augmented Algorithms for MDPs with Q-Value Predictions" NeurIPS '24

Related threads:

- Lin et. al. "Bounded-regret mpc via perturbation analysis: Prediction error, constraints, and nonlinearity" NeurIPS '22.
- Rutten et al "Smoothed Online Optimization with Unreliable Predictions" SIGMETRICS '23

Recent Advances in the Literature

Robustness and consistency tradeoff:

- Li et. al."Robustness and consistency in linear quadratic control with untrusted predictions" SIGMETRICS '23
- Li et. al. "Beyond Black-Box Advice: Learning-Augmented Algorithms for MDPs with Q-Value Predictions" NeurIPS '24

Related threads:

- Lin et. al. "Bounded-regret mpc via perturbation analysis: Prediction error, constraints, and nonlinearity" NeurIPS '22.
- Rutten et al "Smoothed Online Optimization with Unreliable Predictions" SIGMETRICS '23

Recent Advances in the Literature

Robustness and consistency tradeoff:

- Li et. al."Robustness and consistency in linear quadratic control with untrusted predictions" SIGMETRICS '23
- Li et. al. "Beyond Black-Box Advice: Learning-Augmented Algorithms for MDPs with Q-Value Predictions" NeurIPS '24

Related threads:

- Lin et. al. "Bounded-regret mpc via perturbation analysis: Prediction error, constraints, and nonlinearity" NeurIPS '22.
- Rutten et al "Smoothed Online Optimization with Unreliable Predictions" SIGMETRICS '23

More Information:

- F. Aslan, G. Iosifidis, J. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Pérez: *Fair Resource Allocation in Virtualized O-RAN Platforms*. ACM Sigmetrics, 2024.
- N. Mhaisen, G. Iosifidis: *Optimistic Online Non-stochastic Control via FTRL*. IEEE CDC, 2024.
- N. Mhaisen, G. Iosifidis: *Adaptive online non-stochastic control*. L4DC 2024.
- N. Mhaisen, G. Iosifidis, D. Leith: *Online Caching With no Regret: Optimistic Learning via Recommendations*. IEEE Trans. Mob. Comput. 23(5), 2024.
- N. Mhaisen, A. Sinha, G. S. Paschos, G. Iosifidis: *Optimistic No-regret Algorithms for Discrete Caching*. ACM Sigmetrics, 2023.
- D. Anderson, G. Iosifidis, D. J. Leith: *Lazy Lagrangians for Optimistic Learning With Budget Constraints*. IEEE/ACM Trans. Netw. 31(5), 2023.
- J. Monteil, G. Iosifidis, I. Dusparic: *Reservation of Virtualized Resources with Optimistic Online Learning*. IEEE ICC 2023.
- D. Leith, G. Iosifidis: *Penalized FTRL with Time-Varying Constraints*. ECML/PKDD, 2022.
- N. Mhaisen, G. Iosifidis, D. Leith: *Online Caching with Optimistic Learning*. IFIP Networking, 2022.
- V. Valls, G. Iosifidis, D. Leith, L. Tassiulas: *Online Convex Optimization with Perturbed Constraints: Optimal Rates against Stronger Benchmarks*. AISTATS 2020.