Optimistic Learning for Resource Management in Communication Systems

Naram Mhaisen, George Iosifidis

ACM SIGMETRICS / IFIP PERFORMANCE, Venice, 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

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

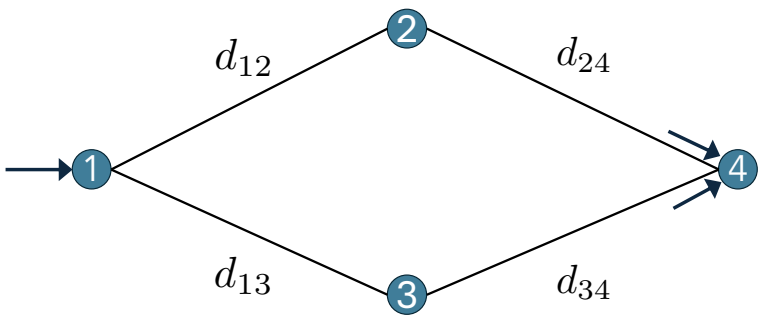
- The network controller (NC) has prior access to user demands (e.g., popularity of files) and network parameters (e.g., link delays).
- Formulates & solves an offline optimization problem.
- Applies the solution for the rest of the network operation (one-off).

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

Facets of Network Control

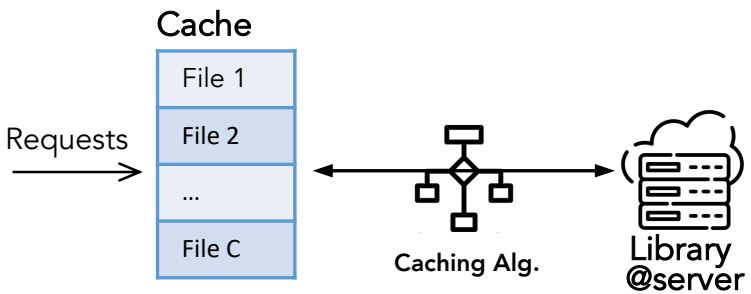
Resource Management via **Static** Optimization

- 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

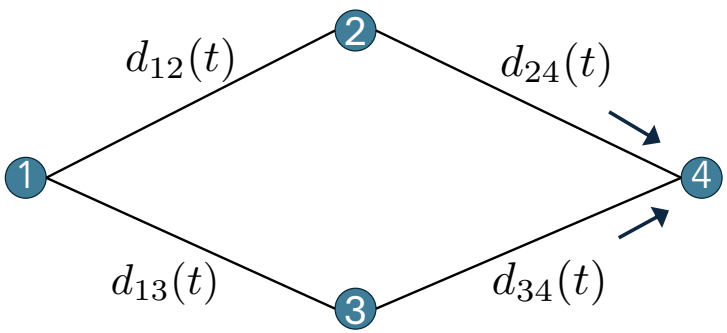
- The system operation is time-slotted.
- NC observes at the beginning of each slot the demands, environment and network state (e.g., unfinished workloads = queues).
- Solves a problem to find the best decision for the current slot.

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?

Online Convex Optimization Approach

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 learning rule (or, strategy).

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

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 learning rule (or, strategy).

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

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 learning rule (or, strategy).

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

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 learning rule (or, strategy).

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

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 learning rule (or, strategy).



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

- 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

- 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

- 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 perturbations 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 perturbations or actions
  - Fast & efficient solution
– Inaccurate under distribution shifts
  - Need data & offline training

Online Learning

+ No prior training
  - Adapts at runtime
+ Works under general conditions
  - 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.

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.

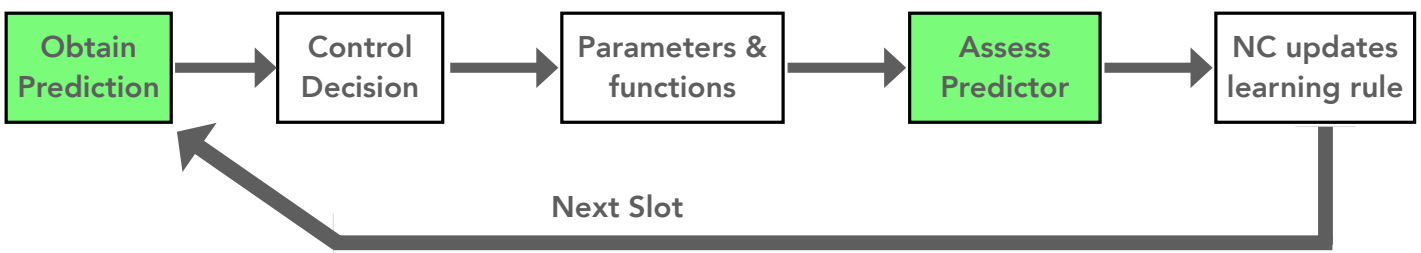


Table of Contents

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

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

Caching (Cache size C, library of N files)

1. Decide the cached files

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

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

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

Caching (Cache size C, library 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

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

Caching (Cache size C, library 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

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

Caching (Cache size C, library 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

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

Caching (Cache size C, library 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

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

Caching (Cache size C, library 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

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

Caching (Cache size C, library 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

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

Caching (Cache size C, library of N files)

1. Decide the cached files
   $$\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ [0,1]^N \;\middle|\; \textstyle\sum_n x_n \leq C \right\}$$
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 $\boldsymbol{x}_t \in \mathcal{X}$
3. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

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

Caching (Cache size C, library 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 $\boldsymbol{x}_t \in \mathcal{X}$
3. Nature reveals utility $f_t(\cdot)$
4. Learner gains $f_t(\boldsymbol{x}_t)$
5. Learner computes prediction error

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

Caching (Cache size C, library of N files)

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

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

Caching (Cache size C, library of N files)

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

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

Caching (Cache size C, library of N files)

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

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

Caching (Cache size C, library of N files)

1. Obtain request prediction $\tilde{\boldsymbol{q}}_t$
1. 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 & Results

- 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: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$

Metric, Algorithms & Results

- 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: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$

Metric, Algorithms & Results

- 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: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$

Metric, Algorithms & Results

- 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 $x^\star$ when $T \to \infty$
- Fact: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$

Metric, Algorithms & Results

- 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: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$

Metric, Algorithms & Results

- 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: $\mathcal{R}_T(\{\boldsymbol{x}_t\}_T) = \Omega(\sqrt{T})$

Metric, Algorithms & Results

- Online Gradient Descent (OGD)

$$\boldsymbol{x}_{t+1} = \mathsf{Proj}_{\mathcal{X}}\left(\boldsymbol{x}_t - \tfrac{2}{\sigma}\nabla f_t(\boldsymbol{x}_t)\right) = \arg\min_{\boldsymbol{x}\in\mathcal{X}}\left\{\langle \boldsymbol{q}_t, \boldsymbol{x}\rangle + \sigma\|\boldsymbol{x} - \boldsymbol{x}_t\|^2\right\}$$

- FTRL (*Follow The Leader + Regularization*)

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

Regularizer depends on $\mathcal{X}$, e.g., $r_t(\boldsymbol{x}) = \sigma_t\|\boldsymbol{x}\|^2$ or $\sigma_t\sum_i x_i \log x_i$

- Optimistic FTRL

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

Regularizer depends also on prediction errors, $r_t(\boldsymbol{x}) = \sigma_t(\epsilon_{1:t})\|\boldsymbol{x}\|^2$

H. B. McMahan: A Survey of Algorithms and Analysis for Adaptive Online Learning. J. Mach. Learn. Res. 18, 2017.

Metric, Algorithms & Results

- Online Gradient Descent (OGD)

$$\boldsymbol{x}_{t+1} = \mathrm{Proj}_{\mathcal{X}}\left(\boldsymbol{x}_t - \tfrac{2}{\sigma}\nabla f_t(\boldsymbol{x}_t)\right) = \arg\min_{\boldsymbol{x}\in\mathcal{X}}\left\{\langle \boldsymbol{q}_t, \boldsymbol{x}\rangle + \sigma\|\boldsymbol{x}-\boldsymbol{x}_t\|^2\right\}$$

- FTRL (*Follow The Leader + Regularization*)

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

Regularizer depends on $\mathcal{X}$, e.g., $r_t(\boldsymbol{x}) = \sigma_t\|\boldsymbol{x}\|^2$ or $\sigma_t \sum_i x_i \log x_i$

- Optimistic FTRL

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

Regularizer depends also on prediction errors, $r_t(\boldsymbol{x}) = \sigma_t(\epsilon_{1:t})\|\boldsymbol{x}\|^2$

H. B. McMahan: A Survey of Algorithms and Analysis for Adaptive Online Learning. J. Mach. Learn. Res. 18, 2017.

Metric, Algorithms & Results

- Online Gradient Descent (OGD)

$$\boldsymbol{x}_{t+1} = \mathrm{Proj}_{\mathcal{X}}\left(\boldsymbol{x}_t - \tfrac{2}{\sigma}\nabla f_t(\boldsymbol{x}_t)\right) = \arg\min_{\boldsymbol{x}\in\mathcal{X}}\left\{\langle \boldsymbol{q}_t, \boldsymbol{x}\rangle + \sigma\|\boldsymbol{x}-\boldsymbol{x}_t\|^2\right\}$$

- FTRL (*Follow The Leader + Regularization*)

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

Regularizer depends on $\mathcal{X}$, e.g., $r_t(\boldsymbol{x}) = \sigma_t\|\boldsymbol{x}\|^2$ or $\sigma_t\sum_i x_i \log x_i$

- Optimistic FTRL

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

Regularizer depends also on prediction errors, $r_t(\boldsymbol{x}) = \sigma_t(\epsilon_{1:t})\|\boldsymbol{x}\|^2$

H. B. McMahan: A Survey of Algorithms and Analysis for Adaptive Online Learning. J. Mach. Learn. Res. 18, 2017.

Metric, Algorithms & Results

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 produces decisions (e.g., caching vectors) $\{\boldsymbol{x}_t\}_T$ such that:

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

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

Metric, Algorithms & Results

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 produces decisions (e.g., caching vectors) $\{\boldsymbol{x}_t\}_T$ such that:

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

Fixed Regs

$\sigma_t \approx G\sqrt{T}, \forall t$

$\mathcal{R}_T \leq G\sqrt{T}$

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

Metric, Algorithms & Results

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 produces decisions (e.g., caching vectors) $\{\boldsymbol{x}_t\}_T$ such that:

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

Fixed Regs

$\sigma_t \approx G\sqrt{T}, \forall t$

$\mathcal{R}_T \leq G\sqrt{T}$

Adaptive Regs

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

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

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

Metric, Algorithms & Results

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 produces decisions (e.g., caching vectors) $\{\boldsymbol{x}_t\}_T$ such that:

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

Fixed Regs

$\sigma_t \approx G\sqrt{T}, \forall t$

$\mathcal{R}_T \leq G\sqrt{T}$

Adaptive Regs

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

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

Optimistic Regs

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

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

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 $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x})$
  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 $\sigma_{t+1} = \sqrt{\sum_{i=1}^{t} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|^2}$
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.

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 $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x})$
  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 $\sigma_{t+1} = \sqrt{\sum_{i=1}^{t} \|\boldsymbol{q}_i - \tilde{\boldsymbol{q}}_i\|^2}$
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.

$$\mathcal{R}_t \leq \sqrt{\#\text{ mistakes}}$$

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

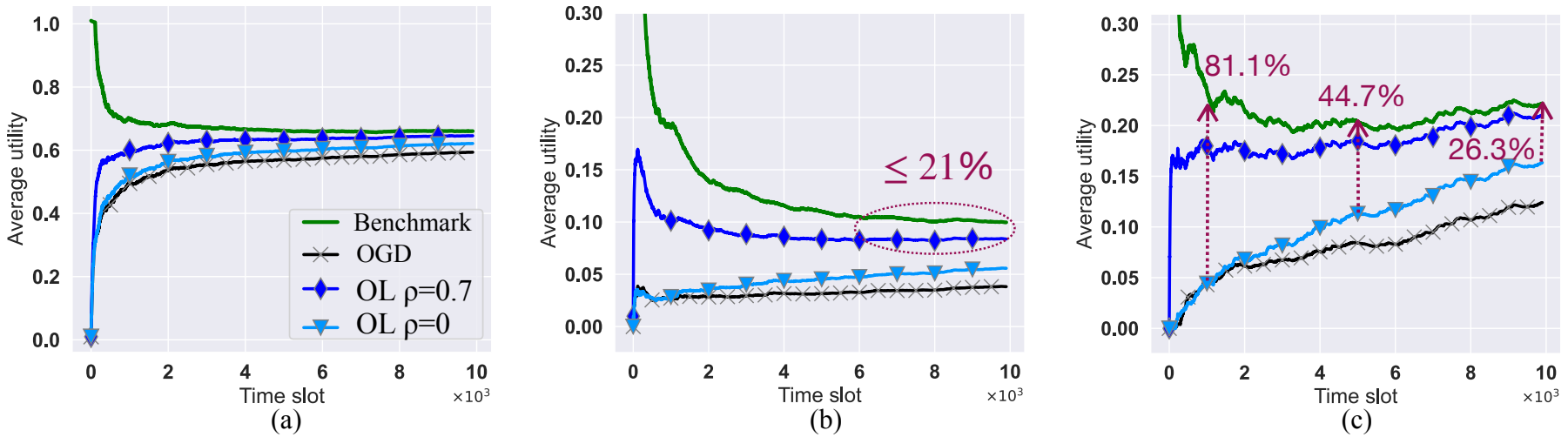


Figure: Cache utility with predictors of different accuracy ($\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_t$ with probability ρ) and request traces from: (a) Zipf distribution; (b) YouTube dataset; (c) MovieLens datasets.

Implementation

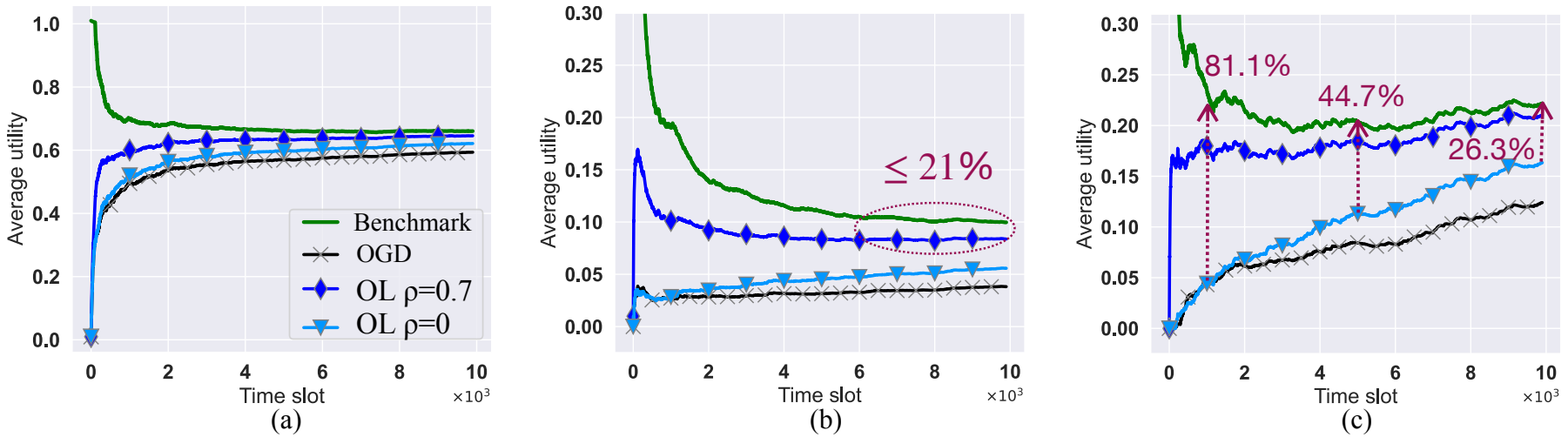


Figure: Cache utility with predictors of different accuracy ($\boldsymbol{q}_t = \tilde{\boldsymbol{q}}_t$ with probability ρ) and request traces from: (a) Zipf distribution; (b) YouTube dataset; (c) MovieLens datasets.

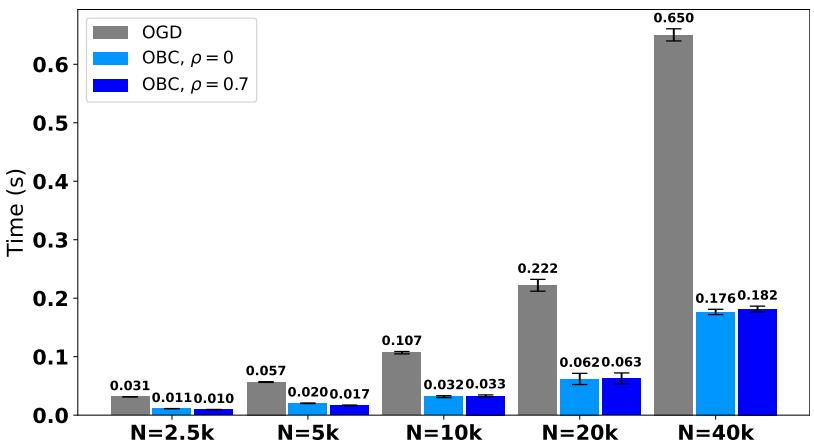


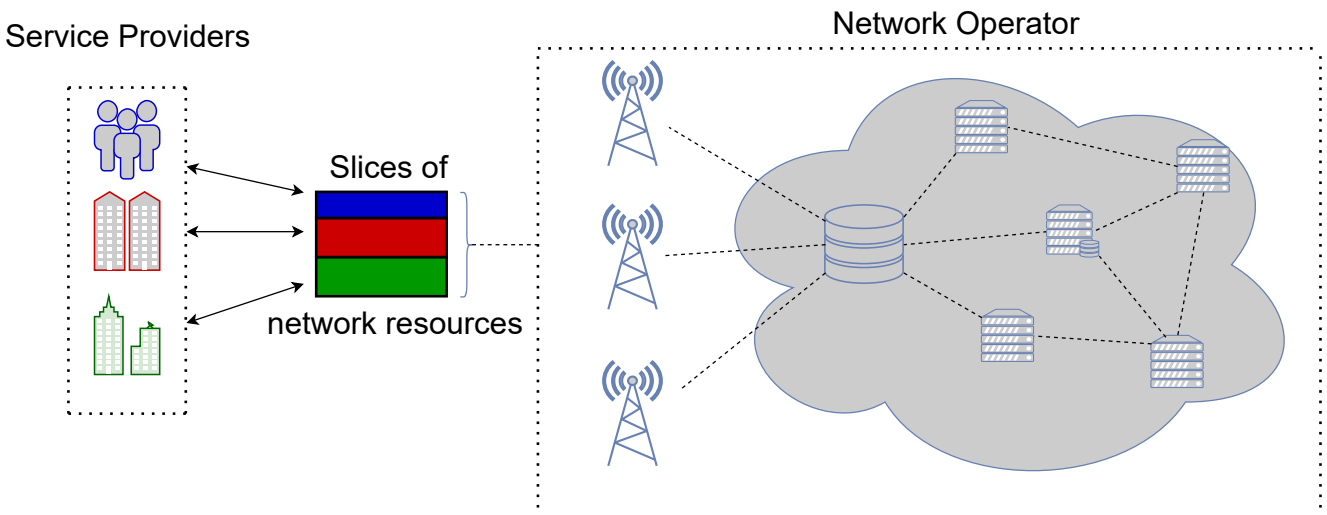
Figure: Average time consumed per decision step for different library size N and horizon $T = 10k$).

OL for: Slice Reservation

- Request virtualized resources from a network (or cloud) operator.
  - How to maximize service utility while minimizing reservation costs?

Utility: $f_t(\boldsymbol{x}) = \alpha_t \log\left(1 + \langle \boldsymbol{\theta}_t, \boldsymbol{x}_t \rangle\right)$ where:

$\boldsymbol{x}_t$: resource request plan;
$\boldsymbol{\theta}_t$: importance of each resource type for QoS
α_t: demand intensity.



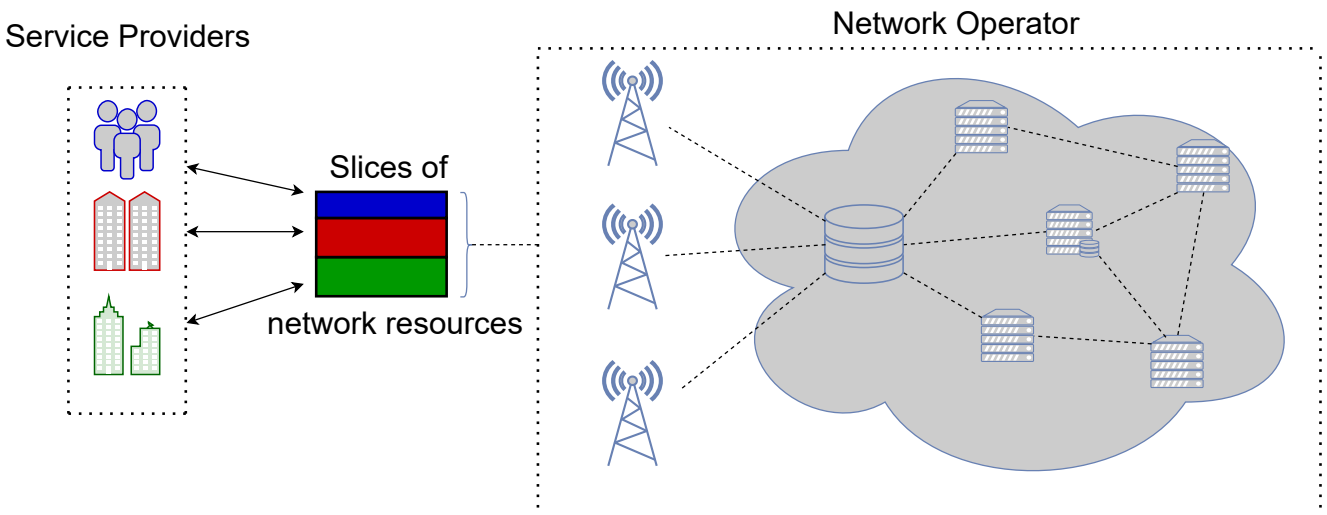
J. B. Monteil, G. Iosifidis, I. Dusparic: Reservation of Virtualized Resources with Optimistic Online Learning. IEEE ICC, 2023.

OL for: Slice Reservation

- Request virtualized resources from a network (or cloud) operator.
  - How to maximize service utility while minimizing reservation costs?

Utility: $f_t(\boldsymbol{x}) = \alpha_t \log\left(1 + \langle \boldsymbol{\theta}_t, \boldsymbol{x}_t \rangle\right)$ where:

$\boldsymbol{x}_t$: resource request plan;
$\boldsymbol{\theta}_t$: importance of each resource type for QoS
α_t: demand intensity.



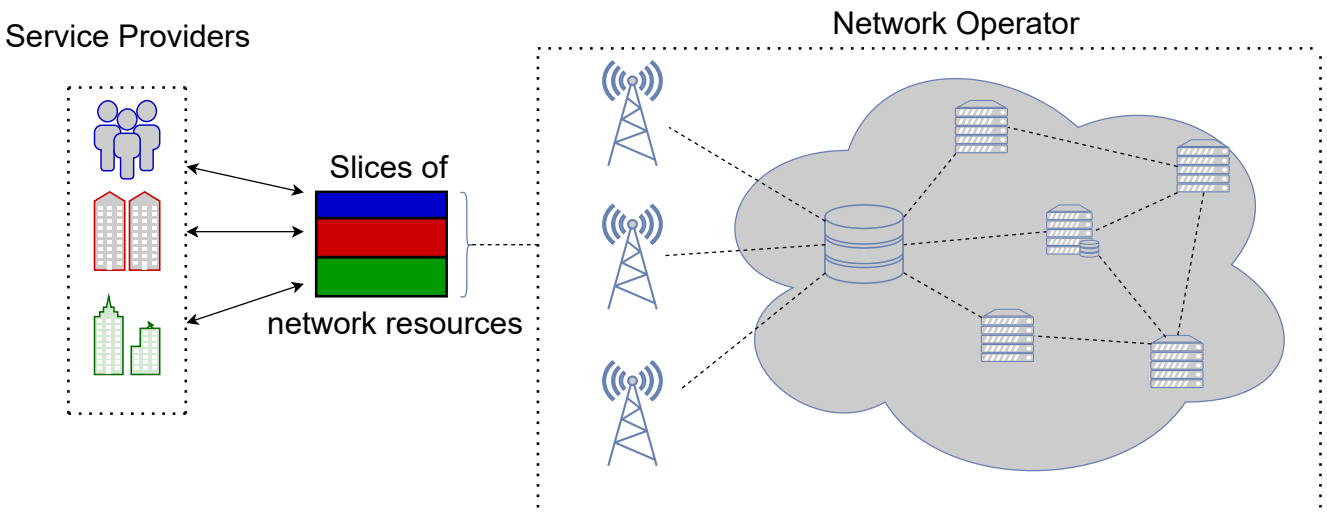
J. B. Monteil, G. Iosifidis, I. Dusparic: Reservation of Virtualized Resources with Optimistic Online Learning. IEEE ICC, 2023.

OL for: Slice Reservation

- Request virtualized resources from a network (or cloud) operator.
  - How to maximize service utility while minimizing reservation costs?

$$\text{Utility:} \quad f_t(\boldsymbol{x}) = \alpha_t \log\left(1 + \langle \boldsymbol{\theta}_t, \boldsymbol{x}_t \rangle\right) \qquad \text{where:}$$

$\boldsymbol{x}_t$: resource request plan;
$\boldsymbol{\theta}_t$: importance of each resource type for QoS
α_t: demand intensity.



J. B. Monteil, G. Iosifidis, I. Dusparic: Reservation of Virtualized Resources with Optimistic Online Learning. IEEE ICC, 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 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.

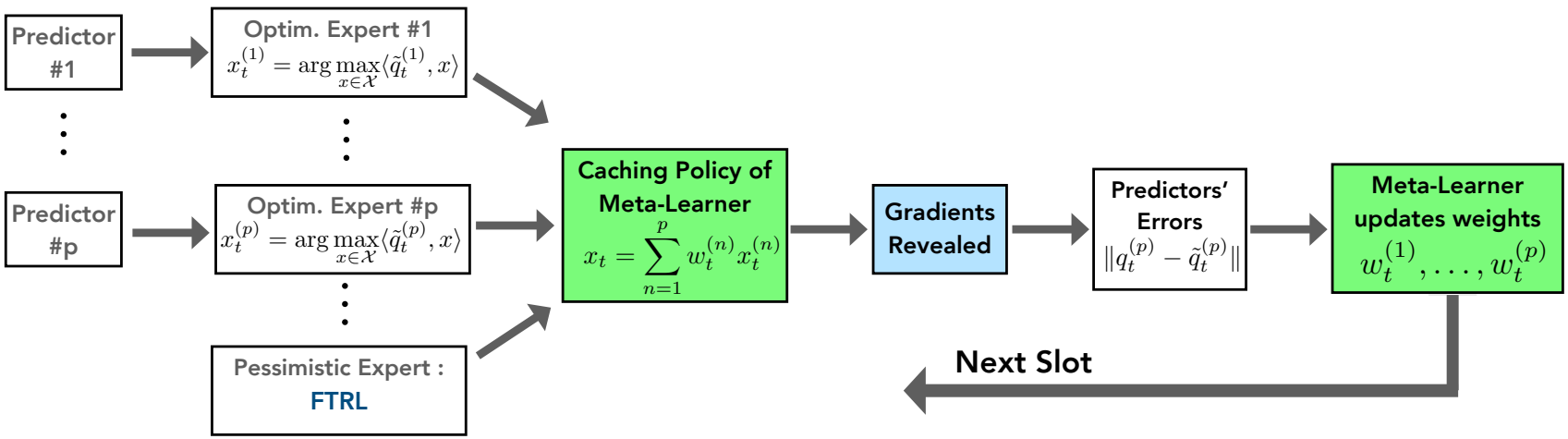
OL with Multiple Predictors

- 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

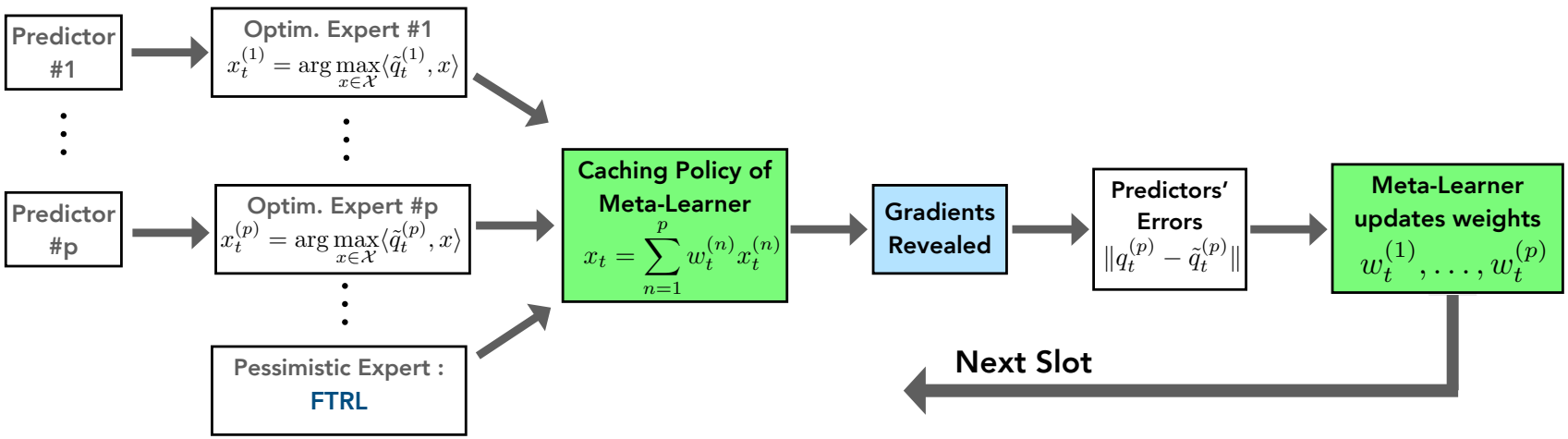
- How to incorporate **multiple** (untrusted) predictors?
  - $\hookrightarrow$ Need (also) to learn who is the best predictor from the set $\mathcal{P}$.
  - Solution: a meta-learner adapts trusting weights for each predictor.



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.

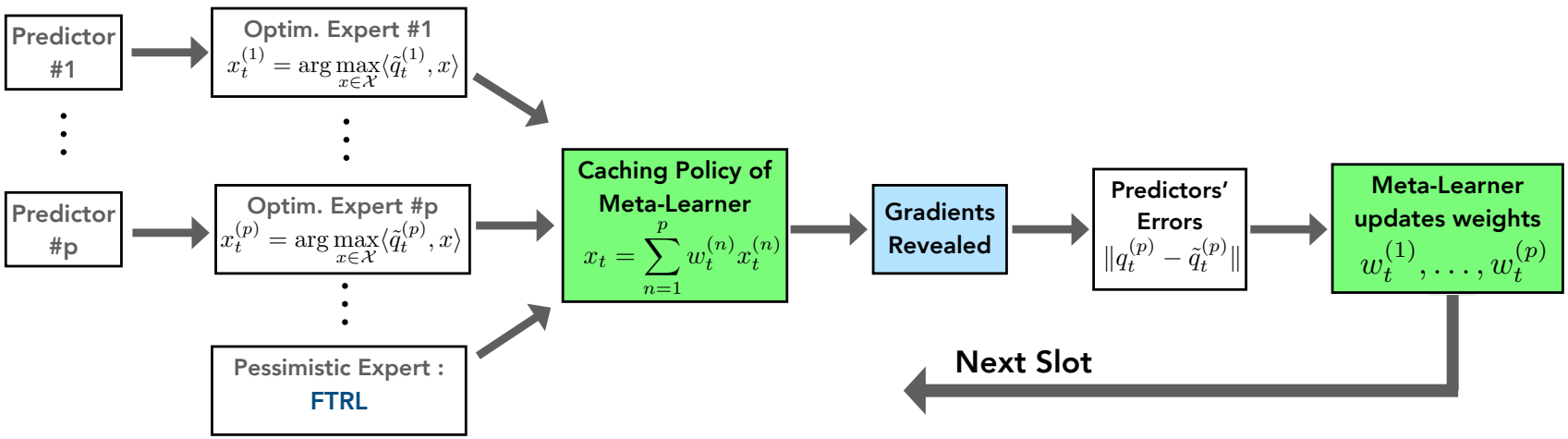


$$\mathcal{R}_T \leq \underbrace{\mathcal{R}_T^{(w)}}_{\text{find best predictor}} + \underbrace{\min_p \mathcal{R}_T^{(p)}}_{\text{regret of best predictor}}$$

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.

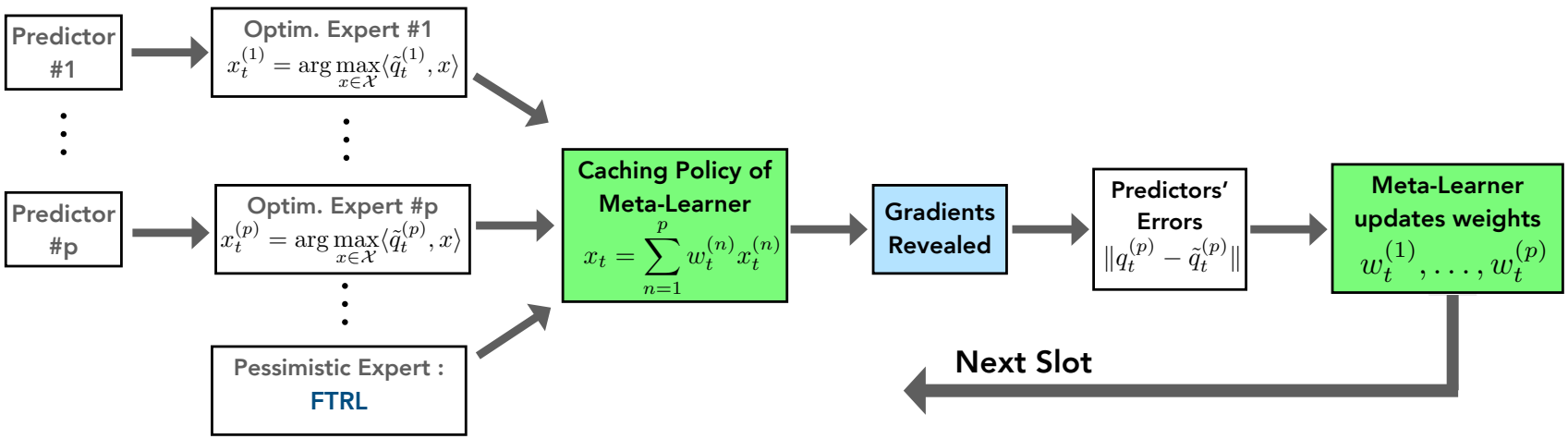


$$\mathcal{R}_T \leq \underbrace{\sqrt{P}\sqrt{T}}_{\text{find best predictor}} + \underbrace{\min_p \mathcal{R}_T^{(p)}}_{\text{max accuracy}}$$

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.



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

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:

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

  ↪ 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.

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

  - 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: Optimism and Adaptivity

- The bounds can become adaptive to scenario (better constants).
  - FTRL with gradient/prediction-adaptive time-varying regularization.

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

$\hookrightarrow$ $r_t(x)$ can be a quadratic or an entropic function, proximal or non-proximal.

- The Regret factors are as small as possible:

$$\mathcal{R}_T \leq \max_t \|\boldsymbol{x} - \boldsymbol{x}_t\| \sqrt{n \inf_{\boldsymbol{s}, \mathbf{1}^\top \boldsymbol{s} \preceq 0} \sum_{t=1}^{T} \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2_{\mathrm{diag}(s)^{-1}}}$$

$\hookrightarrow$ The previous solutions are special cases via different choices of $r_t(\boldsymbol{x})$.

[C1] Steinhardt & Liang: Adaptivity and Optimism: An Improved Exponentiated Gradient Alg. ICML'14
[C2] Mohri & Yang: Accelerating Online Convex Optimization via Adaptive Prediction. AISTATS 2016.
[C3] Joulani et al.: A Modular Analysis of Adaptive (Non-)Convex Optimization [...], ALT 2017.

Prior Work: Optimism and Adaptivity

- The bounds can become adaptive to scenario (better constants).
  - FTRL with gradient/prediction-adaptive time-varying regularization.

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

$\hookrightarrow$ $r_t(x)$ can be a quadratic or an entropic function, proximal or non-proximal.

- The Regret factors are as small as possible:

$$\mathcal{R}_T \leq \max_t \|\boldsymbol{x} - \boldsymbol{x}_t\| \sqrt{n \inf_{\boldsymbol{s}, \mathbf{1}^\top \boldsymbol{s} \preceq 0} \sum_{t=1}^{T} \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2_{\mathsf{diag}(s)^{-1}}}$$

$\hookrightarrow$ The previous solutions are special cases via different choices of $r_t(\boldsymbol{x})$.

[C1] Steinhardt & Liang: Adaptivity and Optimism: An Improved Exponentiated Gradient Alg. ICML'14
[C2] Mohri & Yang: Accelerating Online Convex Optimization via Adaptive Prediction. AISTATS 2016.
[C3] Joulani et al.: A Modular Analysis of Adaptive (Non-)Convex Optimization [...], ALT 2017.

Prior Work: Different Prediction Metrics

- We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|^2$
- What if we know a coordinate, e.g., $g_{t+1,1}$, of the next gradient $\boldsymbol{g}_{t+1} \in \mathbb{R}^n$?

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

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

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

$\hookrightarrow$ The bound relies on the geometry of the decision space (strong convexity).

[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.

Prior Work: Different Prediction Metrics

- We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|^2$
- What if we know a coordinate, e.g., $g_{t+1,1}$, of the next gradient $\boldsymbol{g}_{t+1} \in \mathbb{R}^n$?

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

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

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

↪ The bound relies on the geometry of the decision space (strong convexity).

[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.

Prior Work: Different Prediction Metrics

- We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|^2$
- What if we know a coordinate, e.g., $g_{t+1,1}$, of the next gradient $\boldsymbol{g}_{t+1} \in \mathbb{R}^n$?

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

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

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

↪ The bound relies on the geometry of the decision space (strong convexity).

[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.

Prior Work: Different Prediction Metrics

- We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|^2$
- What if we know a coordinate, e.g., $g_{t+1,1}$, of the next gradient $\boldsymbol{g}_{t+1} \in \mathbb{R}^n$?

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

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

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

$\hookrightarrow$ The bound relies on the geometry of the decision space (strong convexity).

[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.

Prior Work: Different Prediction Metrics

- We used approximate predictions for next gradients, i.e., $\epsilon_t = \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|^2$
- What if we know a coordinate, e.g., $g_{t+1,1}$, of the next gradient $\boldsymbol{g}_{t+1} \in \mathbb{R}^n$?

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

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

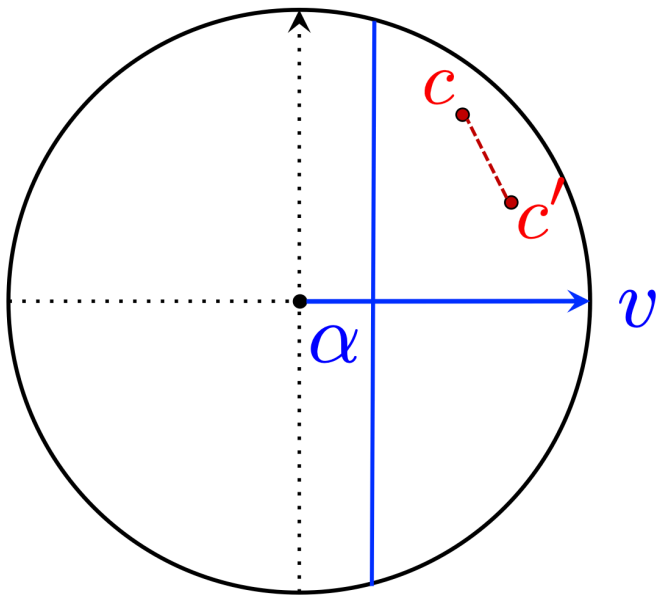
$$\langle \tilde{\boldsymbol{g}}_t, \boldsymbol{g}_t \rangle \geq \alpha \|\boldsymbol{g}_t\| \quad \rightarrow \quad \mathcal{R}_T \leq \mathcal{O}\left(\frac{1}{\alpha} \log T\right)$$

$\hookrightarrow$ The bound relies on the geometry of the decision space (strong convexity).

[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.

Prior Work: Different Prediction Metrics

- Approximate? $\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 \leq \epsilon$ or Directional? $\langle \boldsymbol{u}_t, \boldsymbol{q}_t \rangle \geq \alpha \|\boldsymbol{q}_t\|$



[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.

Prior Work: Different Prediction Metrics

- Approximate or Directional?

$$\mathcal{O}\left(\sqrt{\sum_t \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2}\right) \quad \text{vs} \quad \mathcal{O}\left(\log(T)\sqrt{\max\left(1, \sum_t \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 - \|\tilde{\boldsymbol{q}}_t\|^2\right)}\right)$$

- Improved analysis of approximate errors in [D4], achieves: $\mathcal{R}_T \leq$

$$\sigma_T\|\boldsymbol{x}^\star\| + \sum_{t=1}^{T} \min\left\{\frac{1}{\sigma_t}\text{huber}\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big), D_{\mathcal{X}} \min\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big)\right\}$$

$$\text{with} \quad \text{huber}\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big) = \frac{1}{2}\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 - \frac{1}{2}\big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\| - \|\boldsymbol{q}_t\|\big)_+^2$$

[D4] G. Flaspohler, et al: Online Learning with Optimism and Delay. ICML 2021.

Prior Work: Different Prediction Metrics

- Approximate or Directional?

$$\mathcal{O}\left(\sqrt{\sum_t \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2}\right) \quad \text{vs} \quad \mathcal{O}\left(\log(T)\sqrt{\max\left(1, \sum_t \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 - \|\tilde{\boldsymbol{q}}_t\|^2\right)}\right)$$

- Improved analysis of approximate errors in [D4], achieves: $\mathcal{R}_T \leq$

$$\sigma_T\|\boldsymbol{x}^\star\| + \sum_{t=1}^{T} \min\left\{\frac{1}{\sigma_t}\text{huber}\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big), D_{\mathcal{X}} \min\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big)\right\}$$

$$\text{with} \quad \text{huber}\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big) = \frac{1}{2}\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 - \frac{1}{2}\big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\| - \|\boldsymbol{q}_t\|\big)_+^2$$

[D4] G. Flaspohler, et al: Online Learning with Optimism and Delay. ICML 2021.

Prior Work: Different Prediction Metrics

- Approximate or Directional?

$$\mathcal{O}\left(\sqrt{\sum_t \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2}\right) \quad \text{vs} \quad \mathcal{O}\left(\log(T)\sqrt{\max\left(1, \sum_t \|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 - \|\tilde{\boldsymbol{q}}_t\|^2\right)}\right)$$

- Improved analysis of approximate errors in [D4], achieves: $\mathcal{R}_T \leq$

$$\sigma_T\|\boldsymbol{x}^\star\| + \sum_{t=1}^{T} \min\left\{\frac{1}{\sigma_t}\text{huber}\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big), D_{\mathcal{X}} \min\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big)\right\}$$

$$\text{with} \quad \text{huber}\Big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|, \|\boldsymbol{q}_t\|\Big) = \frac{1}{2}\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|^2 - \frac{1}{2}\big(\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\| - \|\boldsymbol{q}_t\|\big)_+^2$$

[D4] G. Flaspohler, et al: Online Learning with Optimism and Delay. ICML 2021.

Prior Work: Different Prediction Metrics

Error	Approx. Predictor	Direct. Predictor	Improved Appr. Pred.
worst-case	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\log T)$	$\mathcal{O}(\sqrt{T})$
$\tilde{\boldsymbol{q}}_t = \boldsymbol{q}_t$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
$\tilde{\boldsymbol{q}}_t = (1/3)\boldsymbol{q}_t$	$\mathcal{O}\left(\sqrt{\frac{4}{9}\sum_t \|\boldsymbol{q}_t\|^2}\right)$	$\mathcal{O}\left(\sqrt{\frac{3}{9}\sum_t \|\boldsymbol{q}_t\|^2}\right)$	$\mathcal{O}\left(\sqrt{\frac{2}{9}\sum_t \|\boldsymbol{q}_t\|^2}\right)$
$\|\tilde{\boldsymbol{q}}_t - \boldsymbol{q}_t\| \le \rho\|\boldsymbol{q}_t\|$	$\mathcal{O}\left(\sqrt{\sum_t \rho\|\boldsymbol{q}_t\|^2}\right)$	$\mathcal{O}\left(\frac{1}{1-\rho}\log T\right)$	??

Prior Work: Beyond Regret

- Parallel World: Online algorithms with Predictions.

- Competitive Ratio instead of Regret:
  - Compare against the offline (possibly) time-varying benchmark;
  - Penalized for switching costs;
  - Observe the function before committing the action.

- Inherent incompatibility among the two metrics.
  - Still, in some cases one can achieve both under some assumptions.

[E1] L. Andrew, et al: A Tale of 2 Metrics: Simultaneous Bounds on Competitiveness & Regret, COLT'13
[E2] D. Rutten, et al., Smoothed Online Optimization with Untrusted Predictions, Sigmetrics'23.

Prior Work: Beyond Regret

- Parallel World: Online algorithms with Predictions.

- Competitive Ratio instead of Regret:
  - Compare against the offline (possibly) time-varying benchmark;
  - Penalized for switching costs;
  - Observe the function before committing the action.

- Inherent incompatibility among the two metrics.
  - Still, in some cases one can achieve both under some assumptions.

[E1] L. Andrew, et al: A Tale of 2 Metrics: Simultaneous Bounds on Competitiveness & Regret, COLT'13
[E2] D. Rutten, et al., Smoothed Online Optimization with Untrusted Predictions, Sigmetrics'23.

Prior Work: Beyond Regret

- Parallel World: Online algorithms with Predictions.

- Competitive Ratio instead of Regret:
  - Compare against the offline (possibly) time-varying benchmark;
  - Penalized for switching costs;
  - Observe the function before committing the action.

- Inherent incompatibility among the two metrics.
  - Still, in some cases one can achieve both under some assumptions.

[E1] L. Andrew, et al: A Tale of 2 Metrics: Simultaneous Bounds on Competitiveness & Regret, COLT'13
[E2] D. Rutten, et al., Smoothed Online Optimization with Untrusted Predictions, Sigmetrics'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.

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 $\hat{\boldsymbol{x}}_t \in \hat{\mathcal{X}} \doteq \left\{ [0,1]^N \;\middle|\; \sum_n y_n \leq C \right\}$.~~
- Commit to caching policy $\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ \{0,1\}^N \;\middle|\; \sum_n y_n \leq C \right\}$.
- 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 $\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|$

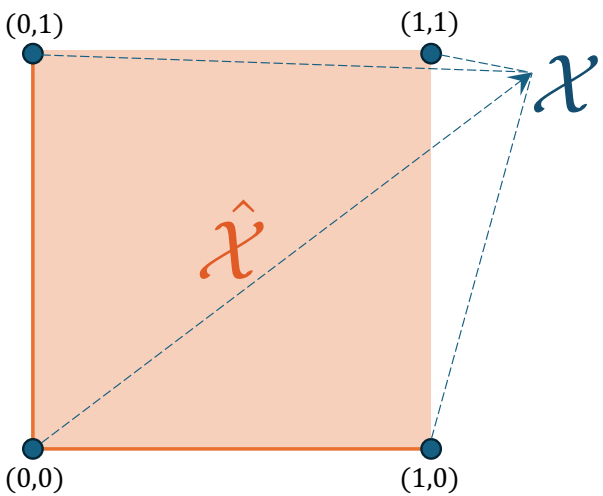
$$\boldsymbol{x}^\star = \arg\max_{x \in \mathcal{X}} \langle \boldsymbol{q}_{1:T}, \boldsymbol{x} \rangle. \text{ (Already Discrete)}$$

Optimistic Discrete Caching

OL for **Discrete** caching

- Obtain request prediction $\tilde{\boldsymbol{q}}_t$
- ~~Commit to caching policy $\hat{\boldsymbol{x}}_t \in \hat{\mathcal{X}} \doteq \left\{ [0,1]^N \;\middle|\; \sum_n y_n \leq C \right\}$.~~
- Commit to caching policy $\boldsymbol{x}_t \in \mathcal{X} \doteq \left\{ \{0,1\}^N \;\middle|\; \sum_n y_n \leq C \right\}$.
- 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 $\|\boldsymbol{q}_t - \tilde{\boldsymbol{q}}_t\|$

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



OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.

    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?

OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.

    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?

OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.

    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?

OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.
    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?

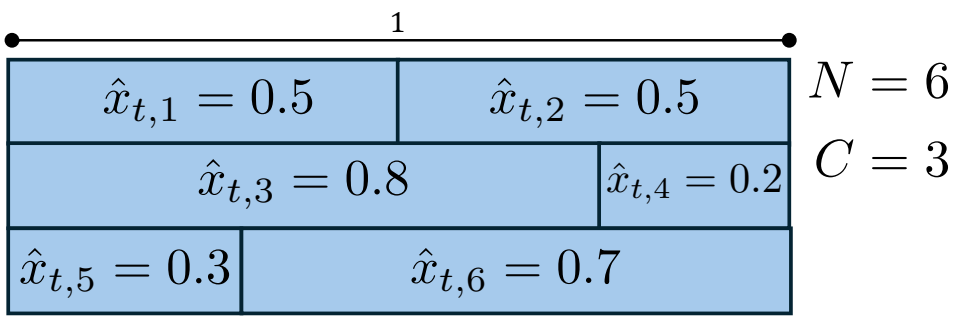
OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.
    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?



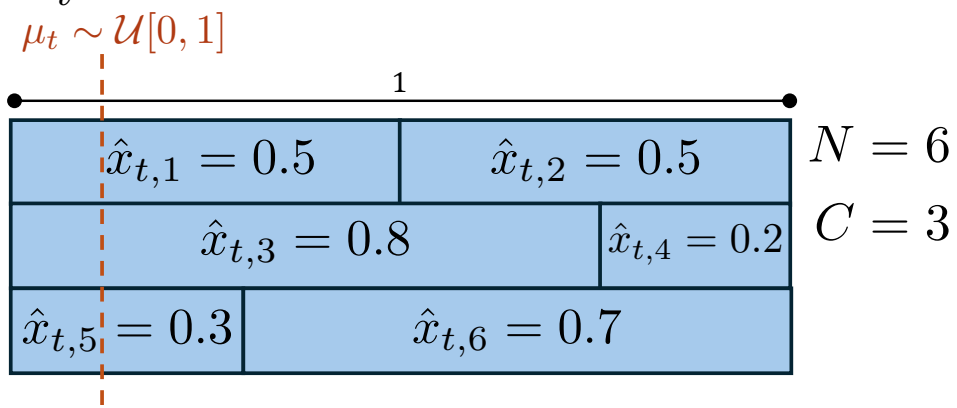
OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.
    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?



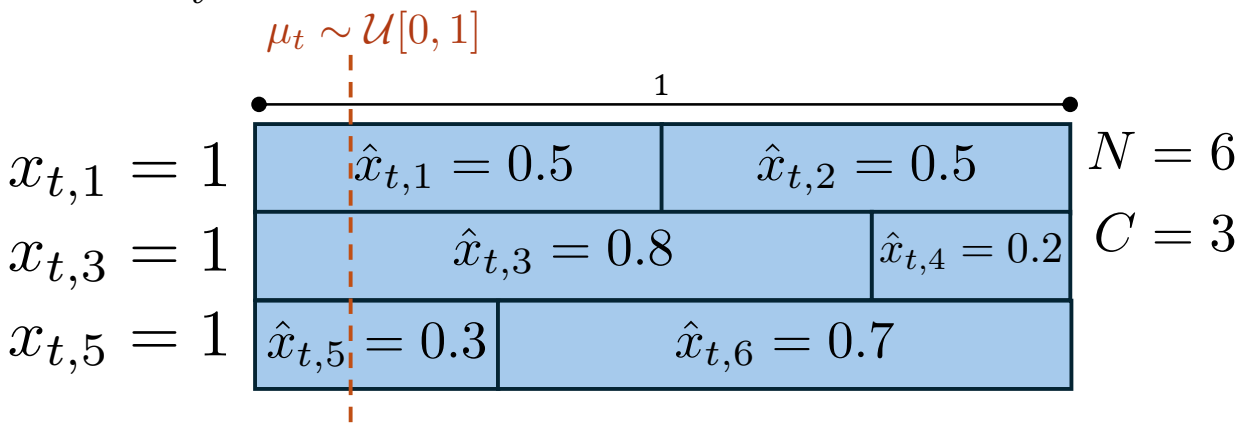
OFTRL + Sampling

- Assume there is an $\mathcal{A}$ that produces $\{\boldsymbol{x}_t\}_T$ s.t.
  - $\mathbb{E}[\boldsymbol{x}_t] = \hat{\boldsymbol{x}}_t \in \text{conv}\{\mathcal{X}\}$, and
  - $\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T) \leq \beta$

  Then
  - The discrete points obtain the same bound $\mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_T)] \leq \beta$

  Reason
  - $f_t(\hat{\boldsymbol{x}}_t) = \langle \boldsymbol{q}_t, \hat{\boldsymbol{x}}_t \rangle = \langle \boldsymbol{q}_t, \mathbb{E}[\boldsymbol{x}]_t \rangle = \mathbb{E}[\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle]$.
    Hence, $\mathcal{R}(\{\boldsymbol{x}_t\}_{t=1}^T) = \mathbb{E}[\mathcal{R}(\{\hat{\boldsymbol{x}}_t\}_{t=1}^T)]$

How to obtain such $\boldsymbol{x}_t$?



OFTRL for Discrete Actions

Algorithm 1: Optimistic DiscreteCaching

Input: ; $x_1 \in \mathcal{X}$; $\sigma = 2/D_{\mathcal{X}}$. **Output**: $x_t = (y_t, z_t)$, $\forall t$.
for $t = 1, 2, \ldots$ **do**
  Receive prediction $\tilde{\boldsymbol{q}}_{t+1}$
  Calculate the new policy $\hat{\boldsymbol{x}}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x})$
  $\boldsymbol{x}_t \leftarrow$ *MadowSample*$(\hat{\boldsymbol{x}}_t)$
  Serve request $\boldsymbol{q}_t$ according to configuration $\boldsymbol{x}_t$
  Incur a hit/miss $\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$, observe prediction error
  Update the stability parameter $\lambda_t + 1 = \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q_t} - \tilde{\boldsymbol{q}}_{\boldsymbol{t}}\|^2}$
end

- $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{q_t} - \tilde{\boldsymbol{q}}_{\boldsymbol{t}}\|^2})$. Sub-linear& scales with accuracy.

Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTRL for Discrete Actions

Algorithm 1: Optimistic DiscreteCaching

Input: ; $x_1 \in \mathcal{X}$; $\sigma = 2/D_{\mathcal{X}}$. **Output**: $x_t = (y_t, z_t)$, $\forall t$.
for $t = 1, 2, \ldots$ **do**
  Receive prediction $\tilde{\boldsymbol{q}}_{t+1}$
  Calculate the new policy $\hat{\boldsymbol{x}}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle + r_t(\boldsymbol{x})$
  $\boldsymbol{x}_t \leftarrow$ *MadowSample*$(\hat{\boldsymbol{x}}_t)$
  Serve request $\boldsymbol{q}_t$ according to configuration $\boldsymbol{x}_t$
  Incur a hit/miss $\langle \boldsymbol{q}_t, \boldsymbol{x}_t \rangle$, observe prediction error
  Update the stability parameter $\lambda_t + 1 = \sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{q_t} - \tilde{\boldsymbol{q}}_{\boldsymbol{t}}\|^2}$
end

- $\mathbb{E}[\mathcal{R}_T] = \mathcal{O}(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{q_t} - \tilde{\boldsymbol{q}}_{\boldsymbol{t}}\|^2})$. Sub-linear& scales with accuracy.

Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL for Discrete Actions

- FTRL update requires N^2 projection.
- Another way to regularize the FTL update $\boldsymbol{x}_{t+1} = \arg\min_{x \in \mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x} \rangle$
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x \in \mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t \gamma_t, \boldsymbol{x} \rangle$, $\gamma_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Quantile finding problem with complexity N

FTPL Regret Bound

The above iterates achieve the regret bound:

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

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

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15
Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL for Discrete Actions

- FTRL update requires N^2 projection.
- Another way to regularize the FTL update $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x}\rangle$
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t\gamma_t, \boldsymbol{x}\rangle$, $\gamma_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Quantile finding problem with complexity N

FTPL Regret Bound

The above iterates achieve the regret bound:

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

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

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15
Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL for Discrete Actions

- FTRL update requires N^2 projection.
- Another way to regularize the FTL update $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x}\rangle$
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t\gamma_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Quantile finding problem with complexity N

FTPL Regret Bound

The above iterates achieve the regret bound:

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

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

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15
Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL for Discrete Actions

- FTRL update requires N^2 projection.
- Another way to regularize the FTL update $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x}\rangle$
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t\gamma_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Quantile finding problem with complexity N

FTPL Regret Bound

The above iterates achieve the regret bound:

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

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

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15
Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL for Discrete Actions

- FTRL update requires N^2 projection.
- Another way to regularize the FTL update $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x}\rangle$
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t\gamma_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Quantile finding problem with complexity N

FTPL Regret Bound

The above iterates achieve the regret bound:

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

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

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15
Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL for Discrete Actions

- FTRL update requires N^2 projection.
- Another way to regularize the FTL update $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t}, \boldsymbol{x}\rangle$
- FT**P**erturbed L update: $\boldsymbol{x}_{t+1} = \arg\min_{x\in\mathcal{X}} \langle \boldsymbol{q}_{1:t} + \eta_t\gamma_t, \boldsymbol{x}\rangle$, $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.
- Quantile finding problem with complexity N

FTPL Regret Bound

The above iterates achieve the regret bound:

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

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

Cohen et. al. Following the Perturbed Leader for Online Structured Learning, ICML '15
Mhaisen et. al. Optimistic No-regret Algorithms for Discrete Caching, SIGMETRICS '22

OFTPL Alg. and Comparison with OFTRL

Algorithm 2: (`OFTPL-Cache`)

Input: $\eta_1 = 0$, $y_1 = \arg\min_{y \in \mathcal{X}} \langle y, \theta_1 \rangle$
Output: $\{y_t \in \mathcal{X}\}_T$ // *Feasible discrete caching vector at each slot*
$\gamma \overset{iid}{\sim} \mathcal{N}(0, \mathbf{1}_{N \times 1})$ // *Sample a perturbation vector*
for $t = 2, 3, \ldots$ **do**
  $\tilde{\theta}_t \leftarrow$ `prediction` //*Obtain request prediction for slot* t
  $\eta_t \approx \sqrt{\sum_{\tau=1}^{t-1} \|\theta_\tau - \tilde{\theta}_\tau\|_1^2}$
  $y_t = \arg\max_{y \in \mathcal{X}} \langle y, \Theta_{t-1} + \tilde{\theta}_t + \eta_t \gamma \rangle$ // *Update the discrete cache*
end

	OFTRL	**OFTPL**
Operation principle	Min. (regularized) past costs	Min. (perturbed) past costs
Comput. Complex.	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$
Dependence on N	None	$log(N)$
Prediction Error N	ℓ_2	ℓ_1
Solution pipeline	Almost disc. → Discrete	$Cont.$ → Almost disc. → Discrete

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

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

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

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

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

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

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

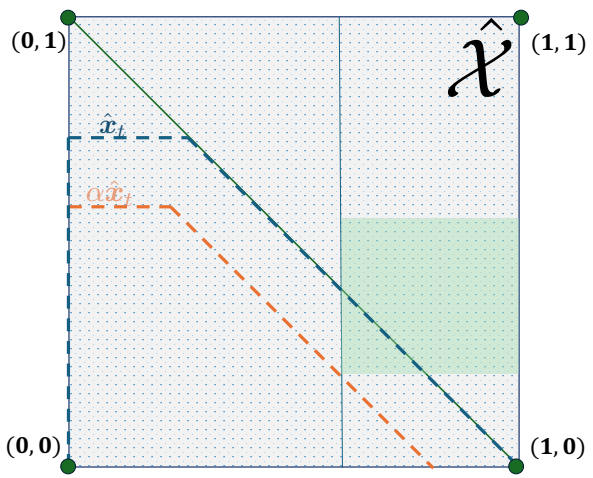
$$\underbrace{\mathcal{R}_T(\{\hat{\boldsymbol{x}}_t\}_T)}_{\text{Possibly Optimistic}} \leq A \rightarrow \mathbb{E}[\mathcal{R}_T^{\alpha}(\{\boldsymbol{x}_t\}_T)] \leq A$$

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

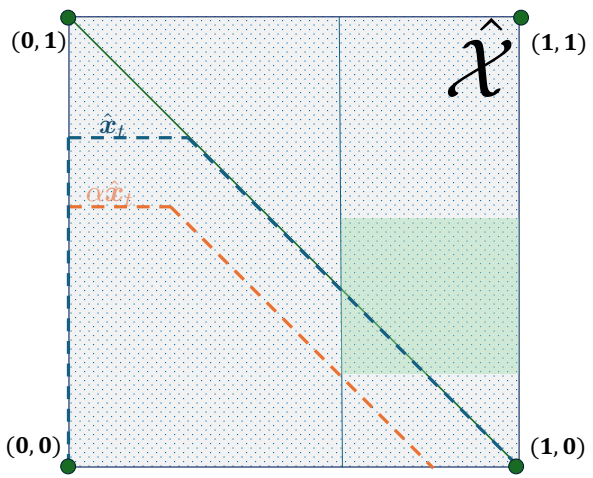
$$\underbrace{\mathcal{R}_T(\{\hat{\boldsymbol{x}}_t\}_T)}_{\text{Possibly Optimistic}} \leq A \rightarrow \mathbb{E}[\mathcal{R}_T^{\alpha}(\{\boldsymbol{x}_t\}_T)] \leq A$$

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

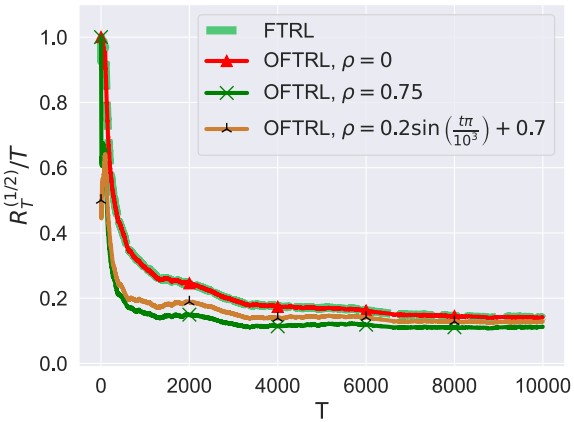
- Approximate benchmark: $\mathcal{R}_T^{\alpha}(\{\boldsymbol{x}_t\}_T) = \alpha \langle \boldsymbol{q}_{1:T}, \boldsymbol{x}^\star \rangle - \langle \boldsymbol{q}_{1:T}, \boldsymbol{x}_t \rangle$, $\alpha < 1$
- Approximate rounding: $\mathbb{E}[\boldsymbol{x}_t] \geq \alpha \hat{\boldsymbol{x}}_t$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



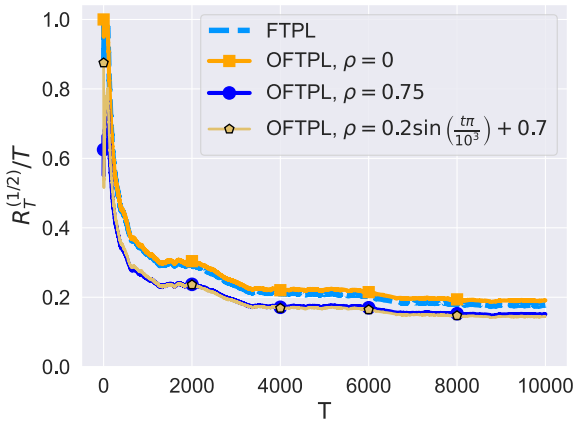
Disc. to Cont. Reduction

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

Implementation



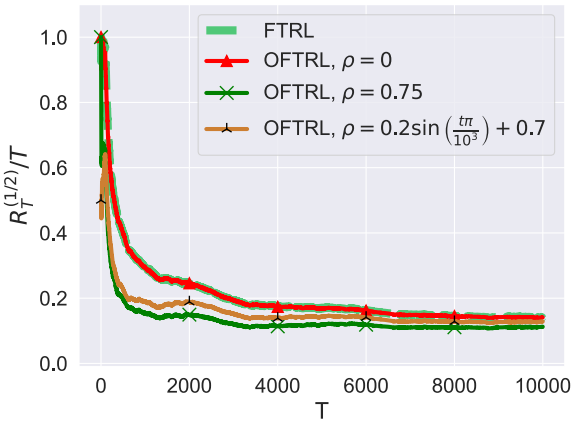
(a) `OFTRL-UneqCache`



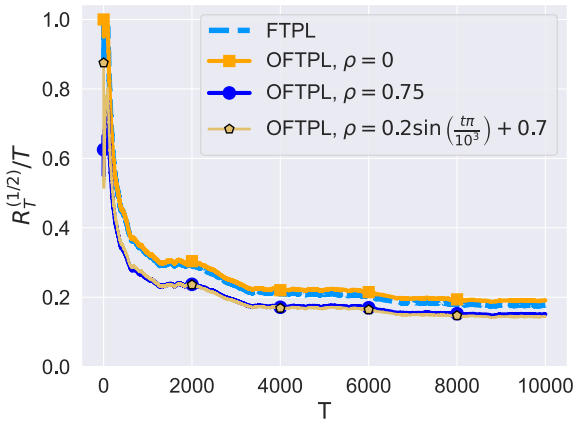
(b) `OFTPL-UneqCache`

Figure: Comparing $R_T^{(1/2)}/T$ for **unequal-sized** files, **single cache**.

Implementation



(a) `OFTRL-UneqCache`



(b) `OFTPL-UneqCache`

Figure: Comparing $R_T^{(1/2)}/T$ for **unequal-sized** files, **single cache**.

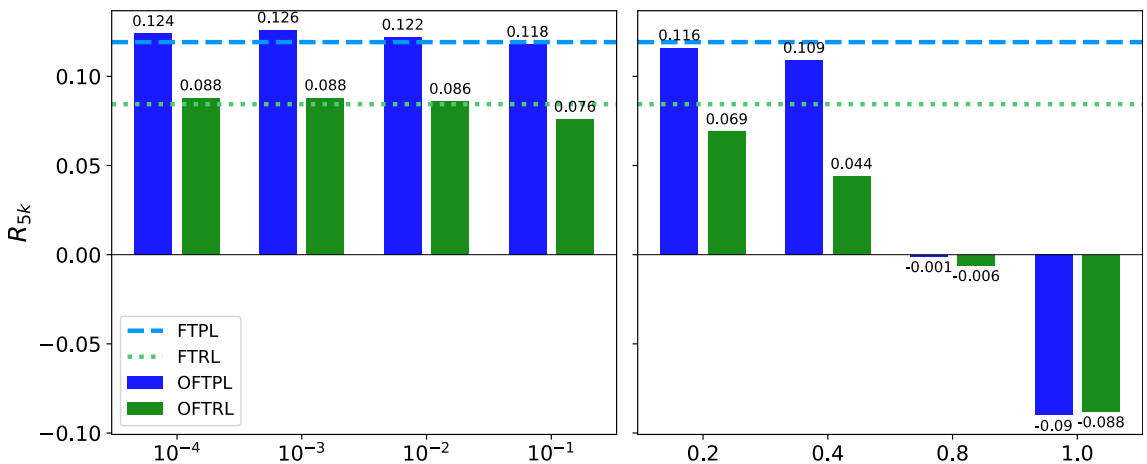


Figure: Prediction accuracy effect

Table of Contents

1 Background & Motivation

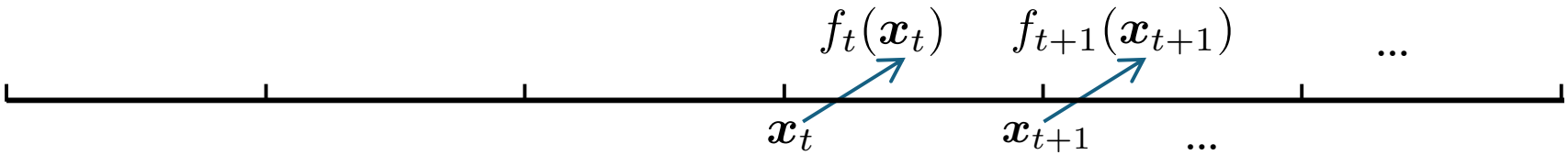
2 Preliminaries & Convex OL

3 Non-convex OL

4 OL & Non-stochastic Control

OCO with Memory

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



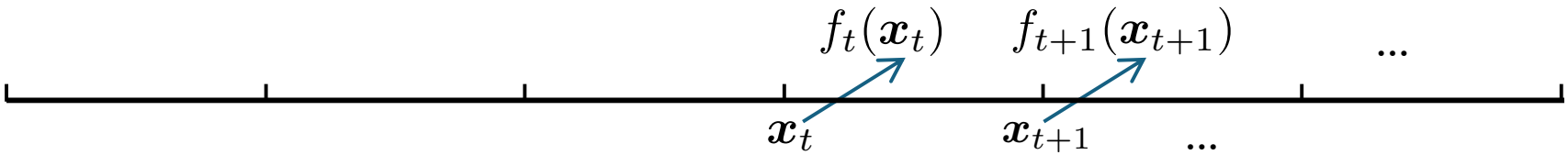
- OCO with memory (OCO-M): Pays for the past m decisions, $\boldsymbol{x}_t$:

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

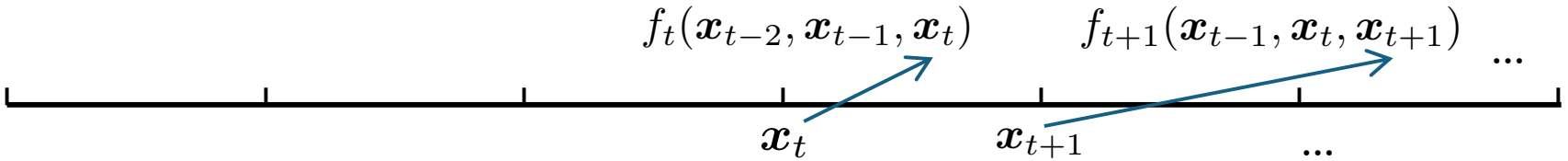
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

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, $\boldsymbol{x}_t$:

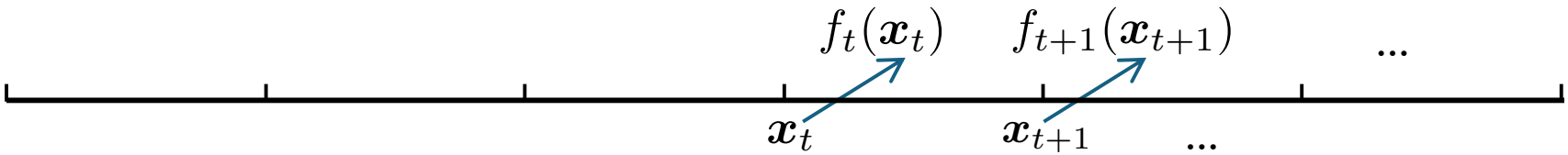


- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

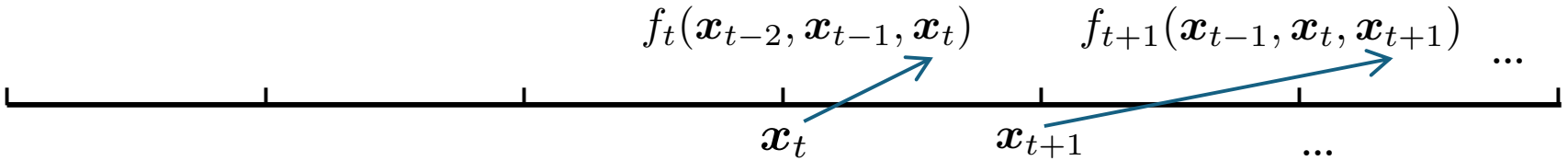
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

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, $\boldsymbol{x}_t$:

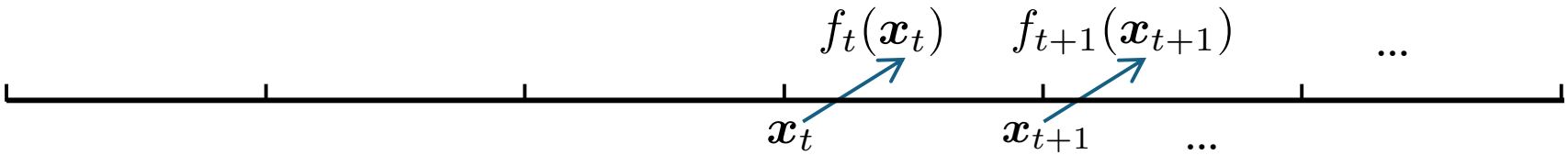


- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

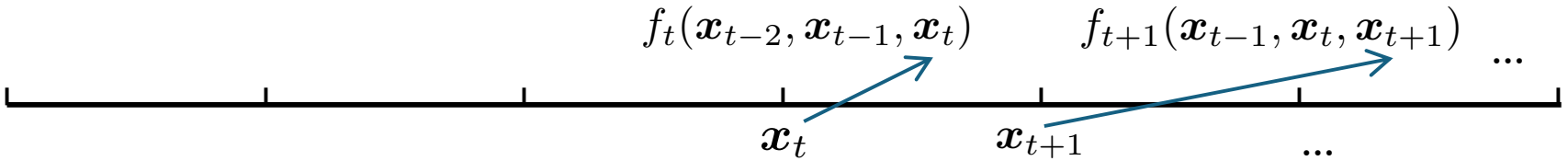
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

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, $\boldsymbol{x}_t$:

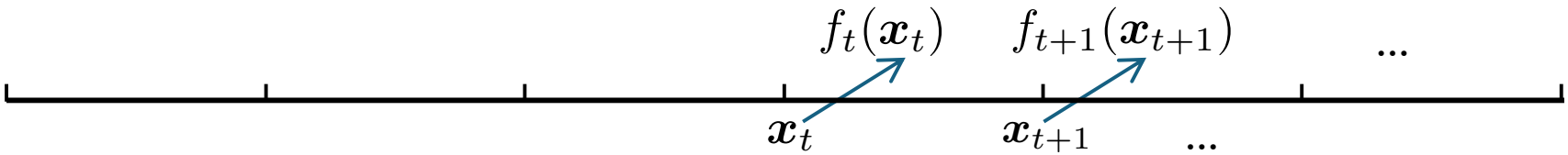


- $\mathcal{R}_T(\{\boldsymbol{x}_t\}_{t=1}^T) \leq \sum_{t=m}^T f_t(\boldsymbol{x}_{t-m}, \ldots, \boldsymbol{x}_t) - f_t(\boldsymbol{x}^\star, \ldots, \boldsymbol{x}^\star)$
- Unlimited memory leads to linear regret.
- OGD/FTRL achieve $\mathcal{O}(m\sqrt{T})$ for $1/m$ attenuated step size.

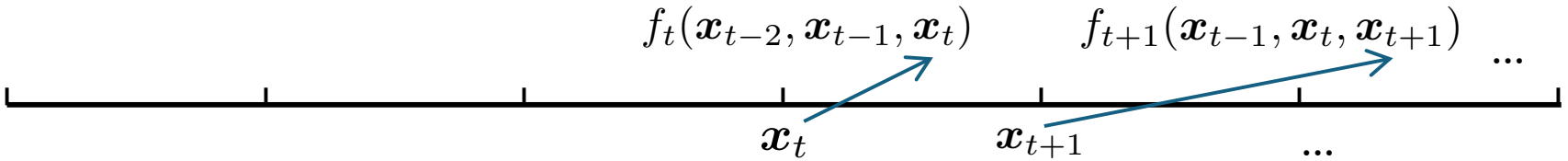
Avana et. al., Online learning for adversaries with memory: price of past mistakes , NeurIPS '15

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

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

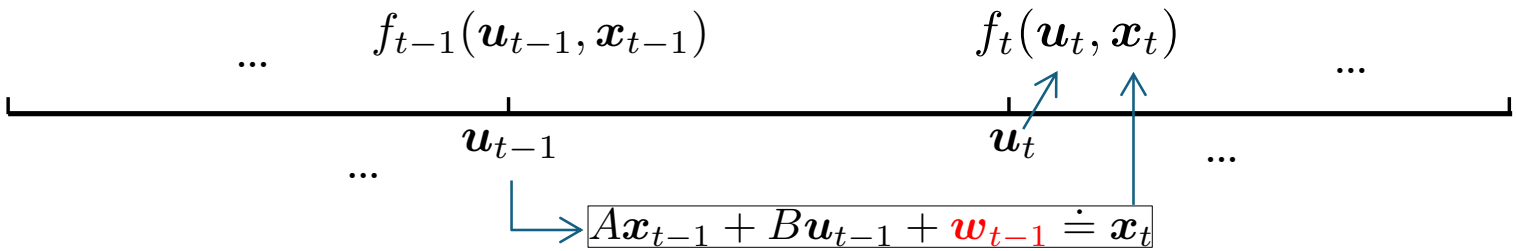
- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

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



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

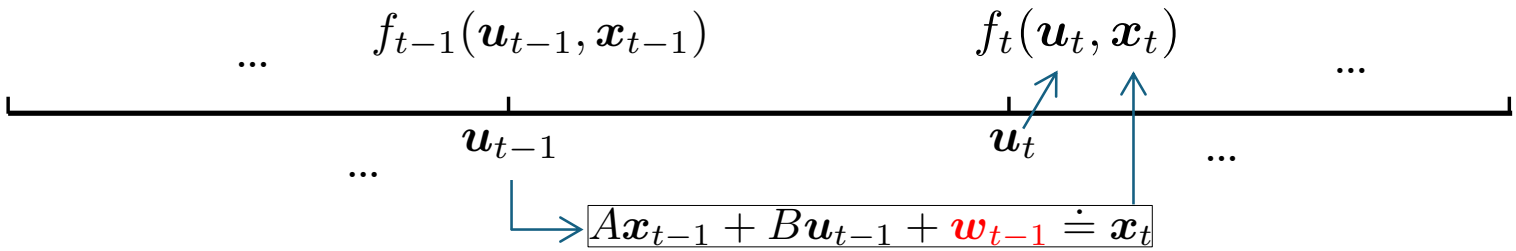
- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

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



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

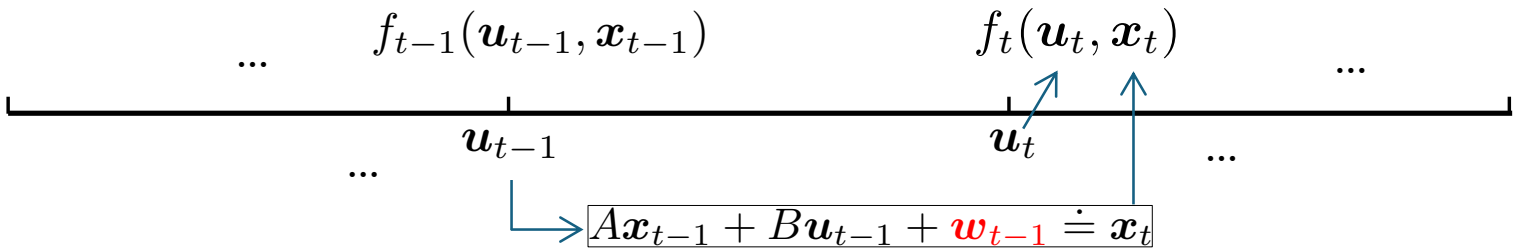
- $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the state-action pair under $\mathcal{A}$; $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)$ is under the benchmark.

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 $\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \underline{\boldsymbol{w}_t}$ *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) 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 3: 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 3: 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 3: 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 3: 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 3: 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. Interpretation 1: Adaptivity

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is small?
- AdaFTRL-C = FTRL + DAC-induced cost functions.

Algorithm 4: Adaptive FTRL Controller (`AdaFRTL-C`)

Input: An intrinsically stable LTI system (A, B), $M_1 \in \mathcal{M}$.
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 $\boldsymbol{w_t} = \boldsymbol{x}_{t+1} - A\boldsymbol{x_t} - B\boldsymbol{u_t}$
- Update regularization param. $\sigma_t \propto \sqrt{g_{1:t-1}}$
- $M_{t+1} = \arg\min_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t c_s(\boldsymbol{x}_s(\pi), \boldsymbol{u}_s(\pi)) + r_s(M) \right\}$.

end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{\sum_{t=1}^T g_t}$ with $g_t = \max(\|G_t\|, \|G_t\|^2)$

Mhaisen et. al., Adaptive online non-stochastic control , L4DC '24

Optimistic NSC. Interpretation 1: Adaptivity

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is small?
- AdaFTRL-C = FTRL + DAC-induced cost functions.

Algorithm 4: Adaptive FTRL Controller (`AdaFRTL-C`)

Input: An intrinsically stable LTI system (A, B), $M_1 \in \mathcal{M}$.
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 Update regularization param. $\sigma_t \propto \sqrt{g_{1:t-1}}$
7 $M_{t+1} = \arg\min_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t c_s(\boldsymbol{x}_s(\pi), \boldsymbol{u}_s(\pi)) + r_s(M) \right\}$.
end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{\sum_{t=1}^T g_t}$ with $g_t = \max(\|G_t\|, \|G_t\|^2)$

Mhaisen et. al., Adaptive online non-stochastic control , L4DC '24

Optimistic NSC. Interpretation 1: Adaptivity

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is small?
- AdaFTRL-C = FTRL + DAC-induced cost functions.

Algorithm 4: Adaptive FTRL Controller (`AdaFRTL-C`)

Input: An intrinsically stable LTI system (A, B), $M_1 \in \mathcal{M}$.
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 $\boldsymbol{w_t} = \boldsymbol{x}_{t+1} - A\boldsymbol{x_t} - B\boldsymbol{u_t}$
- Update regularization param. $\sigma_t \propto \sqrt{g_{1:t-1}}$
- $M_{t+1} = \arg\min_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t c_s(\boldsymbol{x}_s(\pi), \boldsymbol{u}_s(\pi)) + r_s(M) \right\}$.

end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa)\sqrt{\sum_{t=1}^T g_t}$ with $g_t = \max(\|G_t\|, \|G_t\|^2)$

Mhaisen et. al., Adaptive online non-stochastic control , L4DC '24

Optimistic NSC. Interpretation 1: Adaptivity

- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is small?
- AdaFTRL-C = FTRL + DAC-induced cost functions.

Algorithm 4: Adaptive FTRL Controller (`AdaFRTL-C`)

Input: An intrinsically stable LTI system (A, B), $M_1 \in \mathcal{M}$.
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 $\boldsymbol{w_t} = \boldsymbol{x}_{t+1} - A\boldsymbol{x_t} - B\boldsymbol{u_t}$
- Update regularization param. $\sigma_t \propto \sqrt{g_{1:t-1}}$
- $M_{t+1} = \arg\min_{M \in \mathcal{M}} \left\{ \sum_{s=1}^t c_s(\boldsymbol{x}_s(\pi), \boldsymbol{u}_s(\pi)) + r_s(M) \right\}$.

end

- $\mathcal{R}_T^{\text{pol}}(\mathcal{A}) = pol(G_c, \delta, \kappa_M, \kappa) \sqrt{\sum_{t=1}^T g_t}$ with $g_t = \max(\|G_t\|, \|G_t\|^2)$

Mhaisen et. al., Adaptive online non-stochastic control , L4DC '24

Optimistic NSC. Interpretation 2: Predictions (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. Interpretation 2: Predictions (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

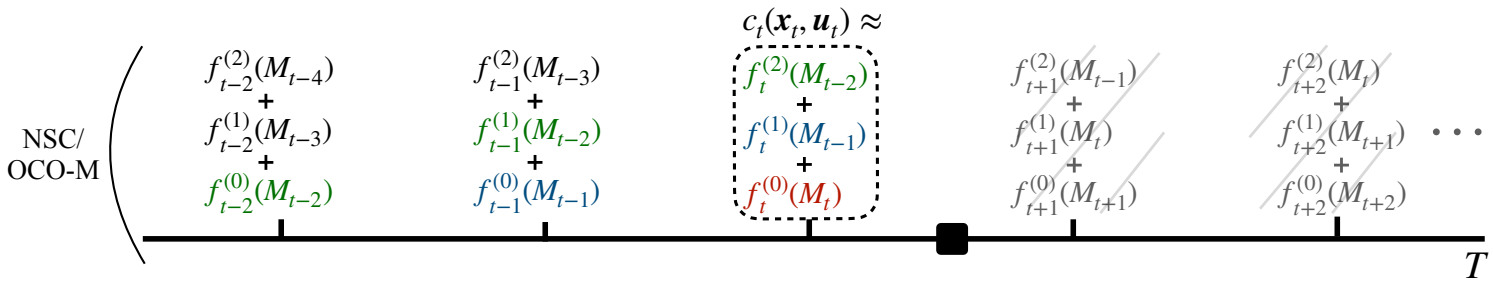
Optimistic NSC. Interpretation 2: Predictions (1/3)

- 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. Interpretation 2: Predictions (1/3)

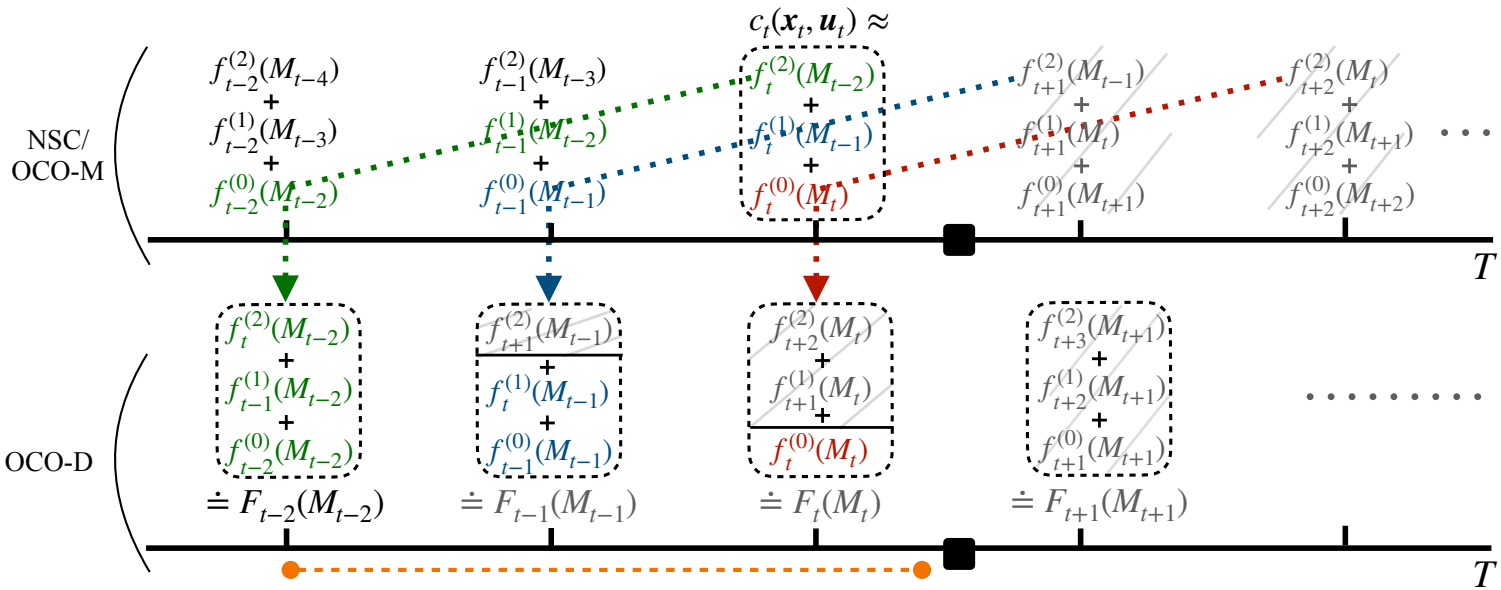
- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable



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

Optimistic NSC. Interpretation 2: Predictions (1/3)

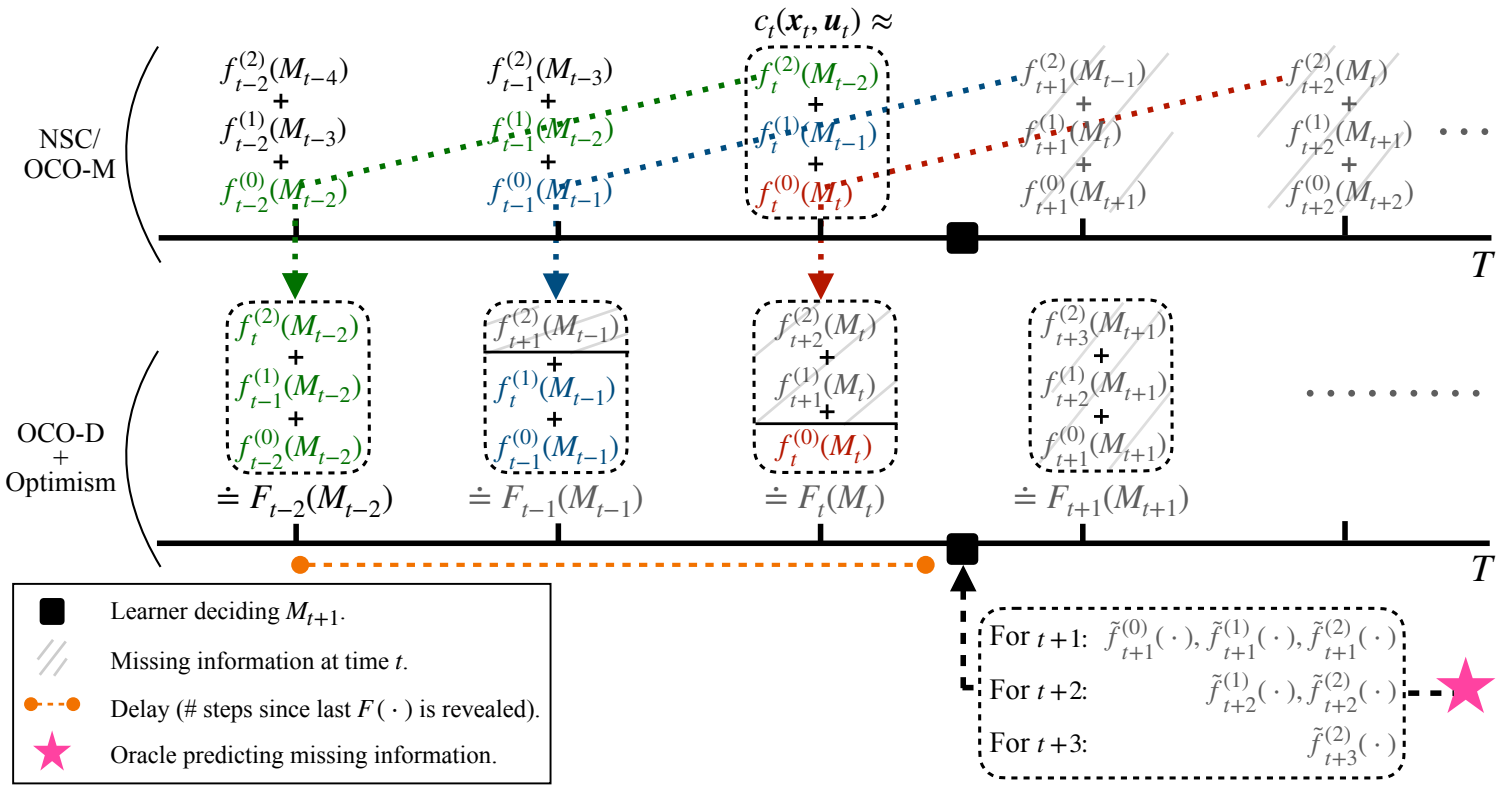
- What if $G_t(M) = \nabla f_t(M) \doteq c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is predictable?
- Recall $\boldsymbol{x}_t(\pi)$ and $\boldsymbol{u}_t(\pi)$ are linear in M
- if $c_t(\cdot, \cdot)$ is also linear, then $f_t(M) = c_t(\boldsymbol{x}(M), \boldsymbol{u}(M))$ is separable



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

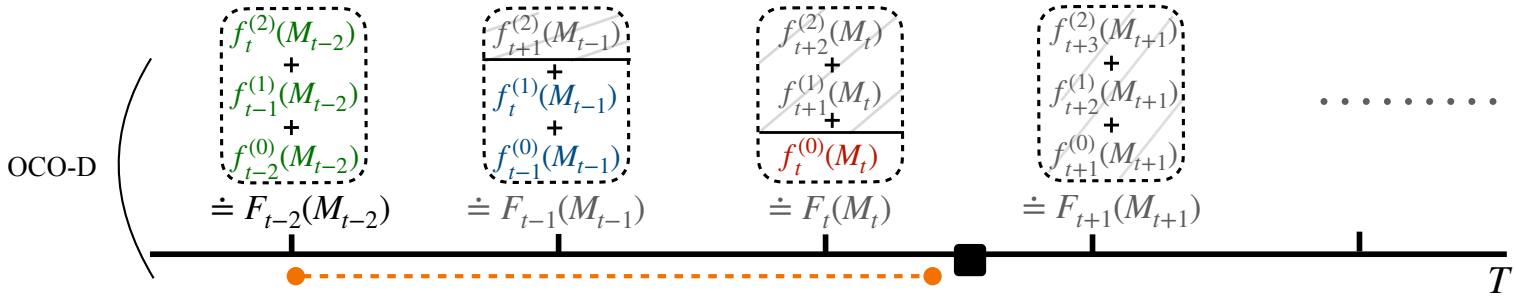
Optimistic NSC. Interpretation 2: Predictions (1/3)

- 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. Interpretation 2: Predictions (2/3)



Algorithm 5: `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**

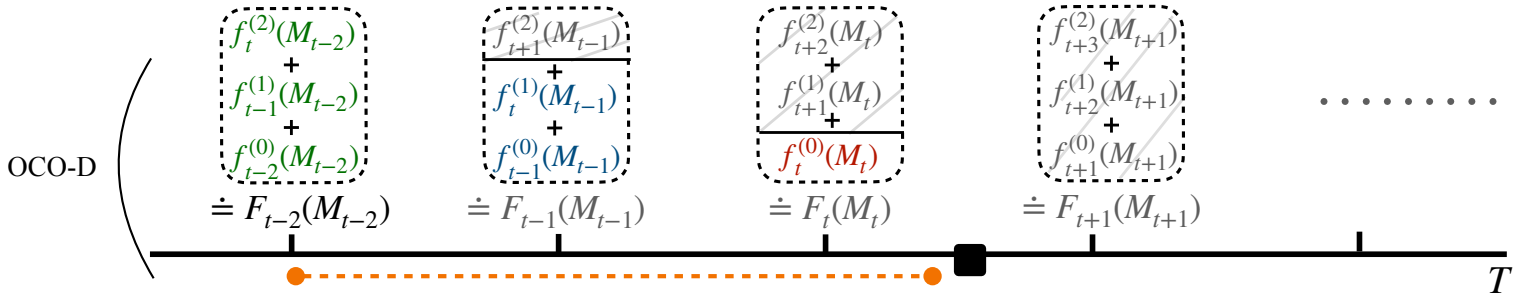
1. Use action $\boldsymbol{u}_t = \sum_{j=1}^p M_t^{[j]} \boldsymbol{w}_{t-j}$
2. Observe $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and record the gradient G_{t-d}
3. Observe new state $\boldsymbol{x}_{t+1}$ and record $\boldsymbol{w}_t$
4. Update s.c. $\sigma_t \approx \sqrt{\sum_{i=0}^{t-1} \epsilon_i^2} + \max_{j \le t-d-2} \epsilon_{j-d+1:j}$
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.

Mhaisen et. al., Optimistic online non-stochastic control via FTRL, arXiv 2404.03309

Optimistic NSC. Interpretation 2: Predictions (2/3)



Algorithm 5: `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

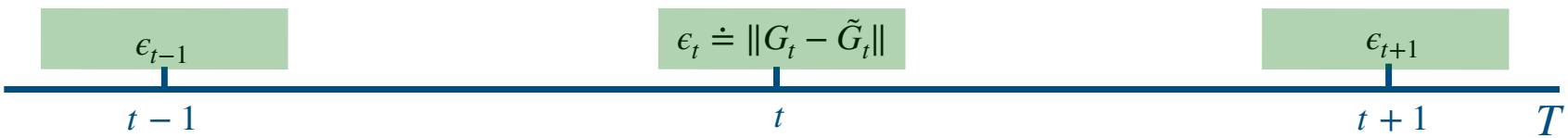
- OptFTRL-C = Delayed OFTRL.

Mhaisen et. al., Optimistic online non-stochastic control via FTRL, arXiv 2404.03309

Optimistic NSC. Interpretation 2: Prediction Oracle (3/2)

- Recall: Optimistic bounds accumulate the prediction errors sublinearly:

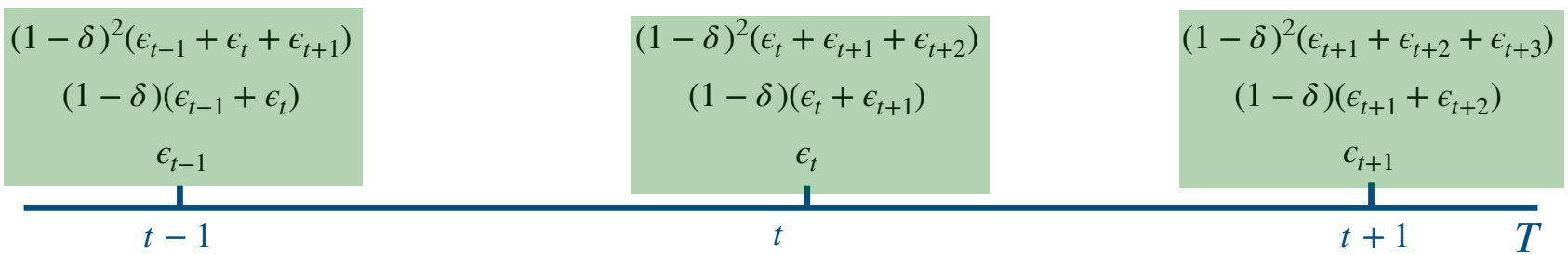
$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \epsilon_t^2}\right).$$



Optimistic NSC. Interpretation 2: Prediction Oracle (3/2)

- 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

NSC extensions:

- Foster et al. Logarithmic regret for adversarial online control ICML '20.
- Liu et al. Online nonstochastic control with adversarial and static constraints ICML '23.
- Gradu et. al "Adaptive regret for control of time-varying dynamics" L4DC '23.
- Goel et. al. "Best of both worlds in online control: Competitive ratio and policy regret" L4DC '23
- Lin et. al "Online Adaptive Policy Selection in Time-Varying Systems: No-Regret via Contractive Perturbations" NeurIPS '24

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

NSC extensions:

- Foster et al. Logarithmic regret for adversarial online control ICML '20.
- Liu et al. Online nonstochastic control with adversarial and static constraints ICML '23.
- Gradu et. al "Adaptive regret for control of time-varying dynamics" L4DC '23.
- Goel et. al. "Best of both worlds in online control: Competitive ratio and policy regret" L4DC '23
- Lin et. al "Online Adaptive Policy Selection in Time-Varying Systems: No-Regret via Contractive Perturbations" NeurIPS '24

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

NSC extensions:

- Foster et al. Logarithmic regret for adversarial online control ICML '20.
- Liu et al. Online nonstochastic control with adversarial and static constraints ICML '23.
- Gradu et. al "Adaptive regret for control of time-varying dynamics" L4DC '23.
- Goel et. al. "Best of both worlds in online control: Competitive ratio and policy regret" L4DC '23
- Lin et. al "Online Adaptive Policy Selection in Time-Varying Systems: No-Regret via Contractive Perturbations" NeurIPS '24

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

The End

- Acknowledgments to projects:
  - H2020 **DAEMON**, Network Intelligence for Adaptive and Self-Learning Mobile Networks, `https://h2020daemon.eu/`.
  - Horizon **ORIGAMI**, Optimized Resource Integration and Global Architecture for Mobile Infrastructure for 6G, `https://sns-origami.eu/`.

- Acknowledgments to collaborators:
  - Douglas J. Leith, Daron Anderson (Trinity College Dublin).
  - Jean-Baptiste Monteil, Ivana Dusparic (Trinity College Dublin).
  - Abhishek Sinha (Tata Institute, India), Georgios Paschos (Amazon, Lux).