Learning-Augmented Algorithms for MTS with Bandit Access to Multiple Predictors

Matei Gabriel Coșa **Marek Eliáš**

ICML 2025

Bocconi University

Illustrating example: Ice cream stall problem

- I want to run an ice cream stall on a beach

Illustrating example: Ice cream stall problem

- I want to run an ice cream stall on a beach

Ice cream stall composition

- large bucket for the main flavor of ice cream
- small containers for the other flavors

Illustrating example: Ice cream stall problem

- I want to run an ice cream stall on a beach

Ice cream stall composition

- large bucket for the main flavor of ice cream
- small containers for the other flavors

Which main flavor to choose?

- main flavor is cheap;
- small container gets empty → request a refill → surcharge

Illustrating example: Ice cream stall problem

- I want to run an ice cream stall on a beach

Ice cream stall composition

- large bucket for the main flavor of ice cream
- small containers for the other flavors

Which main flavor to choose?

- main flavor is cheap;
- small container gets empty → request a refill → surcharge

I have no previous experience

- I can observe local experts on other beaches
- I don't know which expert is going to do better

Strawberry ice cream

Inkfish ice cream – Flavor of the summer 2025!

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
	Surcharge:	Surcharge:	Surcharge:

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
08:00–08:30	Surcharge: 0	Surcharge: $	Surcharge: $

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
	Surcharge:	Surcharge:	Surcharge:
08:00–08:30	0	$	$
08:30–09:00	0	$	$

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
	Surcharge:	Surcharge:	Surcharge:
08:00–08:30	0	$	$
08:30–09:00	0	$	$
09:00–09:30	$	0	0

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
	Surcharge:	Surcharge:	Surcharge:
08:00–08:30	0	$	$
08:30–09:00	0	$	$
09:00–09:30	$	0	0
09:30–10:00	0	$	$

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
	Surcharge:	Surcharge:	Surcharge:
08:00–08:30	0	$	$
08:30–09:00	0	$	$
09:00–09:30	$	0	0
09:30–10:00	0	$	$
10:00–10:30	0	$	$

Illustrating example: Ice cream stall problem

- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2



Time slot	Expert 1	Expert 2	ME
	Surcharge:	Surcharge:	Surcharge:
08:00–08:30	0	$	$
08:30–09:00	0	$	$
09:00–09:30	$	0	0
09:30–10:00	0	$	$
10:00–10:30	0	$	$
10:30–11:00	0	$	$

Illustrating example: Ice cream stall problem

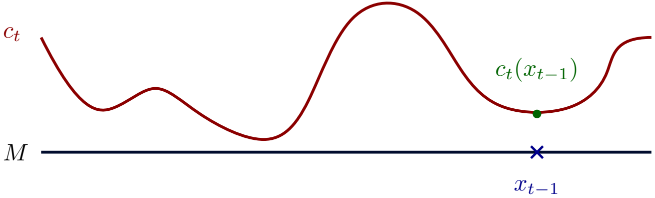
- main flavor cheap; other flavors provided with surcharge
- Let's imitate Expert 2

Time slot	Expert 1	Expert 2	ME
			
	Surcharge:	Surcharge:	Surcharge:
08:00–08:30	0	$	$
08:30–09:00	0	$	$
09:00–09:30	$	0	0
09:30–10:00	0	$	$
10:00–10:30	0	$	$
10:30–11:00	0	$	$
11:00–11:30	0	$	$

Metrical Task Systems (MTS) [Borodin, Linial, Saks ’92]

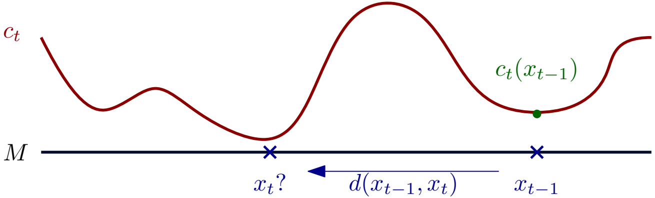
- metric space of states (M, d), initial state $x_0 \in M$

Metrical Task Systems (MTS) [Borodin, Linial, Saks '92]



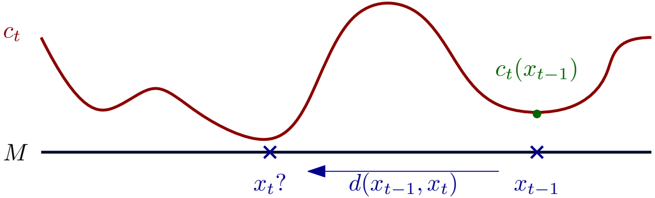
- metric space of states (M, d), initial state $x_0 \in M$
- Sequence of costs received online:
  - at time t, we receive $c_t \colon M \to \mathbb{R}_+ \cup \{+\infty\}$

Metrical Task Systems (MTS) [Borodin, Linial, Saks '92]



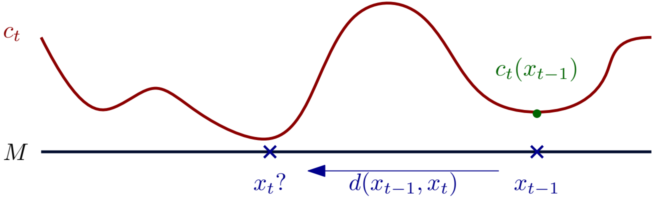
- metric space of states (M, d), initial state $x_0 \in M$
- Sequence of costs received online:
  - at time t, we receive $c_t \colon M \to \mathbb{R}_+ \cup \{+\infty\}$
  - we choose $x_t \in M$ and pay $c_t(x_t) + d(x_t, x_{t-1})$

Metrical Task Systems (MTS) [Borodin, Linial, Saks '92]



- metric space of states (M, d), initial state $x_0 \in M$
- Sequence of costs received online:
  - at time t, we receive $c_t \colon M \to \mathbb{R}_+ \cup \{+\infty\}$
  - we choose $x_t \in M$ and pay $c_t(x_t) + d(x_t, x_{t-1})$
- Target: minimize $\sum_{t=1}^{T} c_t(x_t) + d(x_t, x_{t-1})$

Metrical Task Systems (MTS) [Borodin, Linial, Saks '92]



- metric space of states (M, d), initial state $x_0 \in M$
- Sequence of costs received online:
  - at time t, we receive $c_t \colon M \to \mathbb{R}_+ \cup \{+\infty\}$
  - we choose $x_t \in M$ and pay $c_t(x_t) + d(x_t, x_{t-1})$
- Target: minimize $\sum_{t=1}^{T} c_t(x_t) + d(x_t, x_{t-1})$

Benchmark

- Offline optimum: the best trajectory $x_1, \ldots, x_T \in M$

$$\mathrm{OFF} = \min_{x_1,\ldots,x_T} \sum_{t=1}^{T} c_t(x_t) + d(x_t, x_{t-1})$$

Combining heuristics

What are our experts?

- heuristics $H_1, \ldots, H_\ell$ simulated on the same MTS instance
- our decisions: based on their states $s_t^1, \ldots, s_t^\ell$
- we want: cost $\leq (1+\epsilon)\min\{H_1, \ldots, H_\ell\}$ on every input

Combining heuristics

What are our experts?

- heuristics $H_1, \ldots, H_\ell$ simulated on the same MTS instance
- our decisions: based on their states $s_t^1, \ldots, s_t^\ell$
- we want: cost $\leq (1+\epsilon)\min\{H_1, \ldots, H_\ell\}$ on every input

Challenge

- online setting: which H_i will be the best?

Combining heuristics

What are our experts?

- heuristics $H_1, \ldots, H_\ell$ simulated on the same MTS instance
- our decisions: based on their states $s_t^1, \ldots, s_t^\ell$
- we want: cost $\leq (1+\epsilon)\min\{H_1, \ldots, H_\ell\}$ on every input

Challenge

- online setting: which H_i will be the best?

Full information

At each time t, observe the state s_t^i of H_i for each $i = 1, \ldots, \ell$

Combining heuristics

What are our experts?

- heuristics $H_1, \dots, H_\ell$ simulated on the same MTS instance
- our decisions: based on their states $s_t^1, \dots, s_t^\ell$
- we want: cost $\leq (1+\epsilon)\min\{H_1, \dots, H_\ell\}$ on every input

Challenge

- online setting: which H_i will be the best?

Full information

At each time t, observe the state s_t^i of H_i for each $i = 1, \dots, \ell$

Bandit-style access

At each time t, choose single $i_t \in \{1, \dots, \ell\}$ and observe $s_t^{i_t}$

Related works

Full information setting

- [Blum, Burch 2000]
- [Antoniadis et al. 2023]

Related works

Full information setting

- [Blum, Burch 2000]
- [Antoniadis et al. 2023]

Bandits with memory-bounded adversaries

- [Arora et al. 2012]

Related works

Full information setting

- [Blum, Burch 2000]
- [Antoniadis et al. 2023]

Bandits with memory-bounded adversaries

- [Arora et al. 2012]

Bandits with switching costs

- [Dekel et al. 2013]

Our results

Setting and parameters

- MTS on a metric space with diameter D
- bandit access to heuristics $H_1, \dots, H_\ell$
- benchmark $\mathrm{OPT} := \min\{H_1, \dots, H_\ell\}$

Our results

Setting and parameters

- MTS on a metric space with diameter D
- bandit access to heuristics $H_1, \ldots, H_\ell$
- benchmark $\mathrm{OPT} := \min\{H_1, \ldots, H_\ell\}$

Theorem

There is an algorithm ALG with cost

$$\mathbb{E}[\mathrm{ALG}] \leq \mathrm{OPT} + O\big((D\ell \log \ell)^{1/3}\, \mathrm{OPT}^{2/3}\big) = (1 + o(1))\, \mathrm{OPT}$$

on any input instance.

Our results

Setting and parameters

- MTS on a metric space with diameter D
- bandit access to heuristics $H_1, \ldots, H_\ell$
- benchmark $\mathrm{OPT} := \min\{H_1, \ldots, H_\ell\}$

Theorem

There is an algorithm ALG with cost

$$\mathbb{E}[\mathrm{ALG}] \le \mathrm{OPT} + O\big((D\ell \log \ell)^{1/3}\, \mathrm{OPT}^{2/3}\big) = (1 + o(1))\, \mathrm{OPT}$$

on any input instance.

Theorem

For any algorithm ALG there is an input instance such that

$$\mathbb{E}[\mathrm{ALG}] \ge \mathrm{OPT} + \tilde{\Omega}\big((D\ell)^{1/3}\, \mathrm{OPT}^{2/3}\big)$$

Implications for learning-augmented algorithms

Algorithms with predictions

- Algorithm receives predictions from ML models
- predictions are untrusted: arbitrarily good/bad
- consistency: great performance with good predictions
- robustness: never (much) worse than without predictions

Implications for learning-augmented algorithms

Algorithms with predictions

- Algorithm receives predictions from ML models
- predictions are untrusted: arbitrarily good/bad
- consistency: great performance with good predictions
- robustness: never (much) worse than without predictions

Achieving robustness

H_1 uses some prediction model, H_2 is a classical R-competitive algorithm. We pay at most $(1+\epsilon)\min\{H_1, R\cdot \mathrm{OFF}\}$.

Implications for learning-augmented algorithms

Algorithms with predictions

- Algorithm receives predictions from ML models
- predictions are untrusted: arbitrarily good/bad
- consistency: great performance with good predictions
- robustness: never (much) worse than without predictions

Achieving robustness

H_1 uses some prediction model, H_2 is a classical R-competitive algorithm. We pay at most $(1+\epsilon)\min\{H_1, R \cdot \mathrm{OFF}\}$.

Choosing prediction model online

Each $H_1, \ldots, H_\ell$ uses a different prediction model. We pay at most $(1+\epsilon)\min\{H_1, \ldots, H_\ell\}$.

Thank you for your attention!

mateigabriel.cosa@studbocconi.it

marek.elias@unibocconi.it