Algorithms with Predictions: learning challenges

Marek Eliáš, Bocconi University, Milan

Bocconi

Algorithms with Predictions (ALPS): Context

a.k.a. Learning-Augmented Algorithms

- Lykouris and Vassilvitskii (ICML'18, JACM)
- Kraska, Beutel, Chi, Dean, Polyzotis (SIGMOD'18)

Algorithms with Predictions (ALPS): Context

a.k.a. Learning-Augmented Algorithms

- Lykouris and Vassilvitskii (ICML'18, JACM)
- Kraska, Beutel, Chi, Dean, Polyzotis (SIGMOD'18)

Traditional theory of algorithms

- focus on worst-case performance
- theoretical analysis, strong formal guarantees

Algorithms with Predictions (ALPS): Context

a.k.a. Learning-Augmented Algorithms

- Lykouris and Vassilvitskii (ICML'18, JACM)
- Kraska, Beutel, Chi, Dean, Polyzotis (SIGMOD'18)

Traditional theory of algorithms

- focus on worst-case performance
- theoretical analysis, strong formal guarantees

Heuristic approaches

- focus on "typical" performance
- empirical analysis, no formal guarantees

Algorithms with Predictions (ALPS): Context

a.k.a. Learning-Augmented Algorithms

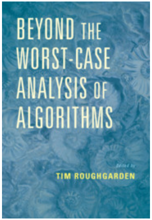
- Lykouris and Vassilvitskii (ICML'18, JACM)
- Kraska, Beutel, Chi, Dean, Polyzotis (SIGMOD'18)

Traditional theory of algorithms

- focus on worst-case performance
- theoretical analysis, strong formal guarantees

Heuristic approaches

- focus on "typical" performance
- empirical analysis, no formal guarantees



Beyond the worst-case analysis [Roughgarden 2021]

- What are the "typical" inputs?
  - stochastic properties, specific structure/patterns

Algorithms with Predictions (ALPS): Context

a.k.a. Learning-Augmented Algorithms

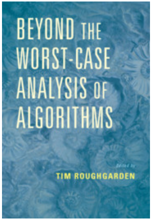
- Lykouris and Vassilvitskii (ICML'18, JACM)
- Kraska, Beutel, Chi, Dean, Polyzotis (SIGMOD'18)

Traditional theory of algorithms

- focus on worst-case performance
- theoretical analysis, strong formal guarantees

Heuristic approaches

- focus on "typical" performance
- empirical analysis, no formal guarantees



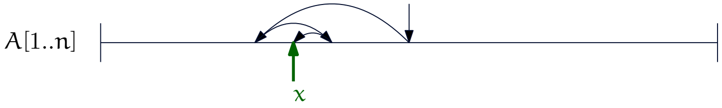
Beyond the worst-case analysis [Roughgarden 2021]

- What are the "typical" inputs?
  - stochastic properties, specific structure/patterns
- Chapter 30 on ALPS [Mitzenmacher, Vassilvitskii]

Example: Binary Search with Predictions

Classical Binary Search:

- Sorted array A
- Given x, find i s.t. $A[i] = x$



Example: Binary Search with Predictions

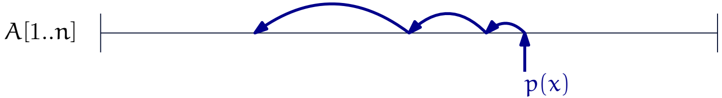
Binary Search with Predictions [Kraska et al. ’18]:

- Sorted array A
- Given x and untrusted prediction $p(x)$, find i s.t. $A[i] = x$

Example: Binary Search with Predictions

Binary Search with Predictions [Kraska et al. '18]:

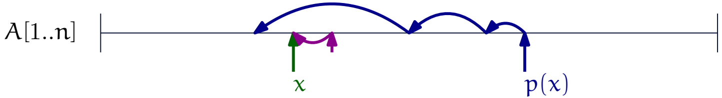
- Sorted array A
- Given x and untrusted prediction $p(x)$, find i s.t. $A[i] = x$



Example: Binary Search with Predictions

Binary Search with Predictions [Kraska et al. '18]:

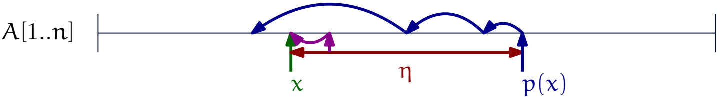
- Sorted array A
- Given x and untrusted prediction $p(x)$, find i s.t. $A[i] = x$



Example: Binary Search with Predictions

Binary Search with Predictions [Kraska et al. '18]:

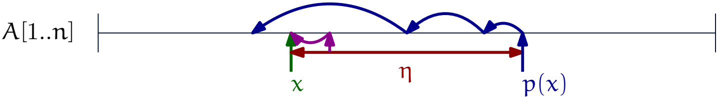
- Sorted array A
- Given x and untrusted prediction $p(x)$, find i s.t. $A[i] = x$



Example: Binary Search with Predictions

Binary Search with Predictions [Kraska et al. '18]:

- Sorted array A
- Given x and untrusted prediction $p(x)$, find i s.t. $A[i] = x$



Properties

- consistency: $O(1)$ steps with perfect predictions
- smoothness: $O(\log \eta)$ steps with error η
- robustness: never worse than $O(\log n)$ steps $(\eta \leqslant n)$

Template

Pick up an algorithmic problem

- online: caching, scheduling, routing, ...
- offline: matching, clustering, flows, ...

Template

Pick up an algorithmic problem

- online: caching, scheduling, routing, …
- offline: matching, clustering, flows, …

Identify helpful predictions

- reoccurence time, machine affinity, job size, …
- primal solution, dual solution, centers of clusters, …

Template

Pick up an algorithmic problem

- online: caching, scheduling, routing, ...
- offline: matching, clustering, flows, ...

Identify helpful predictions

- reoccurence time, machine affinity, job size, ...
- primal solution, dual solution, centers of clusters, ...

Define prediction error

- $\eta(\hat{p}, I)$: error of the prediction $\hat{p}$ on input I
- should be simple and reflect prediction's quality

Template

Pick up an algorithmic problem

- online: caching, scheduling, routing, ...
- offline: matching, clustering, flows, ...

Identify helpful predictions

- reoccurence time, machine affinity, job size, ...
- primal solution, dual solution, centers of clusters, ...

Define prediction error

- $\eta(\hat{p}, I)$: error of the prediction $\hat{p}$ on input I
- should be simple and reflect prediction's quality

Design algorithms

- for learning the predictions with small error (efficient)
- for using the predictions (consistent, smooth, robust)

Outline

(1) Learning the predictions

- PAC learning and beyond

(2) Achieving smoothness

- how to detect small errors?

(3) Adapting algorithmic strategy online

- learning from the best algorithm

(1) Learning the predictions

PAC learning paradigm

- $I_1, \ldots, I_k \sim \mathcal{D}$
- find prediction $\hat{p}$ s.t.

$$\mathbb{E}_{I \sim \mathcal{D}}[\eta(\hat{p}, I)] \leqslant \min_{p} \mathbb{E}_{I \sim \mathcal{D}}[\eta(p, I)] + \epsilon.$$

- related to Data-Driven Algorithm Design [Balcan et al.]

(1) Learning the predictions

PAC learning paradigm

- $I_1, \ldots, I_k \sim \mathcal{D}$
- find prediction $\hat{p}$ s.t.

$$\mathbb{E}_{I\sim\mathcal{D}}[\eta(\hat{p}, I)] \leqslant \min_{p} \mathbb{E}_{I\sim\mathcal{D}}[\eta(p, I)] + \epsilon.$$

- related to Data-Driven Algorithm Design [Balcan et al.]

We want:

- k polynomial
- polytime algorithm for minimizing empirical error

(1) Learning the predictions

PAC learning paradigm

- $I_1, \ldots, I_k \sim \mathcal{D}$
- find prediction $\hat{p}$ s.t.

$$\mathbb{E}_{I \sim \mathcal{D}}[\eta(\hat{p}, I)] \leqslant \min_p \mathbb{E}_{I \sim \mathcal{D}}[\eta(p, I)] + \epsilon.$$

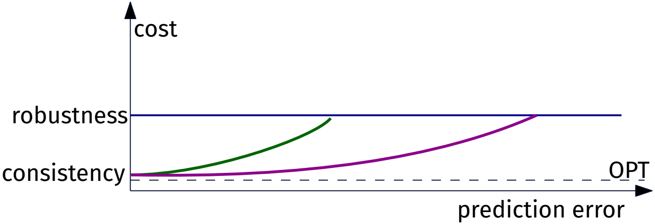
- related to Data-Driven Algorithm Design [Balcan et al.]

We want:

- k polynomial
- polytime algorithm for minimizing empirical error
  - often easy: scheduling [Lattanzi et al.'20], matching [Dinitz et al.'21]
  - but not always: shortest path [Lattanzi et al.'23]

Typical situation

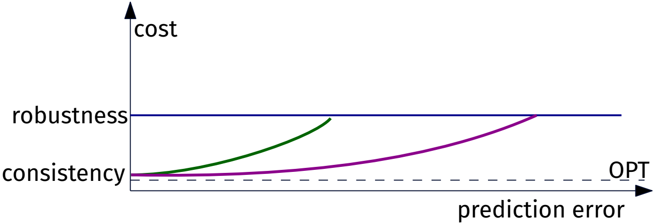
- prediction error η small but non-zero



(2) Smoothness

Typical situation

- prediction error η small but non-zero



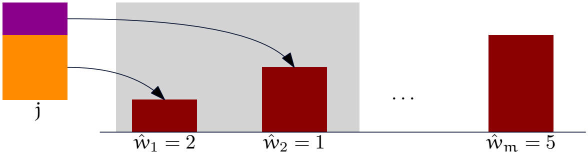
How to deal with imprecise predictions?

- $\hat{p}$ contains useful information but a few errors
- we need to find the errors (and correct them)

(2) Smoothness: Example

Online Load Balancing (Restricted Assignment) [Moseley et al. '20]

- predictions: machine weights $\hat{w}_1, \ldots, \hat{w}_m$,
  - jobs assigned to machines proportionally to the weights

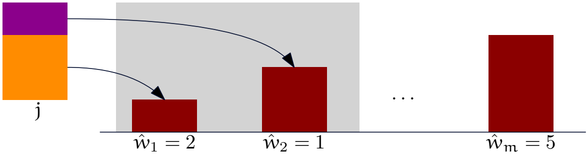


- $\eta = \max_i \frac{\hat{w}_i}{w_i}$

(2) Smoothness: Example

Online Load Balancing (Restricted Assignment) [Moseley et al. '20]

- predictions: machine weights $\hat{w}_1, \ldots, \hat{w}_m$,
  - jobs assigned to machines proportionally to the weights



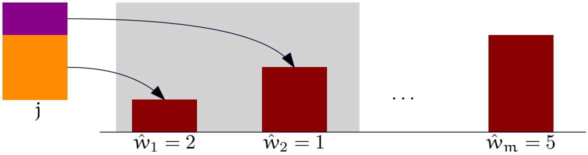
- $\eta = \max_i \frac{\hat{w}_i}{w_i}$

How to fix incorrect weights?

- load of machine i too high $\rightarrow$ divide w_i by 2

Online Load Balancing (Restricted Assignment) [Moseley et al. '20]

- predictions: machine weights $\hat{w}_1, \dots, \hat{w}_m$,
  - jobs assigned to machines proportionally to the weights



- $\eta = \max_i \frac{\hat{w}_i}{w_i}$

How to fix incorrect weights?

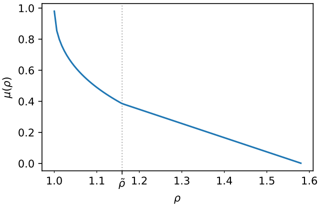
- load of machine i too high $\rightarrow$ divide w_i by 2

Smoothness bound [Moseley et al. '20]

There is an algorithm with competitive ratio $O(\log \eta)$ for the fractional restricted assignment.

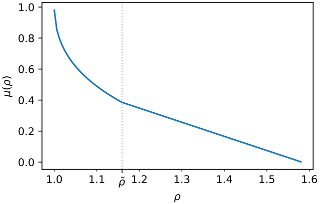
Online Dynamic Power Management

- $\mathrm{ALG}_\rho \leqslant \rho\,\mathrm{OPT} + \mu(\rho)\eta$ [Antoniadis, Coester, Eliáš, Polak, Simon '21]



Online Dynamic Power Management

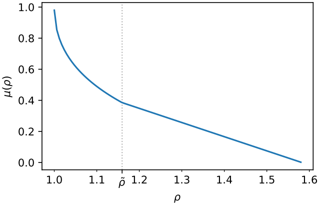
- $\mathrm{ALG}_\rho \leqslant \rho\,\mathrm{OPT} + \mu(\rho)\eta$ [Antoniadis, Coester, Eliáš, Polak, Simon '21]

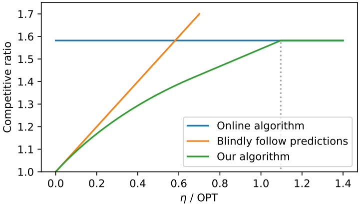


- we have to learn η online to set ρ optimally

Online Dynamic Power Management

- $\mathrm{ALG}_\rho \leqslant \rho\,\mathrm{OPT} + \mu(\rho)\eta$ [Antoniadis, Coester, Eliáš, Polak, Simon '21]

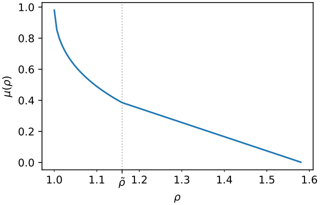


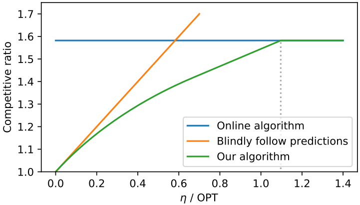


- we have to learn η online to set ρ optimally

Online Dynamic Power Management

- $\mathrm{ALG}_\rho \leqslant \rho\,\mathrm{OPT} + \mu(\rho)\eta$ [Antoniadis, Coester, Eliáš, Polak, Simon '21]





- we have to learn η online to set ρ optimally

Learning question

How much can we trust the prediction?

Algorithmic questions

How important is the current decision?

What is the alternative to following the prediction?

(3) Adapting strategy online

Two scenarios in online setting

- algorithm A uses predictor trained on distribution $\mathcal{D}$
- $\mathrm{I} \sim \mathcal{D} \Rightarrow A$ works great
- $\mathrm{I} \nsim \mathcal{D} \Rightarrow$ some worst-case A' still works ok
- I arriving online: we want cost $\approx \min\{A(\mathrm{I}), A'(\mathrm{I})\}$

(3) Adapting strategy online

Two scenarios in online setting

- algorithm A uses predictor trained on distribution $\mathcal{D}$
- $I \sim \mathcal{D} \Rightarrow A$ works great
- $I \nsim \mathcal{D} \Rightarrow$ some worst-case A' still works ok
- I arriving online: we want cost $\approx \min\{A(I), A'(I)\}$

Multiple scenarios in online setting

- A_i uses predictor trained on $\mathcal{D}_i$, $i = 1, \ldots, \ell$
- I is arriving online, we do not know what $\mathcal{D}_i$ it comes from
- we want $\approx \min\{A_1(I), \ldots, A_k(I)\}$

(3) Adapting strategy online

Two scenarios in online setting

- algorithm A uses predictor trained on distribution $\mathcal{D}$
- $I \sim \mathcal{D} \Rightarrow A$ works great
- $I \nsim \mathcal{D} \Rightarrow$ some worst-case A' still works ok
- I arriving online: we want cost $\approx \min\{A(I), A'(I)\}$

Multiple scenarios in online setting

- A_i uses predictor trained on $\mathcal{D}_i$, $i = 1, \ldots, \ell$
- I is arriving online, we do not know what $\mathcal{D}_i$ it comes from
- we want $\approx \min\{A_1(I), \ldots, A_k(I)\}$

Challenge:

- past actions $\rightarrow$ current state of the algorithm
- if we take an action of a bad algorithm, we cannot take it back

Related works

Online choice of online algorithms

- k-server [Fiat, Rabani, Ravid 1990]
- caching [Fiat, Karp, Luby, McGeoch, Sleator, Young 1991]
- general MTS [Azar, Broder, Manasse 1993]

Related works

Online choice of online algorithms

- k-server [Fiat, Rabani, Ravid 1990]
- caching [Fiat, Karp, Luby, McGeoch, Sleator, Young 1991]
- general MTS [Azar, Broder, Manasse 1993]

Algorithms with multiple predictions

- Online Covering [Anand et al. 2022], [Kevi, Nguyen 2023]
- Offline Matching, Scheduling [Dinitz et al. 2022]
- Ski Rental [Gollapudi, Panigrahi 2019], [Wang et al. 2020]
- Online Facility Location [Almanza et al. 2021]
- offline algorithms [Srinivas, Blum 2024]

Metrical Task Systems (MTS)

- very broad class of online problems
- includes caching, k-server, convex body chasing,...

Metrical Task Systems (MTS)

- very broad class of online problems
- includes caching, k-server, convex body chasing,...

Full information

- $\min\{A_1, \dots, A_\ell\} + \text{Regret}$ [Blum, Burch '00]
- $O(\ell^2)$ dyn.comb$(A_1, \dots, A_\ell)$ [Antoniadis, Coester, Eliáš, Polak, Simon '23]
  - Layered Graph Traversal [Bubeck, Coester, Rabani '22]

Metrical Task Systems (MTS)

- very broad class of online problems
- includes caching, k-server, convex body chasing,...

Full information

- $\min\{A_1, \ldots, A_\ell\} + \text{Regret}$ [Blum, Burch '00]
- $O(\ell^2)$ dyn.comb$(A_1, \ldots, A_\ell)$ [Antoniadis, Coester, Eliáš, Polak, Simon '23]
  - Layered Graph Traversal [Bubeck, Coester, Rabani '22]

Bandit setting

- $\min\{A_1, \ldots, A_\ell\} + \text{Regret}$ [Cosa, Eliáš '25]
- Dynamic combination: open

Further perspectives

Improve the current techniques to:

- improve existing bounds
- capture more problems and settings

Further perspectives

Improve the current techniques to:

- improve existing bounds
- capture more problems and settings

End-to-end learning of online algorithms

- Algorithms with Explicit Predictors
  [Eliáš, Kaplan, Mansour, Moran '23]
- algorithm has direct access to the dataset of past inputs
- algorithm learns while processing the input

Further perspectives

Improve the current techniques to:

- improve existing bounds
- capture more problems and settings

End-to-end learning of online algorithms

- Algorithms with Explicit Predictors
  [Eliáš, Kaplan, Mansour, Moran '23]
- algorithm has direct access to the dataset of past inputs
- algorithm learns while processing the input

Offline and Approximation algorithms

- currently understudied
- many techniques do not translate to offline setting

Further information

Posters

- Xizhi Tan: Learning-Augmented Mechanism Design
- Kaito Fujii: The Secretary Problem with Predictions

https://algorithms-with-predictions.github.io/

- by Alexander Lindermayr
- database of the papers in the area