Applied Online Algorithms with Heterogeneous Predictors

Jessica Maghakian[1], Russell Lee[2], Mohammad Hajiesmaili[2], Jian Li[3], Ramesh Sitaraman[2,4], Zhenhua Liu[1]

[1]Stony Brook University, [2]UMass Amherst, [3]SUNY Binghamton, [4]Akamai Tech

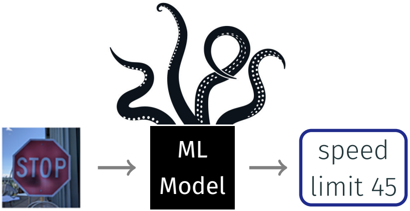
ICML 2023
Honolulu, Hawai'i
July 23 – 29, 2023

ICML
International Conference
On Machine Learning

Obstacles to adopting ML for applications

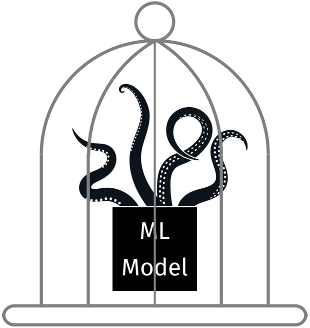
ML
Model

Obstacles to adopting ML for applications



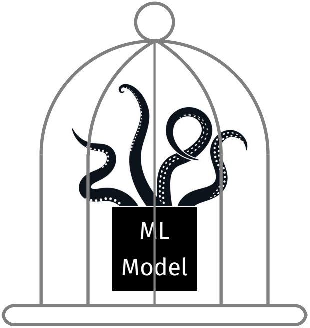
Obstacles to adopting ML for applications

Obstacles to adopting ML for applications



Learning-augmented algorithms offer a possible solution!

Obstacles to adopting ML for applications



Learning-augmented algorithms offer a possible solution!

Idea: design algorithms that can flexibly balance between data-driven and conservative worst-case decision making

Although learning-augmented algorithms are promising...

advantages of learning-augmented algorithms

Provide performance guarantees

Can exploit good predictions

Improve explainability

Learning-Augmented Algorithms

Although learning-augmented algorithms are promising... more work is required before they are suitable for applications in practice...

advantages of learning-augmented algorithms

- Provide performance guarantees
- Can exploit good predictions
- Improve explainability

disadvantages of learning-augmented algorithms

- Typically only use one prediction
- Computationally expensive
- Underutilize domain knowledge

Although learning-augmented algorithms are promising... more work is required before they are suitable for applications in practice...

advantages of learning-augmented algorithms

- Provide performance guarantees
- Can exploit good predictions
- Improve explainability

disadvantages of learning-augmented algorithms

- Typically only use one prediction
- Computationally expensive
- Underutilize domain knowledge

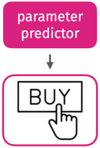
Our goal: provide insights that help future learning-augmented algorithms research be better aligned with real world desiderata

How to incorporate more data?

Our idea: using predictions of multiple *different* quantities

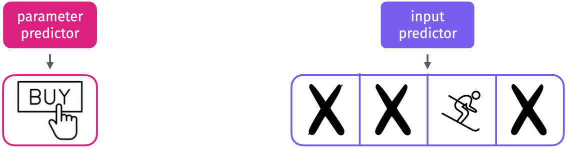
How to incorporate more data?

Our idea: using predictions of multiple *different* quantities



How to incorporate more data?

Our idea: using predictions of multiple *different* quantities



How to incorporate more data?

Our idea: using predictions of multiple *different* quantities

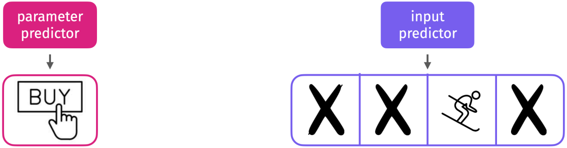
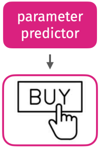


Table 1: Theoretical performance guarantees of using different predictions.

Algorithm	Consistency	Robustness
no predictions	2	2
parameter predictions	1	$1 + \epsilon_p \max\{\mu^-, \mu^+\}$
input predictions	$2 - \frac{w}{T}$	$2 - \frac{w}{T} + \epsilon_i \frac{\mu^-}{T}$

How to incorporate more data?

Our idea: using predictions of multiple *different* quantities



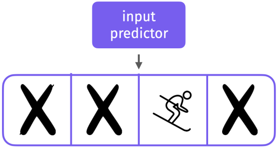


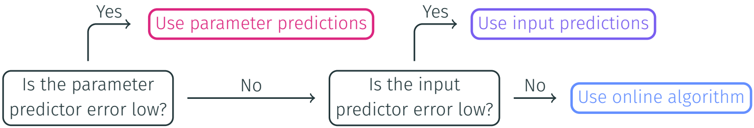
Table 1: Theoretical performance guarantees of using different predictions.

Algorithm	Consistency	Robustness
no predictions	2	2
parameter predictions	1	$1 + \epsilon_p \max\{\mu^-, \mu^+\}$
input predictions	$2 - \frac{w}{T}$	$2 - \frac{w}{T} + \epsilon_i \frac{\mu^-}{T}$

Different quantities have different performance profiles!

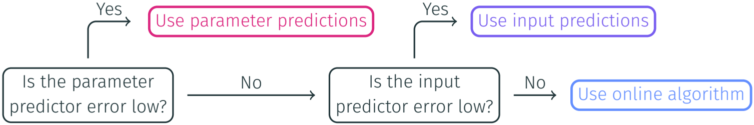
Result: Sometimes simpler is better

If we knew the errors of the different predictors:



Result: Sometimes simpler is better

If we knew the errors of the different predictors:



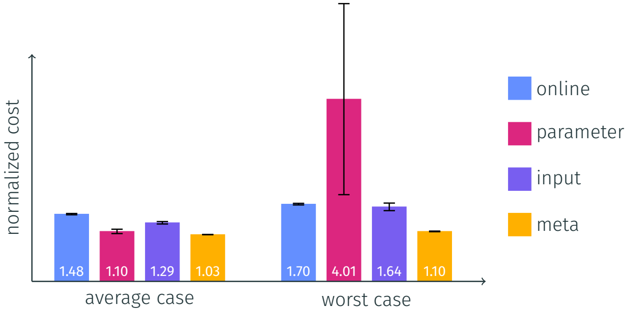
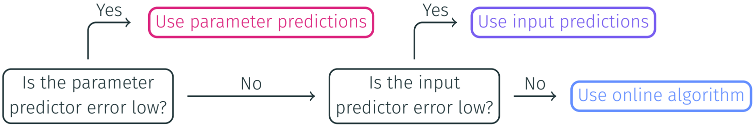


Figure 1: Results of large scale experiments (18000+ trials)

Result: Sometimes simpler is better

If we knew the errors of the different predictors:



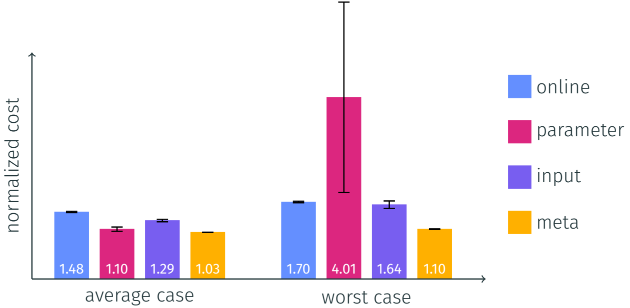


Figure 1: Results of large scale experiments (18000+ trials)

A simple classification approach succeeds on real-world data!

Result: Existing algorithms are too conservative

Algorithms with trust parameters: use a trust parameter λ to smoothly trade-off between average case and worst case performance

[1] Online algorithms for multi-shop ski rental with machine learned advice (*NeurIPS 2020*)

Result: Existing algorithms are too conservative

Algorithms with trust parameters: use a trust parameter λ to smoothly trade-off between average case and worst case performance

Table 2: Results over 9000+ trials with two different available parameter predictions

Algorithm	Average Case	Worst Case
parameter prediction #1	1.014 ± 0.005	2.817 ± 1.625
parameter prediction #2	1.039 ± 0.011	6.107 ± 4.164
our classification meta-alg	$\mathbf{1.022 \pm 0.007}$	$\mathbf{1.236 \pm 0.055}$
Wang et al. [1], $\lambda = 0.25$	1.052 ± 0.004	1.630 ± 0.315
Wang et al., $\lambda = 0.5$	1.102 ± 0.006	1.669 ± 0.312
Wang et al., $\lambda = 0.75$	1.155 ± 0.008	1.730 ± 0.312

[1] Online algorithms for multi-shop ski rental with machine learned advice (*NeurIPS 2020*)

Result: Existing algorithms are too conservative

Algorithms with trust parameters: use a trust parameter λ to smoothly trade-off between average case and worst case performance

Table 2: Results over 9000+ trials with two different available parameter predictions

Algorithm	Average Case	Worst Case
parameter prediction #1	1.014 ± 0.005	2.817 ± 1.625
parameter prediction #2	1.039 ± 0.011	6.107 ± 4.164
our classification meta-alg	$\mathbf{1.022 \pm 0.007}$	$\mathbf{1.236 \pm 0.055}$
Wang et al. [1], $\lambda = 0.25$	1.052 ± 0.004	1.630 ± 0.315
Wang et al., $\lambda = 0.5$	1.102 ± 0.006	1.669 ± 0.312
Wang et al., $\lambda = 0.75$	1.155 ± 0.008	1.730 ± 0.312

Increasing λ made both the average and worst case performance worse!

[1] Online algorithms for multi-shop ski rental with machine learned advice (*NeurIPS 2020*)

Result: 2 noisy predictors are better than 1

Both in theory...

Algorithm	Consistency	Robustness
RoBO-1	$1+\lambda$	$1+\lambda+\epsilon_{\text{par}}\big(\min\{\frac{1}{\lambda},\mu^-\}-\lambda\big)$
RoBO-2	$1+\lambda(1-\frac{W}{T})$	$1+\lambda(1-\frac{W-\epsilon_{\text{in}}\mu^-}{T})+\epsilon_{\text{par}}\big(\min\{\frac{1}{\lambda},\mu^+\}-\lambda\big)$

Table 3: Performance guarantees of using just one predictor (RoBO-1) vs two (RoBO-2)

Both in theory...

Algorithm	Consistency	Robustness
RoBO-1	$1 + \lambda$	$1 + \lambda + \epsilon_{\text{par}}\big(\min\{\frac{1}{\lambda}, \mu^-\} - \lambda\big)$
RoBO-2	$1 + \lambda(1 - \frac{W}{T})$	$1 + \lambda(1 - \frac{W - \epsilon_{\text{in}}\mu^-}{T}) + \epsilon_{\text{par}}\big(\min\{\frac{1}{\lambda}, \mu^+\} - \lambda\big)$

Table 3: Performance guarantees of using just one predictor (RoBO-1) vs two (RoBO-2)

... and in practice

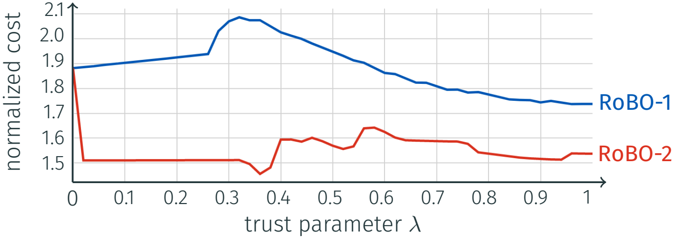


Figure 2: Algorithm performance on Covid-19 distributional shift dataset

Key takeaways from today's talk

1. Different predictors have different profiles that can be exploited in algorithm design (e.g. no longer require exploration).

Key takeaways from today's talk

1. Different predictors have different profiles that can be exploited in algorithm design (e.g. no longer require exploration).

2. Balancing online decision making with predictions at each time step is computationally expensive and unnecessary for many scenarios in practice.

Key takeaways from today's talk

1. Different predictors have different profiles that can be exploited in algorithm design (e.g. no longer require exploration).

2. Balancing online decision making with predictions at each time step is computationally expensive and unnecessary for many scenarios in practice.

3. Real-world applications require that we update our adversarial models to be more realistic and less pessimistic.