Algorithms with Predictions

Yingkai Li

EC4501/EC4501HM

Simple Illustration: Ski Rental

- A skier must decide whether to rent or buy skis.
- Renting costs r per day, buying costs B.
- The skier will ski for an "unknown" number of days T.
  - the weather becomes intolerable after T days.

Simple Illustration: Ski Rental

- A skier must decide whether to rent or buy skis.
- Renting costs r per day, buying costs B.
- The skier will ski for an "unknown" number of days T.
  - the weather becomes intolerable after T days.

Objective: minimize total cost without knowing T in advance.

Simple Illustration: Ski Rental

- A skier must decide whether to rent or buy skis.
- Renting costs r per day, buying costs B.
- The skier will ski for an "unknown" number of days T.
  - the weather becomes intolerable after T days.

Objective: minimize total cost without knowing T in advance.

Prediction: the number of days $\hat{T}$ that have good forecasted weather.

Simple Illustration: Ski Rental

Without predictions: the skier cannot make the optimal decision as if he knows the weather.

- the skier suffers from a loss in cost if he rents in early dates but T is large;
- the skier suffers from a loss in cost if he buys immediately but T is small.

Simple Illustration: Ski Rental

Without predictions: the skier cannot make the optimal decision as if he knows the weather.

- the skier suffers from a loss in cost if he rents in early dates but T is large;
- the skier suffers from a loss in cost if he buys immediately but T is small.

With perfect predictions: make the optimal choice based on whether $\hat{T} > \frac{B}{r}$.

Simple Illustration: Ski Rental

Without predictions: the skier cannot make the optimal decision as if he knows the weather.

- the skier suffers from a loss in cost if he rents in early dates but T is large;
- the skier suffers from a loss in cost if he buys immediately but T is small.

With perfect predictions: make the optimal choice based on whether $\hat{T} > \frac{B}{r}$.

Large prediction error: unbounded loss if naively following the prediction

- skier may rent the ski forever if the prediction is below $\frac{B}{r}$.

Simple Illustration: Ski Rental

Without predictions: the skier cannot make the optimal decision as if he knows the weather.

- the skier suffers from a loss in cost if he rents in early dates but T is large;
- the skier suffers from a loss in cost if he buys immediately but T is small.

With perfect predictions: make the optimal choice based on whether $\hat{T} > \frac{B}{r}$.

Large prediction error: unbounded loss if naively following the prediction

- skier may rent the ski forever if the prediction is below $\frac{B}{r}$.

Use predictions with caution!

Introduction

Applications of predictions: predictions acquired through ML/RL/AI or human expertise

- medical diagnosis and treatment planning;
- financial trading and investment;
- loan approval and credit scoring;
- fraud detection in banking;
- dynamic pricing on ride-sharing platforms;
- smart energy management;
- ...

Introduction

Applications of predictions: predictions acquired through ML/RL/AI or human expertise

- medical diagnosis and treatment planning;
- financial trading and investment;
- loan approval and credit scoring;
- fraud detection in banking;
- dynamic pricing on ride-sharing platforms;
- smart energy management;
- . . .

Why This Matters:

- ML predictions increasingly available in practice.
- Better predictions $\Rightarrow$ Better algorithms "for free".

Introduction

Use predictions with caution!

Introduction

Use predictions with caution!

Goals: robust performance guarantee when prediction performance is not always reliable

- Near-optimal when predictions are good;
- Graceful degradation with bad predictions.

Outline

- Framework and definitions
- Consistency-robustness tradeoff
- Classic Examples
  - Ski rental
  - Auctions
  - Job scheduling

Formal Framework

Algorithm Components

- **Predictor:** h maps inputs to predictions
- **Error Metric:** $\eta(h)$
- **Algorithm:** A_h parameterized by h

Formal Framework

Algorithm Components

- **Predictor:** h maps inputs to predictions
- **Error Metric:** $\eta(h)$
- **Algorithm:** A_h parameterized by h

Performance Guarantees

For approximation ratio $\mathrm{APX}(A_h)$:

- α-**Consistency**: $\lim_{\eta \to 0} \mathrm{APX}(A_h) \leq \alpha$
- β-**Robustness**: $\sup_{\eta} \mathrm{APX}(A_h) \leq \beta$

Ski Rental Problem

A skier must decide whether to rent or buy skis.

Ski Rental Problem

A skier must decide whether to rent or buy skis.

Approximation Ratio:

- The cost of an optimal offline strategy (knows T) is:

$$C^* = \min(B, T \cdot r).$$

- Let C be the cost of an online strategy (without knowing T).
- Approximation Ratio: $\sup_T \frac{C}{C^*}$.

Ski Rental: Optimal Online Strategy

- Rent for $\frac{B}{r}$ days, then buy if still skiing.

Ski Rental: Optimal Online Strategy

- Rent for $\frac{B}{r}$ days, then buy if still skiing.
- Total cost:

$$C = \begin{cases} Tr, & \text{if } T \leq \frac{B}{r} \\ B + (\frac{B}{r})r = 2B, & \text{if } T > \frac{B}{r} \end{cases}$$

Ski Rental: Optimal Online Strategy

- Rent for $\frac{B}{r}$ days, then buy if still skiing.
- Total cost:

$$C = \begin{cases} Tr, & \text{if } T \leq \frac{B}{r} \\ B + (\frac{B}{r})r = 2B, & \text{if } T > \frac{B}{r} \end{cases}$$

- Approximation ratio: $\sup_T \frac{C}{C^*} = 2$.

Ski Rental: Optimal Online Strategy

- Rent for $\frac{B}{r}$ days, then buy if still skiing.
- Total cost:

$$C = \begin{cases} Tr, & \text{if } T \leq \frac{B}{r} \\ B + (\frac{B}{r})r = 2B, & \text{if } T > \frac{B}{r} \end{cases}$$

- Approximation ratio: $\sup_T \frac{C}{C^*} = 2$.

This is the optimal deterministic strategy in an online setting.

Ski Rental: Predictions

Prediction error: $\eta \triangleq \left|\hat{T} - T\right|$.

Ski Rental: Predictions

Prediction error: $\eta \triangleq \left|\hat{T} - T\right|$.

Naively following the prediction leads to unbounded loss.

Ski Rental: Predictions

Prediction error: $\eta \triangleq \left|\hat{T} - T\right|$.

Naively following the prediction leads to unbounded loss.

Robust algorithm with prediction: parameter $\lambda \in [0, 1]$

- if $\hat{T} \geq \frac{B}{r}$, rent the ski until day $\lambda \cdot \frac{B}{r}$;
- if $\hat{T} < \frac{B}{r}$, rent the ski until day $\frac{B}{\lambda r}$.

Ski Rental: Predictions

Prediction error: $\eta \triangleq \left|\hat{T} - T\right|$.

Naively following the prediction leads to unbounded loss.

Robust algorithm with prediction: parameter $\lambda \in [0, 1]$

- if $\hat{T} \geq \frac{B}{r}$, rent the ski until day $\lambda \cdot \frac{B}{r}$;
- if $\hat{T} < \frac{B}{r}$, rent the ski until day $\frac{B}{\lambda r}$.

A smooth transition between optimal online and naively following prediction:

- $\lambda = 0$: naively following the prediction;
- $\lambda = 1$: optimal online.

Ski Rental: Predictions

Theorem

For any $\lambda \in [0,1]$, the approximation ratio of the robust algorithm is

$$1 + \min\left\{\frac{1}{\lambda}, \lambda + \frac{\eta r(1+\lambda)}{C^*}\right\}$$

Ski Rental: Predictions

Theorem

For any $\lambda \in [0,1]$, the approximation ratio of the robust algorithm is

$$1 + \min\left\{\frac{1}{\lambda}, \lambda + \frac{\eta r(1+\lambda)}{C^*}\right\}$$

- $\eta = 0$: approximation ratio is $1+\lambda$; $(1+\lambda)$-consistency
- $\eta \to \infty$: approximation ratio is $1+\frac{1}{\lambda}$; $(1+\frac{1}{\lambda})$-robustness.

Ski Rental: Proof

Case 1: $\hat{T} \geq \frac{B}{r}$ (Buy early)

- Subcase 1.1: $T \leq \lambda \cdot \frac{B}{r}$

$$\text{ALG} = \text{OPT} = T \cdot r$$
$$\mathrm{APX} = 1$$

- Subcase 1.2: $T > \lambda \cdot \frac{B}{r}$

$$\begin{aligned}
\text{ALG} &= \lambda B + B \\
\text{OPT} &\geq r \cdot T \geq r \cdot \max\{\hat{T} - \eta, \lambda \cdot \frac{B}{r}\} \\
&\geq \max\{B - \eta r, \lambda B\} \\
\mathrm{APX} &\leq 1 + \min\left\{\frac{1}{\lambda}, \lambda + \frac{\eta r(1+\lambda)}{B - \eta r}\right\}
\end{aligned}$$

Ski Rental: Proof

Case 1: $\hat{T} \geq \frac{B}{r}$ (Buy early)

- Subcase 1.1: $T \leq \lambda \cdot \frac{B}{r}$

$$\text{ALG} = \text{OPT} = T \cdot r$$
$$\mathrm{APX} = 1$$

- Subcase 1.2: $T > \lambda \cdot \frac{B}{r}$

$$\text{ALG} = \lambda B + B$$
$$\text{OPT} \geq r \cdot T \geq r \cdot \max\{\hat{T} - \eta, \lambda \cdot \frac{B}{r}\}$$
$$\geq \max\{B - \eta r, \lambda B\}$$
$$\mathrm{APX} \leq 1 + \min\left\{\frac{1}{\lambda}, \lambda + \frac{\eta r(1+\lambda)}{B - \eta r}\right\}$$

Case 2: $\hat{T} < \frac{B}{r}$ (Buy late)

- Subcase 2.1: $T \leq \frac{B}{r}$

$$\text{ALG} = \text{OPT} = T \cdot r$$
$$\mathrm{APX} = 1$$

- Subcase 2.2: $T > \frac{B}{r}$

$$\text{ALG} \leq \min\{\frac{B}{\lambda} + B, r \cdot (\hat{T} + \eta)\}$$
$$\leq \min\{\frac{B}{\lambda} + B, B + \eta r\}$$
$$\text{OPT} = B$$
$$\mathrm{APX} \leq 1 + \min\left\{\frac{1}{\lambda}, \frac{\eta r}{B}\right\}$$

Auctions

Selling a single item to a single buyer, with value $v \in [1, H]$

Auctions

Selling a single item to a single buyer, with value $v \in [1, H]$

Goal: revenue maximization.

- the seller does not observe v;
- the seller may have access to value estimate $\hat{v}$.

Auctions

Selling a single item to a single buyer, with value $v \in [1, H]$

Goal: revenue maximization.

- the seller does not observe v;
- the seller may have access to value estimate $\hat{v}$.

Prediction error: $\eta = \max\left\{\frac{\hat{v}}{v}, \frac{v}{\hat{v}}\right\}$.

Auctions

Without Predictions: uniformly randomly positing a price in $1, 2, 4, 8, \ldots, H$

- with probability $\frac{1}{\log H}$, the item is sold with a price within half of the value;
- the approximation ratio is $O(\log H)$.

Auctions

Without Predictions: uniformly randomly positing a price in $1, 2, 4, 8, \ldots, H$

- with probability $\frac{1}{\log H}$, the item is sold with a price within half of the value;
- the approximation ratio is $O(\log H)$.

With perfect prediction: post price $\hat{v}$.

- Leads to unbounded loss if $\eta > 1$.

Auctions: With Prediction

Theorem

For any parameter $\lambda \in (0,1), \gamma \geq 1$, *we can achieve an approximation ratio of* $\min\left\{ \frac{\log H}{\lambda}, \frac{\gamma}{1-\lambda} \cdot \mathbf{1}\left(\eta \leq \gamma\right) + \infty \cdot \mathbf{1}\left(\eta > \gamma\right) \right\}$.

Auctions: With Prediction

Theorem

For any parameter $\lambda \in (0,1), \gamma \geq 1$*, we can achieve an approximation ratio of* $\min\left\{\frac{\log H}{\lambda}, \frac{\gamma}{1-\lambda} \cdot \mathbf{1}\left(\eta \leq \gamma\right) + \infty \cdot \mathbf{1}\left(\eta > \gamma\right)\right\}$.

- with probability λ, uniformly randomly positing a price in $1, 2, 4, 8, \ldots, H$;
- with probability $1-\lambda$, posting a price $\frac{\hat{v}}{\gamma}$.

Auctions: With Prediction

Theorem

For any parameter $\lambda \in (0,1), \gamma \geq 1$, *we can achieve an approximation ratio of* $\min\left\{\frac{\log H}{\lambda}, \frac{\gamma}{1-\lambda} \cdot \mathbf{1}\left(\eta \leq \gamma\right) + \infty \cdot \mathbf{1}\left(\eta > \gamma\right)\right\}$.

- with probability λ, uniformly randomly positing a price in $1, 2, 4, 8, \ldots, H$;
- with probability $1-\lambda$, posting a price $\frac{\hat{v}}{\gamma}$.

Scale down the price with prediction by a factor of γ to tolerate more error in predictions.

Scheduling

An agent wishes to finish n tasks sequentially, and each task i requires c_i days to complete.

Scheduling

An agent wishes to finish n tasks sequentially, and each task i requires c_i days to complete.

Goal: minimizes the total waiting time of all tasks.

- the agent does not observe the required time c_i before the completion of task i;
- the agent may have access to predictions of c_i as $\hat{c}_i$.

Scheduling

An agent wishes to finish n tasks sequentially, and each task i requires c_i days to complete.

Goal: minimizes the total waiting time of all tasks.

- the agent does not observe the required time c_i before the completion of task i;
- the agent may have access to predictions of c_i as $\hat{c}_i$.

Assume w.l.o.g. non-decreasing actual processing times, i.e. $c_1 \leq \ldots \leq c_n$.

Scheduling: Without Predictions

Optimal Offline: If the agent can observe c_i, the optimal can be found by a greedy algorithm

- the agent completes the task with the lowest c_i first.

Scheduling: Without Predictions

Optimal Offline: If the agent can observe c_i, the optimal can be found by a greedy algorithm

- the agent completes the task with the lowest c_i first.

Optimal Online: round-robin procedure (RR)

- work on each unfinished task one by one.
- approximation ratio is 2.

Scheduling: Without Predictions

Optimal Offline: If the agent can observe c_i, the optimal can be found by a greedy algorithm

- the agent completes the task with the lowest c_i first.

Optimal Online: round-robin procedure (RR)

- work on each unfinished task one by one.
- approximation ratio is 2.

Let $d(i, j)$ be the amount of time by which i delays j. The performance of the algorithm is

$$\mathsf{ALG} = \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} \left(d(i,j) + d(j,i)\right).$$

Scheduling: Without Predictions

Optimal Offline: If the agent can observe c_i, the optimal can be found by a greedy algorithm

- the agent completes the task with the lowest c_i first.

Optimal Online: round-robin procedure (RR)

- work on each unfinished task one by one.
- approximation ratio is 2.

Let $d(i,j)$ be the amount of time by which i delays j. The performance of the algorithm is

$$\text{ALG} = \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} \left(d(i,j) + d(j,i)\right).$$

In RR, $d(i,j) + d(j,i) = 2\min\{c_i, c_j\}$.
In OPT, $d(i,j) + d(j,i) = \min\{c_i, c_j\}$.

Scheduling: With Predictions (SPJF)

Shortest Predicted Job First (SPJF):

- sort the tasks according to predicted time $\hat{c}_i$;
- complete the task with the smallest predicted time.

Scheduling: With Predictions (SPJF)

Shortest Predicted Job First (SPJF):

- sort the tasks according to predicted time $\hat{c}_i$;
- complete the task with the smallest predicted time.

Prediction error: $\eta = \sum_i \eta_i$ where $\eta_i = |\hat{c}_i - c_i|$.

Scheduling: With Predictions (SPJF)

Shortest Predicted Job First (SPJF):

- sort the tasks according to predicted time $\hat{c}_i$;
- complete the task with the smallest predicted time.

Prediction error: $\eta = \sum_i \eta_i$ where $\eta_i = |\hat{c}_i - c_i|$.

Lemma (Kumar, Purohit and Svitkina '18)

The SPJF algorithm has approximation ratio at most $1 + \frac{2\eta}{n}$.

Scheduling: With Predictions (SPJF)

Shortest Predicted Job First (SPJF):

- sort the tasks according to predicted time $\hat{c}_i$;
- complete the task with the smallest predicted time.

Prediction error: $\eta = \sum_i \eta_i$ where $\eta_i = |\hat{c}_i - c_i|$.

Lemma (Kumar, Purohit and Svitkina '18)

The SPJF algorithm has approximation ratio at most $1 + \frac{2\eta}{n}$.

The performance of SPJF is

$$\text{ALG} = \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} \left(d(i,j) + d(j,i)\right).$$

Scheduling: With Predictions (SPJF)

For $i < j$ such that $\hat{c}_i < \hat{c}_j$, in the optimal we have $d(i, j) + d(j, i) = c_i + 0$.

Scheduling: With Predictions (SPJF)

For $i < j$ such that $\hat{c}_i < \hat{c}_j$, in the optimal we have $d(i,j) + d(j,i) = c_i + 0$.
However, wrong prediction can lead the longer job to be scheduled first, so $d(i,j) + d(j,i) = 0 + c_j$.

Scheduling: With Predictions (SPJF)

For $i < j$ such that $\hat{c}_i < \hat{c}_j$, in the optimal we have $d(i,j) + d(j,i) = c_i + 0$. However, wrong prediction can lead the longer job to be scheduled first, so $d(i,j) + d(j,i) = 0 + c_j$. This yields

$$\mathsf{ALG} = \sum_{j=1}^{n} c_j + \sum_{\substack{(i,j):i<j \\ \hat{c}_i < \hat{c}_j}} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} c_j = \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} (c_j - c_i)$$

$$\leq \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} (\eta_i + \eta_j) = \mathsf{OPT} + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} (\eta_i + \eta_j) \leq \mathsf{OPT} + (n-1)\eta,$$

Scheduling: With Predictions (SPJF)

For $i < j$ such that $\hat{c}_i < \hat{c}_j$, in the optimal we have $d(i,j) + d(j,i) = c_i + 0$. However, wrong prediction can lead the longer job to be scheduled first, so $d(i,j) + d(j,i) = 0 + c_j$. This yields

$$\mathsf{ALG} = \sum_{j=1}^{n} c_j + \sum_{\substack{(i,j):i<j \\ \hat{c}_i < \hat{c}_j}} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} c_j = \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} (c_j - c_i)$$

$$\leq \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} (\eta_i + \eta_j) = \mathsf{OPT} + \sum_{\substack{(i,j):i<j \\ \hat{c}_i \geq \hat{c}_j}} (\eta_i + \eta_j) \leq \mathsf{OPT} + (n-1)\eta,$$

which yields

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} \leq 1 + \frac{(n-1)\eta}{\mathsf{OPT}}.$$

Scheduling: With Predictions (SPJF)

For $i < j$ such that $\hat{c}_i < \hat{c}_j$, in the optimal we have $d(i,j) + d(j,i) = c_i + 0$. However, wrong prediction can lead the longer job to be scheduled first, so $d(i,j) + d(j,i) = 0 + c_j$. This yields

$$\mathsf{ALG} = \sum_{j=1}^{n} c_j + \sum_{\substack{(i,j):i<j \\ \hat{c}_i<\hat{c}_j}} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i\geq\hat{c}_j}} c_j = \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i\geq\hat{c}_j}} (c_j - c_i)$$

$$\leq \sum_{j=1}^{n} c_j + \sum_{(i,j):i<j} c_i + \sum_{\substack{(i,j):i<j \\ \hat{c}_i\geq\hat{c}_j}} (\eta_i + \eta_j) = \mathsf{OPT} + \sum_{\substack{(i,j):i<j \\ \hat{c}_i\geq\hat{c}_j}} (\eta_i + \eta_j) \leq \mathsf{OPT} + (n-1)\eta,$$

which yields

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} \leq 1 + \frac{(n-1)\eta}{\mathsf{OPT}}.$$

Now, using our assumption that all jobs have length at least 1, we have $\mathsf{OPT} \geq \frac{n(n+1)}{2}$. This yields an upper bound of

$$1 + \frac{2(n-1)\eta}{n(n+1)} < 1 + \frac{2\eta}{n}.$$

Scheduling: With Predictions

Preferential Round-Robin (PRR)

- run SPJF with rate λ;
- run RR with rate $1 - \lambda$.

Scheduling: With Predictions

Preferential Round-Robin (PRR)

- run SPJF with rate λ;
- run RR with rate $1-\lambda$.

Theorem (Kumar, Purohit and Svitkina '18)

The preferential round-robin algorithm with parameter $\lambda \in (0,1)$ has an approximation ratio at most $\min\left\{\frac{1}{\lambda}\cdot(1+\frac{2\eta}{n}), \frac{2}{1-\lambda}\right\}$. In particular, it is $\frac{2}{1-\lambda}$-robust and $\frac{1}{\lambda}$-consistent.

Scheduling: With Predictions

An algorithm is *monotonic* if the resulting waiting time is lower if the required time for each task is lower.

- SPJF, RR, PRR are monotonic algorithms.

Scheduling: With Predictions

An algorithm is *monotonic* if the resulting waiting time is lower if the required time for each task is lower.

- SPJF, RR, PRR are monotonic algorithms.

Lemma (Kumar, Purohit and Svitkina '18)

Given two monotonic algorithms A and B with approximation ratios α *and* β *and a parameter* $\lambda \in (0,1)$, *one can obtain an algorithm with competitive ratio approximation* $\min\left\{\frac{\alpha}{\lambda}, \frac{\beta}{1-\lambda}\right\}$.

Runs algorithm A in λ fraction of the time and B in $1-\lambda$ fraction of the time.

Scheduling: With Predictions

An algorithm is *monotonic* if the resulting waiting time is lower if the required time for each task is lower.

- SPJF, RR, PRR are monotonic algorithms.

Lemma (Kumar, Purohit and Svitkina '18)

Given two monotonic algorithms A and B with approximation ratios α and β and a parameter $\lambda \in (0,1)$, one can obtain an algorithm with competitive ratio approximation $\min\left\{\frac{\alpha}{\lambda}, \frac{\beta}{1-\lambda}\right\}$.

Runs algorithm A in λ fraction of the time and B in $1-\lambda$ fraction of the time.

- running A in λ fraction of the time delays the completion by $\frac{1}{\lambda}$;
- running B simultaneously only decrease the required time from A's perspective, which improves the performance due to assumed monotonicity.