Learning-Augmented Algorithms

Mario Lavina

London School of Economics

April 1, 2022

Optimisation under uncertainty

Optimisation under uncertainty

Machine learning: deals with uncertainty by making predictions of the future using past data.

Optimisation under uncertainty

Machine learning: deals with uncertainty by making predictions of the future using past data.

Online algorithms: are algorithms that do not know the full input in advance but are designed to work under the worst-case scenario.

Optimisation under uncertainty

Machine learning: deals with uncertainty by making predictions of the future using past data.

- Adapts to input data

Online algorithms: are algorithms that do not know the full input in advance but are designed to work under the worst-case scenario.

Optimisation under uncertainty

Machine learning: deals with uncertainty by making predictions of the future using past data.

- Adapts to input data

Online algorithms: are algorithms that do not know the full input in advance but are designed to work under the worst-case scenario.

- Robust to outliers

Measuring Performance

Definition (Competitive Ratio)

The Competitive Ratio is the ratio of the cost of the algorithm we are analysing in the worst-case scenario and the optimal cost of the offline algorithm, over all possible inputs.

Ski Rental Problem

Ski Rental Problem

- A skier wants to ski for the length of the ski season $\boldsymbol{x}$.
- He can rent the skis for **1** for one day.
- Or he can buy the skis for a cost of $\boldsymbol{b}$.
- The problem is that we do not know $\boldsymbol{x}$ in advance.
- Following algorithms try to solve the question: When should the skier the buy the skis?

Section 1

Ski Rental without Prediction

Algorithm 1: A Deterministic Algorithm

Algorithm 1 Deterministic algorithm

```
while current day is not equal to b do
    Rent skis
end while
Buy skis
```

Algorithm 1: A Deterministic Algorithm

Algorithm 1 Deterministic algorithm

```
while current day is not equal to b do
    Rent skis
end while
Buy skis
```

Theorem

Competitive Ratio of Algorithm 1 is 2

Algorithm 2: A Randomised Algorithm

Algorithm 2 Randomised algorithm

1: k given from randomised function
2: Rent for $k - 1$ days
3: Buy on kth day

Algorithm 2: A Randomised Algorithm

Algorithm 2 Randomised algorithm

1: k given from randomised function
2: Rent for $k - 1$ days
3: Buy on kth day

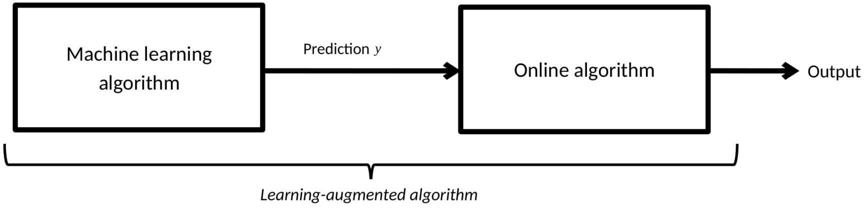
Theorem

Competitive Ratio of Algorithm 2 is $\frac{e}{e-1} \approx 1.58$

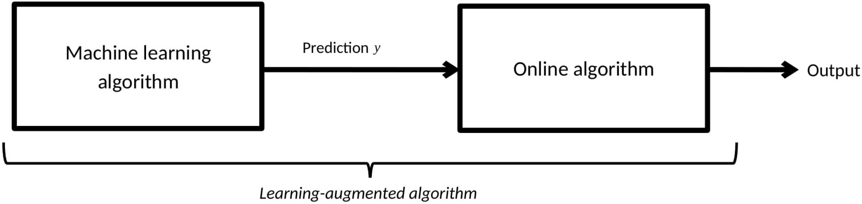
Section 2

Ski Rental with Prediction

Ski Rental with Prediction



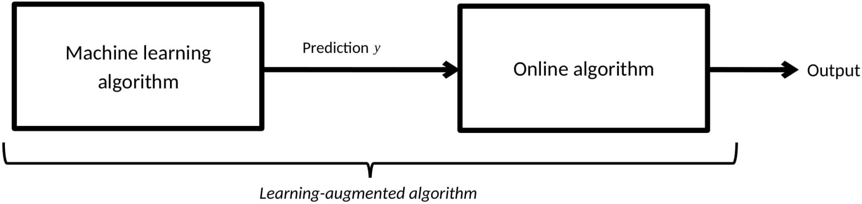
Ski Rental with Prediction



Notation:

- ***y*** = predicted number of days by the machine learning algorithm.

Ski Rental with Prediction



Notation:

- $\boldsymbol{y}$ = predicted number of days by the machine learning algorithm.
- $\eta = |\boldsymbol{x} - \boldsymbol{y}|$ = prediction error.

Properties of learning-augmented algorithms

Properties of learning-augmented algorithms

1. **Independence:** The algorithm should be independent of the machine-learned prediction and make no assumptions about the prediction's error types and distribution.

Properties of learning-augmented algorithms

1. **Independence:** The algorithm should be independent of the machine-learned prediction and make no assumptions about the prediction's error types and distribution.
2. **Consistency:** If the prediction is good, then the algorithm should perform close to the best offline algorithm. We say that an algorithm is $\boldsymbol{\beta}$-consistent if when $\boldsymbol{\eta} = \mathbf{0}$ the Competitive Ratio $= \boldsymbol{\beta}$.

Properties of learning-augmented algorithms

1. **Independence:** The algorithm should be independent of the machine-learned prediction and make no assumptions about the prediction's error types and distribution.
2. **Consistency:** If the prediction is good, then the algorithm should perform close to the best offline algorithm. We say that an algorithm is β-consistent if when $\eta = \mathbf{0}$ the Competitive Ratio $= \beta$.
3. **Robustness:** If the prediction is bad, then the algorithm should perform close to that of an algorithm that does not use predictors. We say that an algorithm is γ-robust if the Competitive Ratio is less than or equal to γ for all possible inputs.

Algorithm 3: A Simple Consistent, Non-Robust Algorithm

Algorithm 3 Simple 1-consistent algorithm

```
if y ≥ b then
    Buy on the first day
else
    Keep renting for all the ski season
end if
```

Algorithm 3: A Simple Consistent, Non-Robust Algorithm

Algorithm 3 Simple 1-consistent algorithm

```
if y ≥ b then
    Buy on the first day
else
    Keep renting for all the ski season
end if
```

Theorem

Algorithm 3 is **1***-consistent but not robust*

Algorithm 4: A Deterministic Robust and Consistent Algorithm

Let $\lambda \in (\mathbf{0}, \mathbf{1})$

Algorithm 4 Deterministic robust and consistent algorithm

if $\boldsymbol{y} \geq \boldsymbol{b}$ **then**
    Buy on day $\lceil \boldsymbol{\lambda b} \rceil$
else
    Buy on day $\lceil \frac{\boldsymbol{b}}{\boldsymbol{\lambda}} \rceil$
end if

Algorithm 4: A Deterministic Robust and Consistent Algorithm

Let $\lambda \in (0, 1)$

Algorithm 4 Deterministic robust and consistent algorithm

1: **if** $y \geq b$ **then**
2: Buy on day $\lceil \lambda b \rceil$
3: **else**
4: Buy on day $\lceil \frac{b}{\lambda} \rceil$
5: **end if**

Theorem

Algorithm 4 is $(1 + \lambda)$-consistent and $\left(\frac{1+\lambda}{\lambda}\right)$-robust.

Algorithm 5: A Randomised Robust and Consistent Algorithm

Algorithm 5 Randomised robust and consistent algorithm

Let $\lambda \in (\frac{1}{b}, 1)$

if $y \geq b$ **then**
  Let $k \leftarrow \lfloor \lambda b \rfloor$
  Define $q_i \leftarrow (\frac{b-1}{b})^{k-i} \frac{1}{b(1-(1-1/b)^k}$ for all $1 \leq i \leq k$
  Choose $j \in \{1, 2, ..., k\}$ randomly from distribution defined by q_i
  Buy on day j
else
  Let $l \leftarrow \lceil \frac{b}{\lambda} \rceil$
  Define $r_i \leftarrow (\frac{b-1}{b})^{l-i} \frac{1}{b(1-(1-1/b)^l}$ for all $1 \leq i \leq l$
  Choose $j \in \{1, 2, ..., l\}$ randomly from distribution defined by r_i
  Buy on day j
end if

Algorithm 5: A Randomised Robust and Consistent Algorithm

Algorithm 5 Randomised robust and consistent algorithm

Let $\lambda \in (\frac{1}{b}, 1)$

if $y \geq b$ **then**
  Let $k \leftarrow \lfloor \lambda b \rfloor$
  Define $q_i \leftarrow (\frac{b-1}{b})^{k-i} \frac{1}{b(1-(1-1/b)^k}$ for all $1 \leq i \leq k$
  Choose $j \in \{1, 2, \dots, k\}$ randomly from distribution defined by q_i
  Buy on day j
else
  Let $l \leftarrow \lceil \frac{b}{\lambda} \rceil$
  Define $r_i \leftarrow (\frac{b-1}{b})^{l-i} \frac{1}{b(1-(1-1/b)^l}$ for all $1 \leq i \leq l$
  Choose $j \in \{1, 2, \dots, l\}$ randomly from distribution defined by r_i
  Buy on day j
end if

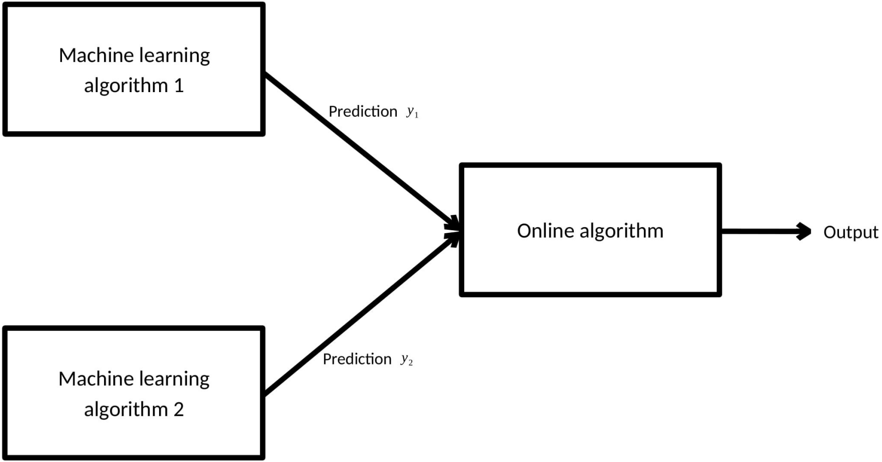
Theorem

Algorithm 5 is $\left(\frac{\lambda}{1-e^{-\lambda}}\right)$*-consistent and* $\left(\frac{1}{1-e^{-(\lambda-\frac{1}{b})}}\right)$*-robust.*

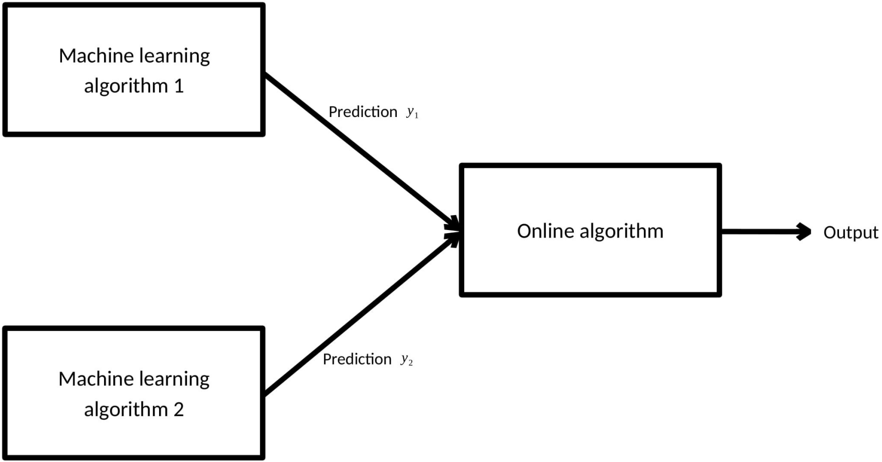
Section 3

Ski Rental with Multiple Predictions

Ski Rental with Multiple Prediction



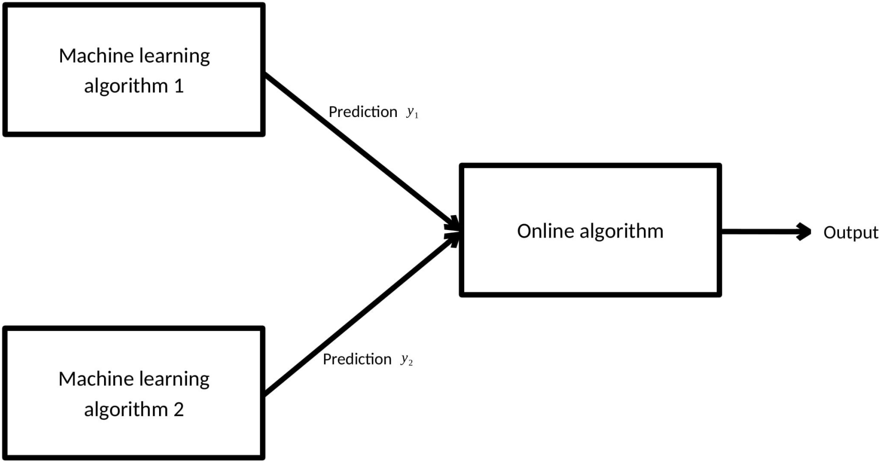
Ski Rental with Multiple Prediction



Notation:

- y_1 is the prediction of machine learning algorithm 1.
- y_2 is the prediction of machine learning algorithm 2.

Ski Rental with Multiple Prediction



Assumption

One of the machine learning algorithms has a prediction error of **0** *meaning it always outputs the correct length of the season.*

Which of the two genies is the correct one?

Which of the two genies is the correct one?

There can be different cases:

1. Machine learning algorithms agree: ($\boldsymbol{y_1} \geq \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$) or ($\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} < \boldsymbol{b}$)

Which of the two genies is the correct one?

There can be different cases:

1. Machine learning algorithms agree: ($\boldsymbol{y_1} \geq \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$) or ($\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} < \boldsymbol{b}$)
2. Machine learning algorithms disagree: $\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$

Which of the two genies is the correct one?

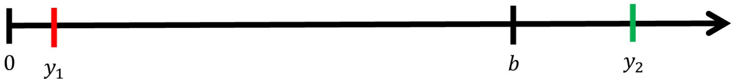
There can be different cases:

1. Machine learning algorithms agree: ($\boldsymbol{y_1} \geq \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$) or ($\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} < \boldsymbol{b}$)
2. Machine learning algorithms disagree: $\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$
   - If we trust machine learning algorithm 1, the worst-case scenario occurs when
     $\boldsymbol{y_1}$ is close to $\boldsymbol{b}$ but $\boldsymbol{y_2}$ is correct. Competitive Ratio $= \frac{\boldsymbol{b}+\boldsymbol{y_1}}{\boldsymbol{b}} \approx \mathbf{2}$.

Which of the two genies is the correct one?

There can be different cases:

1. Machine learning algorithms agree: ($\boldsymbol{y_1} \geq \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$) or ($\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} < \boldsymbol{b}$)
2. Machine learning algorithms disagree: $\boldsymbol{y_1} < \boldsymbol{b}$ and $\boldsymbol{y_2} \geq \boldsymbol{b}$
   - If we trust machine learning algorithm 1, the worst-case scenario occurs when $\boldsymbol{y_1}$ is close to $\boldsymbol{b}$ but $\boldsymbol{y_2}$ is correct. Competitive Ratio $= \frac{\boldsymbol{b}+\boldsymbol{y_1}}{\boldsymbol{b}} \approx \mathbf{2}$.
   - If trust machine learning algorithm 2, the worst-case scenario occurs when $\boldsymbol{y_1}$ is close to $\mathbf{0}$ and $\boldsymbol{y_1}$ is correct. Competitive Ratio $= \frac{\boldsymbol{b}}{\boldsymbol{y_1}}$.



Algorithm 6: A Deterministic Algorithm with Two Predictions

Hence our strategy for this deterministic algorithm is to balance these two Competitive Ratios so that we are covered in extreme situations:

$$\frac{\boldsymbol{b}}{\gamma} = \frac{\boldsymbol{b} + \gamma}{\boldsymbol{b}}$$

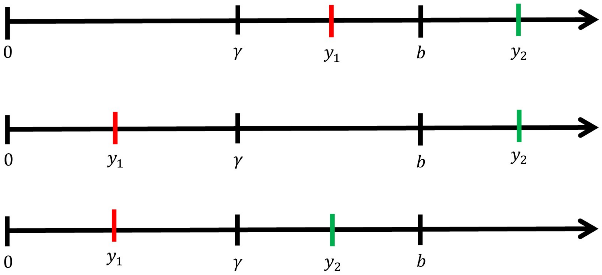
If we solve this equation, we get $\gamma = \frac{-\boldsymbol{b}+\boldsymbol{b}\sqrt{5}}{2}$

Algorithm 6: A Deterministic Algorithm with Two Predictions

Algorithm 6 Deterministic algorithm with 2 machine learning models

$\gamma_0 = 0, \gamma_1 = \frac{-b+b\sqrt{5}}{2}, \gamma_2 = b$

for $i = 1$ to 2 **do**
    if there is no prediction in $[\gamma_{i-1}, \gamma_i)$ **then**
        Rent until γ_{i-1} and buy after γ_{i-1} if the season continues
        break
    end if
end for
Keep renting for all the ski season



Algorithm 6: A Deterministic Algorithm with Two Predictions

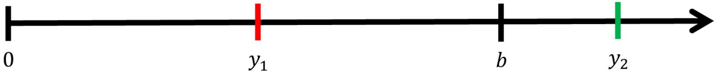
Algorithm 6 Deterministic algorithm with 2 machine learning models

$\gamma_0 = 0, \gamma_1 = \frac{-b+b\sqrt{5}}{2}, \gamma_2 = b$
1: **for** $i = 1$ to 2 **do**
2: **if** there is no prediction in $[\gamma_{i-1}, \gamma_i)$ **then**
3: Rent until γ_{i-1} and buy after γ_{i-1} if the season continues
4: **break**
5: **end if**
6: **end for**
7: Keep renting for all the ski season

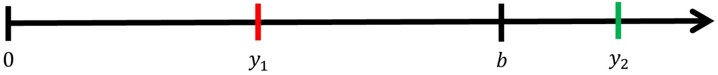
Theorem

Algorithm 6 has a Competitive Ratio of $\frac{1+\sqrt{5}}{2} \approx 1.618$.

Algorithm 7: A Randomised Algorithms with Two Predictions



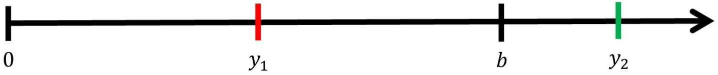
Algorithm 7: A Randomised Algorithms with Two Predictions



Notation:

- Let $\boldsymbol{p}$ be the probability that $\boldsymbol{y_2}$ is correct

Algorithm 7: A Randomised Algorithms with Two Predictions



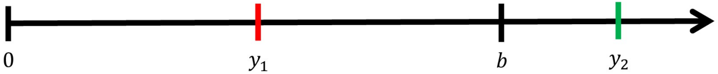
Notation:

- Let p be the probability that y_2 is correct

As before, we are going to balance the two extreme Competitive Ratios so that we are covered in extreme situations

$$\frac{pb + (1 - p)y_1}{y_1} = \frac{pb + (1 - p)(b + y_1)}{b}$$

Algorithm 7: A Randomised Algorithms with Two Predictions



Notation:

- Let p be the probability that y_2 is correct

As before, we are going to balance the two extreme Competitive Ratios so that we are covered in extreme situations

$$\frac{pb + (1 - p)y_1}{y_1} = \frac{pb + (1 - p)(b + y_1)}{b}$$

If we solve this equation, we get $p = \frac{y_1^2}{y_1^2 - y_1 + 1}$ so the Competitive Ratio for this algorithm is $\frac{b}{y_1^2 - y_1 + 1}$. Therefore the Competitive Ratio in the worst-case scenario occurs when $y_1 = \frac{b}{2}$.

Algorithm 7: A Randomised Algorithms with Two Predictions

Algorithm 7 Randomised Algorithm with Multiple Predictions

Let $p = \frac{b^2}{b^2 - 2b + 4}$

if ($y_1 \leq b$ and $y_2 > b$) **then**
    Buy on the first day with probability p.
    Buy on day y_1 with probability $1 - p$.
else if ($y_2 \leq b$ and $y_1 > b$) **then**
    Buy on the first day with probability $1 - p$.
    Buy on day y_2 with probability p.
else if ($y_2, y_1 > b$) **then**
    Buy on the first day.
else
    Keep renting until the ski season ends
end if

Algorithm 7: A Randomised Algorithms with Two Predictions

Algorithm 7 Randomised Algorithm with Multiple Predictions

Let $p = \frac{b^2}{b^2 - 2b + 4}$

```
if (y_1 ≤ b and y_2 > b) then
    Buy on the first day with probability p.
    Buy on day y_1 with probability 1 − p.
else if (y_2 ≤ b and y_1 > b) then
    Buy on the first day with probability 1 − p.
    Buy on day y_2 with probability p.
else if (y_2, y_1 > b) then
    Buy on the first day.
else
    Keep renting until the ski season ends
end if
```

Theorem

Algorithm 7 achieves a Competitive Ratio of $\frac{4}{3}$.

Conclusion and Results

Algorithm	Consistency	Robustness	Competitive Ratio
Algorithm 1			2
Algorithm 2			$\frac{e}{e-1} \approx 1.58$
Algorithm 3	1	Unbounded	
Algorithm 4	$1 < (1+\lambda) < 2$	$2 < \frac{1+\lambda}{\lambda}$	
Algorithm 5	$1 < \frac{\lambda}{1-e^{-\lambda}} < 1.59$	$1.68 < \frac{1}{1-e^{-(\lambda-\frac{1}{b})}}$	
Algorithm 6			$\frac{1+\sqrt{5}}{2} \approx 1.62$
Algorithm 7			$\frac{4}{3} \approx 1.33$