Online Algorithms for Multi-shop Ski Rental with Machine Learned Predictions

Extended Abstract

Shufan Wang
Binghamton University, the State University of New York
Binghamton, New York
swang214@binghamton.edu

Jian Li
Binghamton University, the State University of New York
Binghamton, New York
lij@binghamton.edu

ACM Reference Format:
Shufan Wang and Jian Li. 2020. Online Algorithms for Multi-shop Ski Rental with Machine Learned Predictions. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Uncertainty plays a critical role in many real world applications where the decision maker is faced with multiple alternatives with different costs. These decisions arise in our daily lives, such as whether to rent an apartment or buy a house, which cannot be answered reliably without knowledge of the future. These decision-making problems are usually modeled as online rent-or-buy problems, such as the classical *ski rental* problem [4, 5, 8]. Two paradigms have been widely studied to deal with such uncertainty. On the one hand, online algorithms are designed without prior knowledge to the problem, and *competitive ratio* (CR) is used to characterize the goodness of the algorithm in lack of the future. On the other hand, machine learning is applied to address uncertainty by making future predictions via building robust models on prior data.

Recently, there is a popular trend in the design of online algorithms by incorporating machine learned (ML) predictions to improving their performance [1–3, 6, 7, 9–12]. Two properties are desired: (i) if the predictor is good, the online algorithm should perform close to the best offline algorithm (a design goal called *consistency*); and (ii) if the predictor is bad, the online algorithm should not degrade significantly, i.e., its performance should be close to the online algorithm without predictions (a design goal called *robustness*). Importantly, these properties are achieved under the assumption that the online algorithm has no knowledge about the quality of the predictor or the prediction error types.

The Multi-Shop Ski Rental Problem. While previous studies focused on using ML predictions for a single skier to buy or rent the skis in a single shop, we study the more general setting where the skier has multiple shops to buy or rent the skis with different buying and renting prices. We call this a *multi-shop ski rental* (MSSR) problem. This is often the case in practice, where the skier has to make a *two-fold* decision, i.e, *when and where to buy*, whereas only decision on when to buy is needed in the classical single shop ski rental problem. Specifically, we consider the case that the skier must choose one shop at the beginning of the skiing season, and must buy or rent the skis at that particular shop since then. In other words, *once a shop is chosen by the skier, the only decision variable is when she should buy the skis.* The MSSR not only naturally extends the classical ski rental problem, where a single skier rents or buys the skis in a single shop, but also allows heterogeneity in skier's options. This desirable feature makes the ski rental problem a more general modeling framework for online algorithm design.

Consistency and Robustness. The CR of an online algorithm is defined as the worst-case ratio of the algorithm cost (ALG) to that of the offline optimum (OPT). Inspired by [1, 3, 9, 12], we also use the notions of consistency and robustness to evaluate our algorithms. Let the prediction error be ζ, which is the absolute difference between the prediction and the actual outcome. We say that an online algorithm is α-consistent if ALG $\leq \alpha \cdot$ OPT when the prediction is accurate, i.e., $\zeta = 0$, and β-robust if ALG $\leq \beta \cdot$ OPT for all ζ and feasible outcomes to the problem. Thus consistency characterizes how well the algorithm does in case of perfect predictions, and robustness characterizes how well it does in worst-case predictions.

Main Results. Our main contribution is to develop online algorithms for MSSR with consistency and robustness properties in presence of ML predictions. A hyperparameter $\lambda \in (0, 1)$ is introduced to capture the trust or the quality of ML predictions. With a single ML prediction, we show that if this ML prediction is naively used in algorithm design, the proposed algorithm cannot ensure robustness. We then incorporate ML prediction in a judicious manner by proposing a deterministic and a randomized online algorithm with consistency and robustness guarantee. Greater trust on ML prediction will set λ close to 0 while less trust will set λ close to 1. We numerically evaluate the performance of our online algorithms by investigating impacts of several parameters and provide insights on the benefits of using ML prediction. We also study a more general setting where we get m ML predictions from some ML models. These along with the proofs of main results are available in [13].

2 ONLINE ALGORITHMS FOR MSSR

We consider MSSR with a single ML prediction. We assume there are n shops with buying prices $b_1 > \cdots > b_n$ and renting prices $r_1 < \cdots < r_n$. Let x be the actual number of skiing days which is unknown to the algorithm, and y be the predicted number of skiing days. Then $\zeta = |y - x|$ is the prediction error.

Deterministic Algorithm. We develop a deterministic algorithm by introducing a hyperparameter $\lambda \in (0, 1)$, which gives us a smooth tradeoff between the consistency and robustness of the algorithm.

Theorem 2.1. *The CR of Algorithm 1 is at most* $\min\{(\lambda + 1)r_n + b_1/b_n + \max\{\lambda r_n + 1, \frac{b_1}{b_n} \cdot \frac{1}{1-\lambda}\} \frac{\zeta}{OPT}, \max\{r_n + 1/\lambda, \frac{b_1}{b_n}(1 + 1/\lambda)\}\}$, *where* $\lambda \in (0, 1)$ *is a parameter. In particular, Algorithm 1 is* $((\lambda + 1)r_n + b_1/b_n)$*-consistent and* $(\max\{r_n, b_1/b_n\} + 1/\lambda)$*-robust.*

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

Algorithm 1 A deterministic algorithm

if $y \geq b_n$ **then** Rent until day $\lceil \lambda b_n \rceil - 1$ and buy at shop n
else Rent until day $\lceil b_1/\lambda \rceil - 1$ and buy at shop 1

Remark 1. *The CR is a function of hyperparameter λ and prediction error ζ, which is different from the conventional competitive design. By tuning the λ value, one can achieve different values for CR. The CR might be even worse than online algorithm without predictions for some cases (e.g., prediction error is large). This shows that decision making based on ML prediction comes at the cost of lower worst-case performance guarantee. Finally, it is possible to find the optimal λ to minimize the worst-case CR if the prediction error ζ is known (e.g. from historically observed error values).*

Randomized Algorithm. We consider a class of randomized algorithms for MSSR. Similarly, we consider a hyperparameter λ satisfying $\lambda \in (1/b_n, 1)$. First, we emphasize that a randomized algorithm that naively modifies the distribution used for randomized algorithm design for the classical ski rental algorithm with or without predictions fail to achieve a better consistency and robustness at the same time. We customize the distribution functions carefully by incorporating different renting and buying prices from different shops into the distributions, as summarized in Algorithm 2.

Theorem 2.2. *The CR of Algorithm 2 is at most* $\min\left\{\frac{r_n\lambda}{1-e^{-r_n\lambda}}(1+\frac{\zeta}{OPT}), \frac{b_1}{b_n}\max\left\{\frac{r_n}{1-e^{-r_n(\lambda-1/b_n)}}, \frac{1/\lambda+1/b_1}{1-e^{-1/\lambda}}\right\}\right\}$. *In particular, Algorithm 2 is* $\left(\frac{r_n\lambda}{1-e^{-r_n\lambda}}\right)$*-consistent, and* $\left(\frac{b_1}{b_n}\max\left\{\frac{r_n}{1-e^{-r_n(\lambda-1/b_n)}}, \frac{1/\lambda+1/b_1}{1-e^{-1/\lambda}}\right\}\right)$*-robust.*

Algorithm 2 A randomized algorithm

if $y \geq b_n$ **then** Let $k = \lfloor \lambda b_n \rfloor$
Define $p_i = \left(\frac{b_n - r_n}{b_n}\right)^{k-i} \cdot \frac{r_n}{b_n\left(1-(1-\frac{r_n}{b_n})^k\right)}$, for $i = 1, \cdots, k$
Choose $j \in \{1, 2, ..., k\}$ randomly from the distribution defined by p_i. Rent till day $j-1$ and buy at shop n
else Let $l = \lceil b_1/\lambda \rceil$
Define $q_i = \left(\frac{b_1 - 1}{b_1}\right)^{l-i} \cdot \frac{1}{b_1\left(1-(1-\frac{1}{b_1})^l\right)}$, for $i = 1, \cdots, l$
Choose $j \in \{1, 2, ..., l\}$ randomly from the distribution defined by q_i. Rent till day $j-1$ and buy at shop 1

3 EXPERIMENTS

For all our experiments, we set the number of shops $n = 6$, the buying costs are 100, 95, 90, 85, 80, 75 dollars, and the renting costs 1, 1.05, 1.10, 1.15, 1.20, 1.25 dollars. Note that the actual values of b_i and r_i are not important as we can scale all these values by some constant factors. The actual number of skiing days x is a random variable uniformly drawn from $[1, \Gamma]$, where $\Gamma < \infty$ is a constant. The predicted number of skiing days y is set to $x+\epsilon$ where ϵ is drawn from a normal distribution with mean δ and standard variation σ. We vary either the value of σ from 0 to Γ, or the value of δ to verify the consistency and robustness of our algorithms. To characterize the impact of the hyperparameter λ, we consider the values of 0.25, 0.5, 0.75 and 1 for λ. Note $\lambda = 1$ means that our

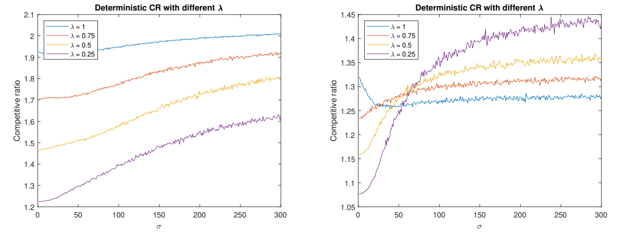

Figure 1: CR of deterministic algorithm. *(Left)* $\Gamma = 3b_1$; *(Right)* $\Gamma = b_1$.

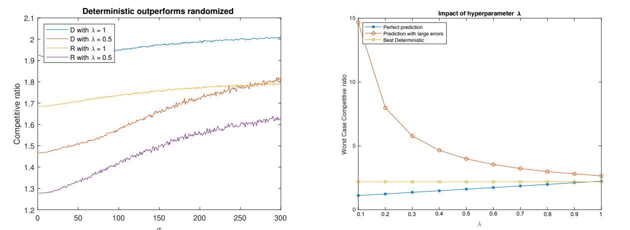

Figure 2: *(Left)* CR of deterministic algorithm vs. randomized algorithm; *(Right)* Impact of hyperparameter.

algorithms ignore the ML prediction, and reduce to the algorithms without prediction. For each value of σ, we plot the average CR by running the corresponding algorithm over 10,000 independent trials. We only present results with unbiased prediction error, and relegate those with biased prediction error to [13].

The Impact of Γ. We consider two possible values of Γ: $\Gamma = 3b_1$ and $\Gamma = b_1$. Since $b_6 = 75$, $\Gamma = 3b_1$ means that it is highly possible the actual number of skiing days x is larger than b_6. Thus according to our algorithm, buying as early as possible will be a better choice, i.e., small λ results in better CR as shown in Figure 1 *(Left)*. On the other hand, with $\Gamma = b_1$, it is highly possible that x is smaller than b_6. Therefore, if the prediction is more accurate (small σ), smaller λ (i.e., more trust on ML prediction) achieves smaller CR, while the prediction is inaccurate (with large σ), larger λ achieves smaller CR. This can be observed from Figure 1 *(Right)*. In particular, with the values of b's and r's in our setting, $\lambda = 1$, i.e., do not trust the prediction achieves the best CR when the prediction error is large.

We further compare the performance of the deterministic and the randomized algorithms, as shown in Figure 2 *(Left)* with $\Gamma = 3b_1$. We make the following observations: (i) With the same prediction errors (e.g., $\lambda = 0.5$), the randomized algorithm always performs better than the deterministic algorithm; (ii) Our deterministic algorithm with ML prediction can beat the performance of classical randomized algorithm without ML predictions when the standard deviation of prediction error is smaller than $2.5b_1 = 250$.

The Impact of Hyperparameter λ. Hyperparameter λ incorporates the trust of ML prediction in online algorithm design. We investigate the impact of λ on the deterministic algorithm by considering a perfect prediction and an extremely erroneous prediction. From Figure 2 *(Right)* with $\Gamma = 3b_1$, we observe (i) With an extremely erroneous prediction, blinding trust the prediction (smaller λ) leads to even worse performance than the online algorithm without ML predictions; (ii) By properly choosing λ, our algorithm achieves better performance than the online algorithm even with extremely erroneous prediction. This demonstrates the importance of hyperparameter λ.

REFERENCES

[1] Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. 2019. Online Computation with Untrusted Advice. *arXiv preprint arXiv:1905.05655* (2019).

[2] Joan Boyar, Lene M Favrholdt, Christian Kudahl, Kim S Larsen, and Jesper W Mikkelsen. 2016. Online algorithms with advice: a survey. *Acm Sigact News* 47, 3 (2016), 93–129.

[3] Sreenivas Gollapudi and Debmalya Panigrahi. 2019. Online Algorithms for Rent-Or-Buy with Expert Advice. In *International Conference on Machine Learning*. 2319–2327.

[4] Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. 1994. Competitive randomized algorithms for nonuniform problems. *Algorithmica* 11, 6 (1994), 542–571.

[5] Ali Khanafer, Murali Kodialam, and Krishna PN Puttaswamy. 2013. The constrained ski-rental problem and its application to online cloud cost optimization. In *2013 Proceedings IEEE INFOCOM*. IEEE, 1492–1500.

[6] Rohan Kodialam. 2019. Optimal Algorithms for Ski Rental with Soft Machine-Learned Predictions. *arXiv preprint arXiv:1903.00092* (2019).

[7] Russell Lee, Mohammad H Hajiesmaili, and Jian Li. 2019. Learning-Assisted Competitive Algorithms for Peak-Aware Energy Scheduling. *arXiv preprint arXiv:1911.07972* (2019).

[8] Zvi Lotker, Boaz Patt-Shamir, and Dror Rawitz. 2008. Rent, Lease or Buy: Randomized Algorithms for Multislope Ski Rental. In *Dans Proceedings of the 25th Annual Symposium on the Theoretical Aspects of Computer Science-STACS 2008, Bordeaux: France (2008)*.

[9] Thodoris Lykouris and Sergei Vassilvtiskii. 2018. Competitive Caching with Machine Learned Advice. In *International Conference on Machine Learning*. 3302–3311.

[10] Andrés Muñoz Medina and Sergei Vassilvitskii. 2017. Revenue optimization with approximate bid predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 1856–1864.

[11] Michael Mitzenmacher. 2018. A model for learned bloom filters and optimizing by sandwiching. In *Advances in Neural Information Processing Systems*. 464–473.

[12] Manish Purohit, Zoya Svitkina, and Ravi Kumar. 2018. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems*. 9661–9670.

[13] Shufan Wang, Jian Li, and Shiqiang Wang. 2020. Online Algorithms for Multi-shop Ski Rental with Machine Learned Predictions. *arXiv preprint arXiv:2002.05808* (2020).