Sequential Ski Rental Problem

Anant Shah
Department of Electrical Engineering
Indian Institute of Technology Madras
anantshah200@gmail.com

Arun Rajkumar
Department of Computer Science and Engineering
Indian Institute of Technology Madras
arunr@cse.iitm.ac.in

ABSTRACT

The classical 'buy or rent' ski-rental problem was recently considered in the setting where multiple experts (such as Machine learning algorithms) advice on the length of the ski season. Here, robust algorithms were developed with improved theoretical performance over adversarial scenarios where such expert predictions were unavailable. We consider a variant of this problem which we call the 'sequential ski-rental' problem. Here, a sequence of ski-rental problems has to be solved in an online fashion where both the buy cost and the length of ski season are unknown to the learner. The learner has access to two sets of experts, one set who advise on the true cost of buying the ski and another set who advise on the length of the ski season. Under certain stochastic assumptions on the experts who predict the buy costs, we develop online algorithms and prove regret bounds for the same. Our experimental evaluations confirm our theoretical results.

KEYWORDS

Online Learning; Hedge Algorithm; Ski-Rental Problem

ACM Reference Format:
Anant Shah and Arun Rajkumar. 2021. Sequential Ski Rental Problem. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

The classic buy or rent ski-rental problem is the following: Given a buy cost b for the ski gear, one needs to make a decision on each day whether to rent the ski for a cost of 1 unit or end the sequence of decisions by buying the ski for a cost of b units. The challenging part of the problem is that one does not have apriori knowledge of how long the ski season would last. The ski-rental problem has found several practical applications including TCP acknowledgement, time scheduling, etc. In the basic version of the problem, a strategy achieving a competitive ratio of 2 is to rent for b days (if the ski season lasts) and buy thereafter. A recent line of work tries to consider the same problem when certain expert advice (potentially machine learning predictions) is available about the length of the ski season. In this scenario, it was shown recently that one can do better when the predictions are within reasonable deviations from the truth. While current work in this area([21], [8]) propose algorithms that require a truthful environment parameter (buy cost), our main contribution is in the case the environment parameter is predicted by a set of experts.

In this work we introduce a variant of the problem which we term the *sequential ski rental problem*. Here, the ski season proceeds in a sequence of rounds and in each round, the player has to come up with a strategy to buy/rent a *perishable* item every day of that season. The item if bought lasts only for that season and becomes unavailable the following seasons. In addition to the unavailability of the length of the season, the player also has to make an *early* rent/buy decision *before* the buy cost is revealed. The problem thus has two levels of uncertainty, in the ski-length and the buy costs respectively.

At first glance, such a problem might seem hopeless to solve as there is no input parameter for the algorithm to work with (i.e., neither the buying price nor the length of the ski season). To tackle this, we consider two types of advice as being available to the player - one on the length of the ski season and other on the buy cost. Both of these could come from potential machine learning prediction models or from certain noisy inputs. In this novel setting, we develop online learning algorithms which have good performance with respect to the worst case bounds. Our algorithm relies on a combination of the stochastic Hedge algorithm and the algorithm for ski-rental problem with expert advice. Our main contributions are as follows

- We introduce the Sequential ski rental problem with two sets of expert advice
- We develop novel algorithms with performance guarantees
- We demonstrate the efficacy of the algorithm on experiments.

Our work can be applied to various real world settings an example of which is the following. Consider a customer on an online retail site, looking to purchase different products in a sequential fashion. For each item she desires, she needs to make a decision on whether to buy or rent the product. The way she decides her strategy is based on a set of experts who are given as input a buy cost and output a rent-buy strategy. The online retail site has a fixed purchase price for each product. However, the customer will need to pay an additional overhead cost which is unknown apriori. A main source of this overhead cost could be delivery charges (or more abstract costs involving time delays in delivery etc). This exact overhead cost is not clear to the customer at the time of deciding her strategy and she only has access to an estimate of this cost. The problem then is how well the customer performs when she is given the estimated cost at the time of deciding her strategy as compared to what she would have done had she been given the exact overall cost to decide her strategy. Clearly since the customer can purchase different items, the buy cost varies.

2 RELATED WORK

Our work is closely related to [8] and [21]. We utilize the concepts of robustness and consistency introduced in [15], who utilized predictions for the online caching problem . [8] consider the setting in

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.
© 2021 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

which multiple experts provide a prediction to the ski-rental problem. They obtain tight bounds for their settings thereby showing the optimal robustness and consistency ratios in the multi-expert setting. However their algorithm depends on the fact that the error in the buy cost is zero, in contrast to our setting where, as mentioned above, we consider that the buy-cost given to the experts may not be true as they are forced to make an early decision.

For the ski-rental problem, [21] utilize predictions from a single expert to improve on the lower bound of $\frac{e}{e-1}$, shown to be optimal by [10]. [21] design a robust, consistent deterministic and randomized algorithm which performs optimally for a zero error prediction, while at the same time matches the optimal lower bound in the case the prediction has infinite error. As was the case in [8], their algorithm depends on the fact that the buy cost is without error. Even for a small perturbation in the buy-cost, the expected algorithmic cost suffered is different. As a part of our online learning subroutine, we propose an algorithm which is robust to small perturbations in the environment parameters, the environment parameter being the buy-cost for the ski-rental problem.

Recent work in online algorithms with machine learned predictions extend to numerous applications such as online scheduling with machine learned advice ([12], [22], [9]) or to the reserve price revenue optimization problem which utilizes predictions of the bids by bounding the gap between the expected bid and the revenue in terms of the average loss of the predictor ([19]).

Our work tackles the problem of uncertainty by utilizing the online learning model. Specifically, we have two sets of experts where the loss value of one set comes from a stochastic distribution. Well studied models to tackle this uncertainty include that of robust optimization ([20]) which gives good guarantees for potential realizations of the inputs. Another model is that of stochastic optimization ([2], [17], [16]) where the input comes from a known distribution. Our work utilizes the multiplicative weight update ([13]), also known as the Constant Hedge algorithm, as our learning algorithm for one set of experts. We utilize the Decreasing Hedge algorithm ([1]) to update the weights of another set of experts whose losses come from a stochastic distribution. Considering this easier setting for learning, ([14], [3], [7], [24], [11], [23], [18]) tackle this by designing algorithms that rely on data dependent tuning of the learning rate or better strategies and give theoretical results on the regret bounds in these settings.

An interpretation of our work is to the classic online learning set-up given in a survey by [25]. Instead of each hypothesis receiving the true environment parameter, based on which they make a recommendation to the learner, they receive an unbiased sample with some noise. Something similar has been studied in the area of differential privacy ([5]) , where certain privacy guarantees are shown if one of the loss vectors in the loss sequence changes. Our model essentially boils down to the scenario where the initial sequence of loss vectors are different but over time they converge to the true loss sequence. We do not show an privacy guarantees but show that our model is a no-regret model.

3 PRELIMINARIES

We consider the online learning model to the ski-rental problem.

Ski-Rental Problem : In the ski-rental problem, an example of a large class rent-or-buy problem, a skier wants to ski and needs to make a decision whether to buy the skis for a cost of b units or rent the skis for a cost of 1 unit per day. The length of the ski season x is *not known*. Trivially, if the ski season lasted more than b days and the skier knew this beforehand, she would buy them at day 1, otherwise she would rent them for all days. The minimum cost that can be suffered is $OPT = \min\{b, x\}$. For this problem, the best a deterministic algorithm can do is obtain a competitive ratio of 2, while [10] designed a randomized algorithm which obtains a competitive ratio of $\frac{e}{e-1}$ which is optimal.

Robustness and Consistency : We utilize the notions of robustness and consistency which are defined when online algorithms utilize machine learned predictions. In online algorithms, the ratio of the algorithmic cost(ALG) to the optimal offline cost(OPT) is defined as the competitive ratio. While utilizing predictions, such a ratio would be a function of the accuracy η of the predictor. Note that the algorithm has no knowledge about the quality of the predictor. An algorithm is α- robust if $\frac{ALG(\eta)}{OPT} \leq \alpha$ for all η and is β-consistent if $\frac{ALG(0)}{OPT} \leq \beta$. The goal is to use the predictions in such a way that if the predictions are true, the algorithm performs close to the offline optimal and even if the prediction is very bad, it performs close to the original online setting without any predictions.

Hedge Algorithm for the expert advice problem : In the classic learning from expert advice setting, also known as decision-theoretic online learning ([6]), the learner maintains a set of weights over the experts and updates these weights according to the losses suffered by the experts. These losses could potentially be chosen in an adversarial manner. Specifically, the learner has weights $\boldsymbol{\alpha^t} = (\alpha_i^t)_{1\leq i \leq M}$ over M experts at time t. The environment chooses a bounded, potentially adversarial, loss vector $\boldsymbol{l^t}$ over these experts at time t. The loss suffered by the learner is $(\boldsymbol{\alpha^t})^T\boldsymbol{l^t}$. The goal of the learner is to compete against the expert with the minimum cumulative loss, i.e to minimize the *regret* which is defined as

$$R_T = \sum_{t=1}^{T} (\boldsymbol{\alpha^t})^T \boldsymbol{l^t} - \min_{i\in[M]} \sum_{t=1}^{T} l_i^t$$

where T denotes the number of instances for which this game is played between the learner and the environment. We say that an algorithm is a no-regret learning algorithm if the regret is sub-linear in T .The multiplicative weights algorithm([13]) updates the weights *optimally* as

$$\alpha_i^t = \frac{e^{-\eta_t \sum_{t'=1}^{t'=t-1} l_i^{t'}}}{\sum_{i=1}^{M} e^{-\eta_t \sum_{t'=1}^{t'=t-1} l_i^{t'}}}$$

where η_t is the learning rate. The learning rate could be constant or dependent on t. The Decrease Hedge algorithm([1]) has a learning rate $\eta_t \propto 1/\sqrt{t}$ while the Constant Hedge algorithm([13]), given a $T \geq 1$, has a learning rate $\eta_t \propto 1/\sqrt{T}$. The standard regret bound for the hedge algorithm (eg : [4]) is sub-linear in T.

We look at the learning from expert advice problem from a different lens. At each stage t, there exists an environment parameter $\mathbf{e}^t$ based on which the experts give their recommendation. This can be the output of some machine learned model which each of the M

experts have, denoted by $(h_i(\mathbf{e}^t))_{1\leq i\leq M}$. A more general view of the problem we tackle in this paper is what happens in the case the experts have access to some estimate $\hat{\mathbf{e}}^{\mathbf{t}} \neq \mathbf{e}^t$ such that $\mathrm{E}[\hat{\mathbf{e}}^{\mathbf{t}}] = \mathbf{e}^t$. The constraint we use on this setting is that the variance of the estimator $\hat{\mathbf{e}^t}$ becomes small as t becomes large.

Stochastic Setting : When the losses are realizations of some unknown random process, we consider it as the stochastic setting. Our work considers the standard i.i.d case where the loss vectors are $\boldsymbol{l}^1, \boldsymbol{l}^2, \ldots, \boldsymbol{l}^t$ which are i.i.d. Note that there need not be independence across experts. We define the sub-optimality gap as $\Delta = \min_{i\neq i^*} \mathrm{E}[l_i^t - l_{i^*}^t]$ where $i^* = argmin_i \mathrm{E}[l_i^t]$. A natural extension is whether the hedge algorithm obtains a better regret guarantee in the nicer stochastic setting. [18] show that for the Decreasing Hedge algorithm, a better regret bound in terms of the sub-optimality gap Δ can be obtained while also showing that for the Constant Hedge algorithm the $\sqrt{T \log M}$ regret bound is the best possible. A part of our setting is inspired by [18] as we have a set of experts whose loss comes from a stochastic distribution.

4 PROBLEM SETTING

In the ski-rental setting, the rental costs are 1-unit per day, b is the buy cost, x is the true number of ski days which is unknown to the algorithm and y is the predicted number of ski days. We use $\eta = |y - x|$ to denote the prediction error. No assumptions are made on how the length of the ski season is predicted. Note that the optimal strategy in hindsight will give us an optimal cost of $OPT = \min\{b, x\}$.

4.1 Sequential Ski Rental Setup

Our learning model has two sets of experts, one set predicting the environment parameters i.e the buy cost of the skis. Let there be m such experts. We will call them as buy-experts from now on. The other set of experts are those that are giving advice to the learner on what strategy to follow based on their prediction of the number of ski-days. They utilize the prediction of the buy-experts to decide their strategy. Let these experts be n in number. We will call them as ski-experts from now on.

We are running multiple ski-rental instances over the time horizon T. For each $t \in [T]$, we denote the ground truth buy cost as b^t and the ground truth ski-days as x^t. The ski experts only make a prediction on x^t,denoted by $\boldsymbol{y}^t = (y_i^t)_{1\leq i\leq n}$,and suggest a strategy to the learner based on the predicted value of the buy experts, which we denote by b_s^t. The way a ski-expert utilizes its prediction y_i^t to suggest a strategy is based on a randomized algorithm, which at a high level suggests to buy very late if $y_i^t \leq b_s^t$ or suggests to buy early if $y_i^t > b_s^t$.

4.2 Buy Expert Predictions

The ground truth buy cost at time t is $b^t \in \mathbb{N}$ .In our model, we consider that each buy cost prediction comes from a stochastic distribution with mean equal to the ground truth buy cost and a certain variance which corresponds to the quality of that buy expert. Let the variance of the best buy expert be γ_{i^*}. We define the sub-optimality gap for buy expert $i \neq i^*$ in terms of the variance as $\Delta_i = \gamma_i - \gamma_{i^*}$ where γ_i corresponds to the variance of the i^{th} buy expert. Let the vector of these predictions be $\boldsymbol{a^t} = (a_i^t)_{1\leq i\leq m}$. In our setting,

$$\boldsymbol{a}^t = b^t\ \vec{1} + \boldsymbol{\epsilon}_{\boldsymbol{b}}^t$$

where $\boldsymbol{\epsilon}_{\boldsymbol{b}}^t$ has zero mean and its covariance matrix is a diagonal matrix. $\boldsymbol{\epsilon}_{\boldsymbol{b}}^t \in [-1, 1]^m$. The algorithm maintains a weight vector $\boldsymbol{\alpha}^t$ over the buy experts corresponding to it's confidence over that particular expert. We use the Decreasing Hedge algorithm to update these weights, where the loss function is the squared error loss

$$\alpha_i^t = \frac{e^{-\eta_t \sum_{t'=1}^{t'=t-1} (a_i^{t'} - b^{t'})^2}}{\sum_{i=1}^{M} e^{-\eta_t \sum_{t'=1}^{t'=t-1} (a_i^{t'} - b^{t'})^2}}$$

The ski-experts are given the buy cost prediction b_s^t on which they base their strategy

$$b_s^t = (\boldsymbol{a}^t)^T \boldsymbol{\alpha}^t$$

In expectation, the prediction given to the ski-experts is equal to the ground truth value.

4.3 Ski Expert Predictions

The ski-experts suggest a strategy to the learner and suffer some loss for the same. They make a prediction on the number of ski days and then suggest a strategy to the learner utilizing the predicted buy cost b_s^t. While we do make an assumption on the prediction distribution, specifically that it is unbiased, of the buy experts, we make no assumption on how the ski expert predictions are obtained. The learner maintains a set of weights $\boldsymbol{\beta}^t$ over these experts which are updated using the Constant Hedge algorithm. The loss suffered by the j^{th} ski-expert at time t is denoted by $l_j^t(b^t, x^t, b_s^t, y_j^t)$. We denote the loss vector suffered by these experts as $\boldsymbol{l}^t(b^t, x^t, b_s^t, \boldsymbol{y}^t)$.

4.4 Regret

Our learning setup is as follows. At each $t \in [T]$ the learner chooses a strategy recommended by a ski-expert by sampling from the distribution $\boldsymbol{\beta}^t$ over these experts and suffers an expected loss. The ski-experts are in turn basing their strategy on a prediction of the buy cost from the buy-experts. At a time t, the loss suffered by the j^{th} ski-expert depends on the ground truth values and the predictions it receives and is hence denoted by $l_j^t(b^t, x^t, b_s^t, y_j^t)$.

The goal of the learner is to compete against the best expert in the case the experts are given the true environment parameters, that being the buy cost in our case. This lends to the expected regret definition which we wish to minimize

$$R_T = \sum_{t=1}^{T} (\boldsymbol{\beta}^t)^T \boldsymbol{l}^t(b^t, x^t, b_s^t, \boldsymbol{y}^t) - \min_j \sum_{t=1}^{T} l_j^t(b^t, x^t, b^t, y_j^t)$$

5 ALGORITHM

We introduce the subroutine used by the ski-experts to compute a strategy, which we call the *CostRobust Randomized Algorithm* and propose the *Sequential Ski Rental* algorithm which the learner utilizes when she has access to two sets of experts.

Symbol	Description
x^t	True number of ski-days at the t^{th} time instant.
y_j^t	j^{th} expert ski-day prediction at time t
b^t	True buy cost at time t
a_i^t	i^{th} expert buy cost prediction at time t
b_s^t	Weighted sum estimate of the buy cost

Table 1: Notation

5.1 CostRobust Randomized Algorithm

Below we present an algorithm which obtains a robust and consistent algorithm for solving the ski-rental problem. The crucial difference between our algorithm and [21] is that our algorithm is robust to small variations in the buy cost i.e if we obtain a noisy sample of the buy cost, our algorithm suffers a cost which is same as the cost suffered if the true value were given. Our algorithm obtains similar consistency and robustness guarantees while considering the competitive ratio along with the added benefit of robustness in terms of the buy cost.

Algorithm 1: CostRobust Randomized Algorithm

```
Function SkiRentStrategy($(b, y)$):
    if $y \geq b$ then
        $k \leftarrow \lfloor \lambda b \rfloor$
        $q_i \leftarrow (1 - \frac{\lambda}{k})^{k-i} . \frac{\lambda}{k(1-(1-\lambda/k)^k)} \forall 1 \leq i \leq k$
        $d \sim \boldsymbol{q}$ // Sample the buy day based on the
          distribution above
        return $d$
    else
        $l \leftarrow \lceil b/\lambda \rceil$
        $r_i \leftarrow (1 - \frac{1}{\lambda l})^{l-i} . \frac{1}{l\lambda(1-(1-\frac{1}{\lambda l})^l)} \forall 1 \leq i \leq l$
        $d \sim \boldsymbol{r}$ // Sample the buy day based on the
          distribution above
        return $d$
    end
```

Let $\lambda \in (1/b, 1)$ be a hyper-parameter. For a chosen λ the algorithm samples a buy day from two different probability distributions depending on the prediction and the input buy cost. Note that the Algorithm 1 outputs a buy day strategy for the prediction on the number of ski days y and the input it gets on the cost b.

We say that the algorithm is ϵ robust in terms of the buy cost if ϵ is the maximum possible value such that when the buy cost prediction b_s lies in the range $(b - \epsilon, b + \epsilon)$, the incurred cost is equal to the cost in the case the true value b were given.

Theorem 1. *The CostRobust randomized algorithm is ϵ robust in terms of the buy cost where ϵ is*

$$\epsilon = \min\left(\frac{1}{\lambda}\min(\{\lambda b\}, 1 - \{\lambda b\}), \lambda \min\left(\left\{\frac{b}{\lambda}\right\}, 1 - \left\{\frac{b}{\lambda}\right\}\right)\right)$$

where $\{x\}$ denotes the fractional part of x.

Proof. Consider the case when $y \geq b$. In this case $k = \lfloor \lambda b \rfloor$. A predicted buy cost b_s is given where $b - \epsilon < b_s < b + \epsilon$, and hence the condition we get on ϵ so that $\lfloor \lambda b \rfloor = \lfloor \lambda b_s \rfloor$ is

$$\epsilon = \frac{1}{\lambda}\min(\{\lambda b\}, 1 - \{\lambda b\})$$

Similarly, in the case $y \leq b$, performing a similar analysis where we require the l values to be equal gives us the condition

$$\epsilon = \lambda \min\left(\left\{\frac{b}{\lambda}\right\}, 1 - \left\{\frac{b}{\lambda}\right\}\right)$$

Hence the result follows.

□

Note that the robustness with respect to the buy cost depends on the λ hyper-parameter. We now show consistency and robustness guarantees, in terms of the competitive ratio, of our proposed algorithm in the case it receives the true prediction of the buy cost. Our analysis is similar to [21], who calculate the expected loss of the algorithm based on the relative values of y, b, x and the distribution defined.

Theorem 2. *The CostRobust randomized ski-rental algorithm yields a competitive ratio of at most* $\min\{\frac{1+1/\lfloor \lambda b \rfloor}{1-e^{-\lambda}}, (\frac{\lambda}{1-e^{-\lambda}})(1+\frac{\eta}{OPT})\}$ *where λ is a hyper-parameter chosen from the set $\lambda \in (1/b, 1]$. The CostRobust Randomized algorithm is $\frac{1+1/\lfloor \lambda b \rfloor}{1-e^{-\lambda}}$-robust and $(\frac{\lambda}{1-e^{-\lambda}})$-consistent.*

Proof. We consider different cases depending on the values of y, b, x, k.

- $y \geq b, x \geq k$ Based on the algorithm $k = \lfloor \lambda b \rfloor$.

$$\begin{aligned}\mathbf{E}[ALG] &= \sum_{i=1}^{k}(b + i - 1)q_i \leq b - \frac{k}{\lambda} + \frac{k}{1 - (1 - \frac{\lambda}{k})^k} \\ &\leq b - \frac{k}{\lambda} + \frac{k}{1 - e^{-\lambda}} \leq b + b\left(\frac{\lambda}{1 - e^{-\lambda}} - 1\right) \\ &\leq \left(\frac{\lambda}{1 - e^{-\lambda}}\right)(OPT + \eta)\end{aligned}$$

  where the second to last inequality from the fact $b > \frac{k}{\lambda}$ and the last inequality from the fact $y > b$.
- $y \geq b, x < k$. For this ordering of the variables $OPT = x$. The algorithm suffers a loss of $ALG = b + i - 1$ if it buys at the beginning of day $i \leq x$. Thus ,

$$\begin{aligned}\mathbf{E}[ALG] &= \sum_{i=1}^{x}(b + i - 1)q_i + \sum_{i=x+1}^{k} xq_i \\ &= \frac{x}{1 - (1 - \frac{\lambda}{k})^k}\left(1 + \left(1 - \left(1 - \frac{\lambda}{k}\right)^x\right)\left(1 - \frac{\lambda}{k}\right)^{k-x}\frac{(b - \frac{k}{\lambda})}{x}\right) \\ &\leq \left(\frac{1}{1 - e^{-\lambda}}\left(1 + \left(\frac{1}{\lambda} - \frac{1}{k}\right)\frac{1 - (1 - \frac{\lambda}{k})^{k-1}}{k - 1}\right)\right)OPT \\ &\leq \left(\frac{1 + 1/\lfloor \lambda b \rfloor}{1 - e^{-\lambda}}\right)OPT\end{aligned}$$

where the second to last inequality follows from the fact that the function $(1-(1-\frac{\lambda}{k})^x)(1-\frac{\lambda}{k})^{k-x}\frac{1}{x}$ is maximized at $x = k-1$ and the last inequality is valid because $k \geq 2$ as in the worst case $x = 1$. To show consistency, we have the inequality

$$
\begin{aligned}
\mathbf{E}[ALG] &\leq b - \frac{k}{\lambda} + \frac{x}{1-e^{-\lambda}} \leq \frac{\{\lambda b\}}{\lambda} + \frac{x}{1-e^{-\lambda}} \\
&\leq \frac{\{\lambda b\} + x}{1-e^{-\lambda}} \leq \frac{\lambda b}{1-e^{-\lambda}} \\
&\leq \left(\frac{\lambda}{1-e^{-\lambda}}\right)(OPT + \eta)
\end{aligned}
$$

where the third inequality comes from the fact that $\lambda \geq 1-e^{-\lambda}$ for all $\lambda \in [0,1]$ and the fourth inequality comes from the fact that $x < k$.

- $y < b, x < l$. Based on the algorithm, $l = \left\lceil \frac{b}{\lambda} \right\rceil$. The algorithm suffers a loss of $ALG = b + i - 1$ if it buys at the beginning of day $i \leq x$. Thus ,

$$
\begin{aligned}
\mathbf{E}[ALG] &= \sum_{i=1}^{x}(b+i-1)r_i + \sum_{i=x+1}^{l} x r_i \leq (b-\lambda l) + \frac{x}{1-(1-\frac{1}{\lambda l})^l} \\
&\leq \frac{x}{1-e^{-\frac{1}{\lambda}}} \leq \left(\frac{1}{1-e^{-1/\lambda}}\right)(OPT+\eta) \\
&\leq \left(\frac{\lambda}{1-e^{-\lambda}}\right)(OPT+\eta)
\end{aligned}
$$

- $y < b, x \geq l$. Here $OPT = b$. The expected cost incurred is

$$
\begin{aligned}
\mathbf{E}[ALG] &= \sum_{i=1}^{l}(b+i-1)r_i \leq b - \lambda l + \frac{l}{1-(1-\frac{1}{\lambda l})^l} \\
&\leq b + l\left(\frac{1}{1-e^{-1/\lambda}} - \lambda\right) \leq b + l\left(\frac{\lambda e^{-\lambda}}{1-e^{-\lambda}}\right) \\
&\leq \left(\frac{1+\lambda e^{-\lambda}/b}{1-e^{-\lambda}}\right) OPT < \frac{(1+1/\lfloor \lambda b \rfloor)}{1-e^{-\lambda}} OPT
\end{aligned}
$$

which shows some sense of robustness. To show consistency, we can write the equations as

$$
\begin{aligned}
\mathbf{E}[ALG] &\leq \frac{l}{1-e^{-1/\lambda}} = \frac{1}{1-e^{-1/\lambda}}(b+l-b) \\
&\leq \frac{1}{1-e^{-1/\lambda}}(OPT+\eta) \leq \left(\frac{\lambda}{1-e^{-\lambda}}\right)(OPT+\eta)
\end{aligned}
$$

Hence the result follows.

□

This result provides a trade-off between the consistency and robustness ratios. Setting $\lambda = 1$ gives us a guarantee that even if the prediction has a very large error($\eta \to \infty$), our competitive ratio is bounded by $\frac{e}{e-1}(1+1/b)$ which is close to the best case theoretical bound without predictions. However if we are very confident in the prediction, i.e confident that $\eta = 0$, then we can set λ to be very small and get a guarantee of performing close to the offline optimal.

5.2 Sequential Ski Rental Algorithm

We utilize the CostRobust algorithm as a subroutine for each ski-expert. Each of the n ski experts are running the algorithm for each instance of the ski-rental problem to determine a strategy using the predicted buy cost and its own prediction on the number of ski days. The loss is calculated with respect to the best hindsight strategy and hence is always positive. We normalize the competitive ratio by an additive factor so that if an expert predicts correctly, she obtains 0 loss.

Algorithm 2: Loss calculation for each expert

```
Function loss($b^t, x^t, b_s^t, y_j^t$):
    $d \leftarrow$ SkiRentStrategy($b_s^t, y_j^t$)
    if $x^t \geq b^t$ then
        $OPT = b^t$
    else
        $OPT = x^t$
    end
    Based on the strategy suggested by the expert in
      Algorithm 1, $d$ is the buy-day.
    if $x^t \geq d$ then
        $ALG = b^t + d - 1$
    else
        $ALG = x^t$
    end
    $l_j^t = \frac{ALG-OPT}{OPT}$ // Loss suffered by the $j^{th}$ ski expert at
      the $t^{th}$ iteration
```

At each instance of the *Sequential Ski Rental Algorithm*, the input to the *CostRobust* subroutine is an estimate $b_s^t \neq b^t$. The statement below shows that even if the ski-days prediction of a ski-expert has very large error, it suffers a finite loss when it uses an estimate of the buy cost.

Theorem 3. *The loss suffered by each ski expert is bounded for every round $t \in [T]$.*

We will denote this bound by B. The proof of this claim is given in the long version of this paper. The sketch of the proof is similar to the previous claim where we consider different cases based on the relative values of b^t, x^t, b_s^t, y_j^t and show that the loss suffered by a ski-expert is at most an extra additive factor which is bounded when the buy predictions are coming from bounded random variables.

We now describe the setting of the online learning algorithm. The buy predictions comes from unbiased experts. The assumption we make is that the buy predictions come from a bounded stochastic distribution which is i.i.d over rounds. The prediction given to the ski experts is a weighted sum of each buy expert prediction. The ski experts utilize this to recommend a strategy and suffer some loss for the same. The buy expert weights are updated using the Decreasing Hedge algorithm while the weights of the ski experts are updated using the Constant Hedge algorithm.

6 REGRET ANALYSIS

In this section, we show our main result - a regret guarantee for the proposed Sequential Ski Rental algorithm.

```
Algorithm 3: Sequential Ski Rental Algorithm
  Input:
    $\lambda$ : Hyperparamter ;
  Initialization:
  $\boldsymbol{w}^1_{\boldsymbol{\beta}} \leftarrow (1, 1, \ldots, 1)$ : Weights corresponding to the ski experts ;
  $\boldsymbol{w}^1_{\boldsymbol{\alpha}} \leftarrow (1, 1, \ldots, 1)$ : Weights corresponding to the buy experts ;
  for $t \leftarrow 1$ to $T$ do
      input $\boldsymbol{a^t}$ : Buy expert predictions ;
      $\boldsymbol{\alpha^t} = \frac{\boldsymbol{w}^t_{\boldsymbol{\alpha}}}{\sum_i (\boldsymbol{w}^t_{\boldsymbol{\alpha}})_i}$ : Probability distribution over the buy experts ;
      $b^t_s \leftarrow \boldsymbol{a^t} \cdot \boldsymbol{\alpha^t}$ : The weighted buy cost prediction to be provided to the ski experts ;
      input $\boldsymbol{y^t}$ : Ski-expert predictions ;
      $\boldsymbol{\beta^t} \leftarrow \frac{\boldsymbol{w}^t_{\boldsymbol{\beta}}}{\sum_j (\boldsymbol{w}^t_{\boldsymbol{\beta}})_j}$ : Probability distribution over the ski experts ;
      $\boldsymbol{l^t} \leftarrow loss(b^t, x^t, b^t_s, \boldsymbol{y^t})$ : Loss vector where $j^{th}$ element corresponds to the loss of the $j^{th}$ ski expert ;
      $\boldsymbol{w}^{t+1}_{\boldsymbol{\beta}} \leftarrow \boldsymbol{w}^t_{\boldsymbol{\beta}} e^{-\epsilon_s \boldsymbol{l^t}}$ : Update the weights according to the Constant hedge algorithm ;
      $\boldsymbol{w}^{t+1}_{\boldsymbol{\alpha}} \leftarrow \boldsymbol{w}^t_{\boldsymbol{\alpha}} e^{-\epsilon_b (\boldsymbol{a^t} - b^t)^2}$ : Update the weights according to the Decreasing Hedge algorithm
  end
```

Theorem 4. *Let the variance of the best buy expert satisfy*

$$\gamma_{min} = \frac{\delta\epsilon^2}{Tc}$$

for some $c \in (1, \infty)$ and the time horizon

$$T > \frac{1}{2\Delta^2 \log m} \log^2 \left(\frac{2T\Delta mc}{\delta\epsilon^2 (c-1)} \right).$$

Then, with probability at least $1 - \delta$, the cumulative regret of the Sequential Ski Rental algorithm is bounded as

$$R_T \le (1 + B^2)\sqrt{T \log n} + B \frac{1}{2\Delta^2 \log m} \log^2 \left(\frac{2T\Delta mc}{\delta\epsilon^2 (c-1)} \right)$$

where n are the number of ski-experts, where B is the bound on the loss suffered by the ski-experts, Δ is the minimum sub-optimality gap of the buy experts in terms of their variance, $m \ge 2$ are the number of buy experts and ϵ is the robustness in terms of the buy cost of the CostRobust Randomized algorithm.

Recalling the regret definition we use, we have

$$R_T = \sum_{t=1}^{T} (\boldsymbol{\beta^t})^T \boldsymbol{l^t}(b^t, x^t, b^t_s, \boldsymbol{y^t}) - \min_j \sum_{t=1}^{T} l^t_j (b^t, x^t, b^t, y^t_j)$$

This can be split w.r.t the optimal ski expert j^*, given the true value as $R_T = R^x_T + R^b_T$ where each component is defined as

$$R^x_T = \sum_{t=1}^{T} (\boldsymbol{\beta^t})^T \boldsymbol{l^t}(b^t, x^t, b^t_s, \boldsymbol{y^t}) - \sum_{t=1}^{T} l^t_{j^*}(b^t, x^t, b^t_s, y^t_j)$$

and

$$R^b_T = \sum_{t=1}^{T} l^t_{j^*}(b^t, x^t, b^t_s, y^t_j) - \sum_{t=1}^{T} l^t_{j^*}(b^t, x^t, b^t, y^t_j)$$

Theorem 5. *The first term in the regret split R^x_T is bounded by*

$$R^x_T \le (1 + B^2)\sqrt{T \log n}$$

where B is the bound on the loss suffered by the ski-experts and n denotes the number of ski-experts.

This regret bound R^x_T follows from the standard regret bound for the Constant Hedge algorithm with n experts when losses for each of these experts lie in the range $[0, B]$.

To bound R^b_T, note that the loss function is ϵ-robust to the buy cost, hence if the predicted buy cost lies in the range $(b - \epsilon, b + \epsilon)$, this would imply that $l^t_{j^*}(b^t, x^t, b^t_s, y^t_j) = l^t_{j^*}(b^t, x^t, b^t, y^t_j)$.

Let us analyze the predicted buy cost b^t_s. Note that $\mathbf{E}[b^t_s | \boldsymbol{\alpha^t}] = b^t$. This is because each of the buy experts in expectation predict correctly and $\sum_{i=1}^{m} \alpha^t_i = 1$. This leads to $\mathbf{E}[b^t_s] = b^t$. Let γ_i denote the variance of the i^{th} buy expert. Now to find the variance of b^t_s.

$$\mathbf{E}[(b^t_s - b^t)^2 | \boldsymbol{\alpha^t}] = \mathbf{E}[(\sum_{i=1}^{m} \alpha^t_i (a^t_i - b^t))^2 | \boldsymbol{\alpha^t}] = \sum_{i=1}^{m} (\alpha^t_i)^2 \gamma_i$$

where the first equality comes from the fact that $\sum_i \alpha^t_i = 1$, the second equality from the fact that agents predicting at time t are independent and are predicting with mean b^t. Thus $\mathbf{E}[(b^t_s - b^t)^2] = \sum_{i=1}^{m} \gamma_i \mathbf{E}[(\alpha^t_i)^2]$. Now using Chebyshev inequality, the probability that b^t_s lies in the ϵ range about b^t is given as

$$Pr[|b^t_s - b^t| \le \epsilon] > 1 - \frac{var(b^t_s)}{\epsilon^2}$$

For this event to hold with probability at least $1 - \delta$, we require $\sum_{i=1}^{m} \gamma_i \mathbf{E}[(\alpha^t_i)^2] \le \delta\epsilon^2$. Under the assumption that one buy expert has variance $\gamma_{min} << \delta\epsilon^2$ this is trivially satisfied as $t \longrightarrow \infty$ .Hence we now require a minimum $t^* < T$ such that the above event is satisfied for all $t \in [t^*, T]$.

Theorem 6. *With probability at least $1 - \delta$, the number of rounds t^* needed so that*

$$\sum_{t=t^*}^{T} l_{j^*}(b^t, x^t, b^t_s, y^t_j) = \sum_{t=t^*}^{T} l_{j^*}(b^t, x^t, b^t, y^t_j) \text{ is}$$

$$t^* \le \frac{1}{2\Delta^2 \log m} \log^2 \left(\frac{2T\Delta mc}{\delta\epsilon^2 (c-1)} \right)$$

under the assumption that the variance of one buy expert γ_{min} satisfies $\gamma_{min} = \frac{\delta\epsilon^2}{Tc}$ for some $c \in (1, \infty)$, where $m \ge 2$ are the number of buy-experts and ϵ is the robustness of the CostRobust algorithm in terms of the buy cost and the time horizon $T > \frac{1}{2\Delta^2 \log m} \log^2 (\frac{2T\Delta mc}{\delta\epsilon^2 (c-1)})$.

PROOF. Consider a time $t_0 = \left\lceil \frac{4}{\Delta^2} \right\rceil$. If the rate of convergence to the desired variance is upper bounded by t_0, then we have a convergence rate which does not depend on T. If the time taken for convergence is greater than t_0, then we consider the analysis below. The variance of b_s^t is

$$var(b_s^t) = \sum_{i=1}^{m} \gamma_i \mathbf{E}[(\alpha_i^t)^2]$$

The update at time t for each buy expert is made based on the squared error, specifically $(a_i^t - b^t)^2$. We define the loss suffered by the i^{th} buy expert at time t as $g_i^t = (a_i^t - b^t)^2$. For every buy expert $i \neq i^*$, we define the variable $Z_i^t := -g_i^t + g_{i^*}^t + \Delta_i$, which belong to $[-1 + \Delta_i, 1 + \Delta_i]$. We define G_i^t as the cumulative loss for buy expert i upto time t i.e $G_i^t = \sum_{s=1}^{t} (a_i^t - b^t)^2$. Applying Hoeffding's inequality, we get

$$\begin{aligned} Pr(G_i^{t-1} - G_{i^*}^{t-1} < \Delta_i \frac{t-1}{2}) &= Pr(\sum_{s=1}^{t-1} Z_i^s > \Delta_i \frac{t-1}{2}) \\ &\le e^{-(t-1)\frac{\Delta_i^2}{8}} \end{aligned}$$

When $G_i^{t-1} - G_{i^*}^{t-1} > \Delta_i \frac{t-1}{2}$, then

$$\begin{aligned} \alpha_i^t &= \frac{e^{-\eta_b (G_i^{t-1} - G_{i^*}^{t-1})}}{1 + \sum_{j \neq i^*} e^{-\eta_b (G_j^{t-1} - G_{i^*}^{t-1})}} \\ &\le e^{-\Delta_i \sqrt{(t-1)(\log m)/2}} \end{aligned}$$

since $t \ge t_0 + 1 \ge 2$ .Thus $(\alpha_i^t)^2 \le e^{-\Delta_i \sqrt{2(t-1)(\log m)}}$. Hence

$$\begin{aligned} \mathbf{E}[(\alpha_i^t)^2] &\le Pr(G_i^{t-1} - G_{i^*}^{t-1} < \Delta_i \frac{t-1}{2}) + e^{-\Delta_i \sqrt{2(t-1)(\log m)}} \\ &\le e^{-(t-1)\frac{\Delta_i^2}{8}} + e^{-\Delta_i \sqrt{2(t-1)(\log m)}} \end{aligned}$$

Let us consider each component separately. Considering the first term, since $\Delta_i \ge \Delta$ we obtain

$$\Delta_i e^{-(t-1)\frac{\Delta_i^2}{8}} \le \Delta e^{-(t-1)\frac{\Delta^2}{8}}$$

when $\frac{\Delta\sqrt{t-1}}{2} \ge 1$ i.e $t \ge 1 + \frac{4}{\Delta^2}$ which is satisfied as $t \ge t_0 + 1 \ge 1 + \frac{4}{\Delta^2}$. Now considering the second component

$$\Delta_i e^{-\Delta_i \sqrt{2(t-1)(\log m)}} \le \Delta e^{-\Delta \sqrt{2(t-1)(\log m)}}$$

is satisfied if $\Delta\sqrt{2(t-1)(\log m)} \ge 1$ i.e $t \ge 1 + \frac{1}{2\Delta^2 \log m}$ which is again ensured by $t \ge t_0 + 1$.

Also,

$$e^{-(t-1)\Delta_i^2/8} \le e^{-(t-1)\Delta^2/8}$$

and

$$e^{-\Delta_i \sqrt{2(t-1)(\log m)}} \le e^{-\Delta \sqrt{2(t-1)(\log m)}}$$

where both these inequalities come from the fact $\Delta_i \ge \Delta$.Now, rewriting the variance in terms of the sub-optimality parameters

$$\begin{aligned} var(b_s^t) &= \sum_{i=1}^{m} \gamma_i \mathbf{E}[(\alpha_i^t)^2] \\ &\le \gamma_{min} + \gamma_{min} \sum_{i \neq i^*} \mathbf{E}[(\alpha_i^t)^2] + \sum_{i \neq i^*} \Delta_i \mathbf{E}[(\alpha_i^t)^2] \end{aligned}$$

Hence for every $t \ge t_0 + 1$, we get

$$var(b_s^t) \le \gamma_{min} + m(\gamma_{min} + \Delta)(e^{-\Delta\sqrt{2(t-1)(\log m)}} + e^{-(t-1)\frac{\Delta^2}{8}})$$

This gives us the condition that we must require $\gamma_{min} < \delta\epsilon^2$ or $\gamma_{min} = \frac{\delta\epsilon^2}{c}$ for some $c \in (1, \infty)$. Thus the convergence rate boils down to finding a minimum $t > t_0$ such that

$$e^{-\Delta\sqrt{2(t-1)(\log m)}} + e^{-(t-1)\frac{\Delta^2}{8}} \le \frac{\delta\epsilon^2 - \gamma_{min}}{m(\gamma_{min} + \Delta)}$$

Under the assumption $\Delta >> \gamma_{min}$, which is justifiable as γ_{min} is required to be small, an upper bound on the convergence would be when each of the components is less than $\frac{\delta\epsilon^2(c-1)}{2m\Delta c}$ for some $c \in (1, \infty)$. Now since in every round from t^* to T the desired event holds, the required equality holds with probability at least $(1-\delta)^{T-t^*}$. At least because the probability of the desired event is only going to increase in rounds t^* to T. Thus the required equality holds with probability $(1-\delta)^{T-t^*} > (1-\delta)^T \approx (1 - T\delta)$. Hence replacing δ with δ/T gives us the result.

□

COROLLARY 7. *With probability at least* $1 - \delta$, *the second term in the regret split* R_T^b *is bounded as*

$$R_T^b \le B \frac{1}{2\Delta^2 \log m} \log^2 \left(\frac{2T\Delta mc}{\delta\epsilon^2(c-1)} \right)$$

where B *is the bound on the loss suffered by the ski experts and* $\gamma_{min} = \frac{\delta\epsilon^2}{Tc}$ *for some* $c \in (1, \infty)$.

Hence the regret bound follows using the above proved results.

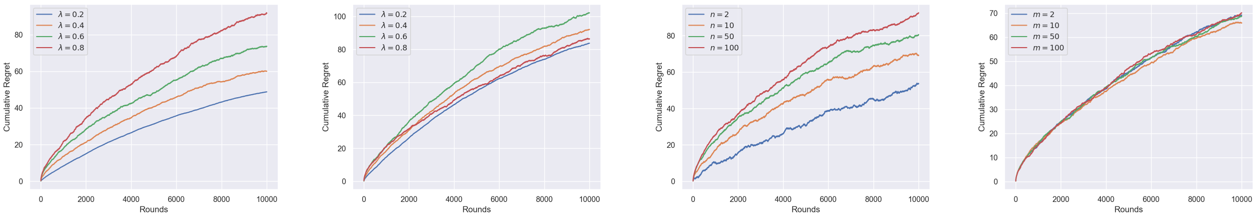
7 EXPERIMENTS

We perform empirical studies to show that our CostRobust algorithm performs similarly to the algorithm proposed by [21]. We also perform simulations of the *Sequential Ski Rental Algorithm* to verify our theoretical regret guarantees.

7.1 CostRobust Randomized Algorithm

To show a comparison, we set the cost of buying to $b = 100$ and sample the actual number of ski days x uniformly as an integer from $[1, 4b]$. To obtain the predicted number of ski-days, we model it as $y = x + \epsilon$, where the noise ϵ is drawn from a normal distribution with mean 0 and standard deviation σ. We compare our algorithms for two different values of the trade-off parameter λ. For each σ, we plot the average competitive ratio obtained by each algorithm over 10000 independent trials.

Figure 2 shows that both the algorithms perform similarly in terms of the competitive ratio. Setting $\lambda = 1$, both algorithms ignore the prediction and guarantee a robustness which are close to the theoretical lower bound of $\frac{e}{e-1}$. Setting $\lambda = \ln\frac{3}{2}$ guarantees an



(a) $\eta_{min} = 1$,$\eta_{max} = 100$ **(b)** $\eta_{min} = 100$,$\eta_{max} = 150$ **(c)** $\eta_{min} = 1$,$\eta_{max} = 50$ **(d)** $\eta_{min} = 1$, $\eta_{max} = 50$

Figure 1: ((a) and (b)) Regret Variation as a function of the hyper-parameter λ; ((c) and (d)) Regret Variation as a function of the number of ski experts n and the number of buy experts m

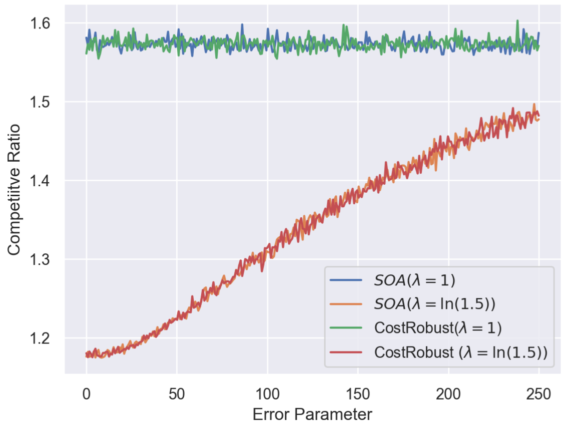


Figure 2: Comparison of the modified randomized algorithm with the randomized algorithm proposed by [21]

upper bound close to 3 for the CostRobust algorithm. We observe that for such a λ, the algorithm performs much better than the classical guarantees.

7.2 Regret Experiments

While we require certain assumptions to get a theoretical bound, our empirical study shows that we obtain a vanishing regret for much weaker conditions. We obtain regret plots for different settings of λ, n and m. For all $t \in [1, T]$, b^t and x^t is a uniformly sampled integer from $[200, 700]$. We consider such a range as some experts could have really large errors in predictions which would lead to a negative prediction in the case the bound on the support was lower. We consider m buy-experts and n ski-experts. The learning rate is set according to the Decreasing Hedge algorithm and Constant Hedge algorithm the sets of experts. In our empirical study, the prediction of each buy expert is $a_i^t = b^t + \epsilon_b$, where ϵ_b is drawn from a truncated normal distribution in the range $[-50, 50]$ with mean 0 and variance γ_i. For the m buy experts, their variance takes values at uniform intervals from the range $[\gamma_{min}, \gamma_{max}]$. Our empirical study uses $\gamma_{min} = 1$ and $\gamma_{max} = 20$. Note that even though our theoretical bound holds when the noise comes from the range $[-1, 1]$ our empirical study shows us that we can achieve vanishing regret for a much weaker constraint. As a modelling choice, we use predictions on the length of the ski season from a normal distribution. The prediction of each ski expert is $y_j^t = x^t + \epsilon_x$, where ϵ_x is drawn from a normal distribution with mean 0 and variance η_j. For the n ski experts, their variance takes values at uniform intervals from the range $[\eta_{min}, \eta_{max}]$.The regret plots are obtained over 100 trials.

7.2.1 Variation in λ. We expect that if there is a "good" ski expert(a ski expert with less error), then using a lower value of λ will give us less algorithmic cost due to the consistency result derived above. However if we do not know the quality of the experts(worst case all of them are bad), the algorithmic cost and hence regret is bounded. The variation is shown in part (a) and (b) Figure 1.

7.2.2 Ski Experts Variation. What the variation in the number of experts shows us is that if we have a few experts at our disposal, making an early decision might be as good as making a decision when the experts have access to the true parameters if not better. An intuition for the learner performing better in the presence of noise is the following situation. Consider the case where $b^t < x^t$. The optimal strategy is to buy early. If the ski expert predictions are less than b^t they would predict sub-optimally when given the true buy cost. If they receive a buy cost sample such that it is less than all of their predictions, then the learner performs better with the noisy sample. However the probability of this decreases as the number of ski-experts increase as all of their predictions need to satisfy this condition. The variation is shown in part (c) of Figure 1.

7.2.3 Buy Experts Variation. The number of buy experts does not affect the cumulative regret as long as the best buy expert comes from a similar error range. This is because the way the learner updates the weights of these experts is based on how far it is from the true buy cost at that time instant. The variation is shown in part (d) of Figure 1.

8 CONCLUSION

In this work, we introduced the sequential ski rental problem, a novel variant of the classical ski buy or rent problem. We developed algorithms and proved regret bounds for the same. Currently we assume that the buy costs are stochastic with different variances. Future work includes considering more general buy cost advice.

REFERENCES

[1] Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. 2002. Adaptive and self-confident on-line learnng algorithms. *Journal of Computer and System Sciences, 64(1):48–75* (2002).

[2] Sebastien Bubeck and Alexsandrs Slivkins. 2012. The best of both worlds : Stochastic and adversarial bandits. *In COLT, pages 42.1–42.23* (2012).

[3] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. 2007. Improved second-order bounds for prediction with expert advice. *Machine Learning, 66:321–352* (2007).

[4] Alexey Chernov and Fedor Zhdanov. 2010. Prediction with expert advice under discounted loss. *International Conference on Algorithmic Learning Theory(ALT), pages 255-269* (2010).

[5] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. 2010. Differential privacy under continual observation. *In Proceedings of the Association for Computing Machinery Symposium on Theory of Computing, pages 715–724. Association for Computing Machinery* (2010).

[6] Yoav Freund and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55(1):119–139* (1997).

[7] Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. 2014. A second-order bound with excess losses. *In proceeedings of the 27th Annual Conference on Learning Theory(COLT), pages 176-196* (2014).

[8] Sreenivas Gollapudi and Debmalya Panigrahi. 2019. Online algorithms for rent-or-buy with expert advice. *In International Conference on Machine Learning, pages 2319–2327* (2019).

[9] Zhihao Jiang, Debmalya Panigrahi, and Kevin Sun. 2020. Online Algorithms for Weighted Paging with Predictions. *arXiv preprint arXiv:2006.09509* (2020).

[10] Anna R. Karlin, Mark S. Manasse, Lyle A. McGeogh, and Susan Owicki. 1994. Competitive Randomized Algorithms for non-uniform problems. *Algorithmica, 11(6):542–571* (1994).

[11] Wouter M. Koolen, Peter Grüwald, and Tim van Erven. 2014. Combining adversarial guarantees and stochastic fast rates in online learning. *In Advances in Neural Information Processing Systems 29, pages 2294-2302* (2014).

[12] Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. 2020. Online scheduling via learned weights. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1859–1877.

[13] N. Littlestone and Warmuth K. M. 1989. The weighted majority algorithm. *In Annual Symposium on Foundations of Computer Science, 1989, pages 256–261. IEEE* (1989).

[14] Haipeng Luo and Robert E. Schapire. 2015. Achieving all with no parameters : AdaNormalHedge. *In Proceedings of the 28th Annual Conference on Learning Theory(COLT), pages 1286-1304* (2015).

[15] Thodoris Lykouris and Sergei Vassilvitskii. 2018. Competitive caching with machine learned advice. *In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, pp. 3302–3311* (2018).

[16] Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. 2012. Online optimization with uncertain information. *ACM TALG, 8(1):2:1–2:29* (2012).

[17] Vahab S. Mirrokni, Shayan Oveis Gharan, and Morteza Zadimoghaddam. 2012. Simultaneous approximations for adversarial and stochastic online budgeted allocation. *In SODA, pages 1690–1701* (2012).

[18] Jaouad Mourtada and Stéphane Gaïffas. 2019. On the optimality of the Hedge algorithm in the stochastic regime. *J. Mach. Learn. Res.* 20 (2019), 83–1.

[19] Andres Munoz and Sergei Vassilvitskii. 2017. Revenue Optimization with Approximate Bid Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 1858–1866. http://papers.nips.cc/paper/6782-revenue-optimization-with-approximate-bid-predictions.pdf

[20] Kouvelis P. and Yu G. 2013. Robust Discrete Optimization and its Applications. *Springer Science and Business Media* (2013).

[21] Manish Purohit, Zoya Svitkina, and Ravi Kumar. 2018. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems, pages 9661–9670* (2018).

[22] Dhruv Rohatgi. 2020. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1834–1845.

[23] Steven D. Rooij, Tim Van Erven, Peter Grünwald, and Wouter M. Koolen. 2014. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research, 15:1281–1316* (2014).

[24] Amir Sani, Gergely Neu, and Allesandro Lazaric. 2014. Exploiting easy data in online optimization. *In Advances in Neural Information Processing Systems 27, pages 810–818* (2014).

[25] Shai Shalev-Shwartz. 2011. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning, 4(2):107–194* (2011).