Online Algorithms with Costly Predictions

Marina Drygala
EPFL

Sai Ganesh Nagarajan
EPFL

Ola Svensson
EPFL

Abstract

In recent years there has been a significant research effort on incorporating predictions into online algorithms. However, work in this area often makes the underlying assumption that predictions come for free (e.g., without any computational or monetary costs). In this paper, we consider a cost associated with making predictions. We show that interesting algorithmic subtleties arise for even the most basic online problems, such as ski rental and its generalization, the Bahncard problem. In particular, we show that with costly predictions, care needs to be taken in (i) asking for the prediction at the right time, (ii) deciding if it is worth asking for the prediction, and (iii) how many predictions we ask for, in settings where it is natural to consider making multiple predictions. Specifically, (i) in the basic ski-rental setting, we compute the optimal delay before asking the predictor, (ii) in the same setting, given apriori information about the true number of ski-days through its mean and variance, we provide a simple algorithm that is near-optimal, under some natural parameter settings, in deciding if it is worth asking for the predictor and (iii) in the setting of the Bahncard problem, we provide a $(1+\varepsilon)$-approximation algorithm and quantify lower bounds on the number of queries required to do so. In addition, we show that solving the problem optimally would require almost complete information of the instance.

1 INTRODUCTION

In recent years there has been an explosion of work that uses machine learning (ML) predictions in many online learning problems to improve performance, going beyond worst case analysis. Many classic online problems are studied in this setting under the umbrella of learning-augmented algorithms first popularized in the ML community by Lykouris and Vassilvtiskii (2018) for caching. A similar idea was initiated by Mahdian et al. (2007) for ad allocations in the presence of unreliable estimates. In this setting, we have access to a ML predictor appropriate for the problem, which may have an unknown error. Algorithms that use these ML predictions must ensure that, when the predictor is correct, the performance should match with that of the offline optimum and gracefully degrade to the best online algorithm when the ML predictions are unreliable. Existing work assumes that predictions from the ML oracle are free of cost. However, in practice, it is natural to consider scenarios where this is not true, as obtaining ML predictions may incur costs that are either computational or monetary due to necessary data collection processes. This potentially restricts the amount of calls one could make to the ML oracle to solve the problem at hand and if perhaps the cost is too high it may not be helpful to even ask for advice. This perspective introduces new aspects that differ from standard learning augmented algorithms. That is, without a costly predictor, it is *always* in our best interest to ask the predictor at the beginning of an online prediction interval ($t = 0$). However, this need not be the case when the prediction carries a cost. To this end, we conceptualize three fundamental questions when predictions carry a cost. (i) When do we ask the predictor? (ii) Given apriori information about the problem, such as certain useful statistics, should we make use of the predictor at all? (iii) How often should we ask the predictor to collect the required information to solve a problem either optimally or approximately? We remark, that question (iii) has been studied very recently in the setting of paging by Im et al. (2022).

In this paper, we study the phenomenon of costly predictions for online problems through the classic ski-rental problem for questions (i) and (ii). For question (iii), we require a setting where we would have repeated calls to the predictor and we therefore consider the bahncard problem, which is a well-studied and natural generalization of the ski-rental problem to a repeated horizon setting Fleischer (2001).

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

1.1 Ski-Rental and the Bahncard Problem

The ski-rental problem is a classic online decision-making problem posed as follows. We will ski for an unknown number of days t. On the beginning of each day, an irrevocable decision of whether to rent or buy skis must be made. Skis can be rented each day at cost 1 or bought for use during the remainder of the season at cost b. The performance of an online algorithm is measured against the offline optimum, which is easily computed once the duration of our ski season is revealed. The ratio of the cost of an online algorithm to the offline optimum on the worst-case instance is called the competitive ratio. It is well known that there is a simple deterministic buying strategy that achieves a competitive ratio of 2 Karlin et al. (1988). Additionally, Karlin et al. (1990) provide a randomized algorithm that achieves a competitive ratio of $\frac{e}{e-1}$. Both of the above algorithms are tight.

Bahncard problem: The bahncard problem can be viewed as a generalization of the ski-rental problem to a repeated horizon setting. Specifically, each day we take train trips of a given cost (potentially 0). If we do not have a valid bahncard at any given point, we have to make an online irrevocable decision of whether or not to purchase one at cost B. If we do not purchase one, we must pay full price for that day's tickets. If we have a valid bahncard on a given day, then we get a discount of $\beta \in [0, 1]$ for our ticket (i.e, discounted cost is $\beta \cdot$ original cost). Each bahncard is valid for T days. An algorithm for the bahncard problem should output the sequence of purchasing times for the bahncards. It was shown by Fleischer (2001) that there is a simple deterministic buying strategy that achieves a competitive ratio of $2 - \beta$. Finally, Karlin et al. (2001) provide a randomized algorithm that achieves a competitive ratio of $\frac{e}{e-1+\beta}$. Note that when $\beta = 0$ and $B = b$ and $T \to \infty$, this reduces to the ski-rental problem.

1.2 Our Contributions

In this paper, we study the ski-rental and bahncard problem with predictions that carry a cost ($c \geq 0$), and we ask the following questions: (i) When do we ask the predictor in the ski-rental problem? (ii) Given apriori information about the mean and variance of the true number of ski days, should we ask the predictor? (iii) In the bahncard problem, how often should we ask the predictor for the true number of trips taken in certain intervals?

Ski-rental with Costs and Prior Information: For the first question, we provide a simple algorithm that waits for an optimum amount of time (roughly $\sqrt{cb}$ days) to ask for the prediction and following it's advice thereafter. We show that this algorithm is optimal by minimizing the competitive ratio whilst considering the cost of asking for a prediction. To address the second question, we propose a simple deterministic algorithm which is near-optimal (tight up to a constant, for any randomized algorithm) under some natural parameter settings in deciding whether to ask the predictor, when only the mean and the variance is known. We show that there is a threshold function $f^*(\mu, \sigma, b)$, that essentially computes the "value" of the prediction. For instance, if $\mu = b$, for a fixed c (say $\sqrt{b}$), we have that as σ varies and crosses c, the uncertainty on the worst-case distribution becomes sub-optimal for any algorithm and in which case it would be better to ask the predictor. We can also make our algorithms robust to prediction errors by using standard techniques from Purohit et al. (2018). For completeness, we describe this in the supplementary sections (A.2,A.6).

Bahncard problem with Few Predictions: Our main technical contributions are on the bahncard problem, a generalization of the ski-rental problem to a repeated horizon setting. When we associate a cost to each prediction, it is natural to ask how many predictions are required to gather enough information to output a buying schedule that is close to optimal. To this end, we characterize the query complexity of solving this problem both optimally and to a factor of $1 + \varepsilon$. We provide upper and lower bounds on the number of queries required to achieve a $(1 + \varepsilon)$-approximation algorithm. The upper bound given by our algorithm is nearly tight. Our approach in this part is based on several novel ideas and is more technically advanced compared to the simple and clean algorithms that we analyze for the ski-rental problem. In particular, we heavily exploit structural properties of the bahncard problem to compute intervals of possibly high cost, that we need not obtain information on in order to compute a good solution. Finally, we describe how to modify our algorithm to accommodate prediction errors. This modification requires new ideas on how to partition the timeline to appropriately to charge costs and prediction errors.

1.3 Related work

There is plenty of recent work focusing on incorporating ML predictions in the design of online algorithms. We restrict our attention to those that are most relevant to us, which are the buy-or-rent type online problems. For instance, Gollapudi and Panigrahi (2019), focused on combining advice from multiple experts, while Wang et al. (2020) studied the multi-shop version of the ski-rental problem. Applications such as caching Rohatgi (2020); Lykouris and Vassilvtiskii (2018) and scheduling Lattanzi et al. (2020) were also studied to improve the performance of online algorithms. In the presence of errors, the main goal is to have the optimal robustness, consistency trade-off Purohit et al. (2018). Moreover, Bamas et al. (2020) identify a way of utilizing the primal-dual method in the learning augmented setting for online covering problems, but we remark that all the aforementioned papers assume that the predictions come for free.

Recent papers such as Anand et al. (2020, 2021), use a PAC learning approach to create an ML predictor for buy-or-rent type problems and derive the sample complexity required to achieve a certain consistency, robustness trade-off.

Meanwhile, Diakonikolas et al. (2021), rely on access to samples from the distribution of the true number of ski-days to obtain an algorithm for ski-rental and focus on the number of samples required to do so. They do this by estimating the empirical distributions from i.i.d samples. We note that in studying the value of predictions, we do not have access to the distribution, nor make any assumptions on it and we only have access to the mean and variance of the true number of ski-days.

Query Minimization It is natural to consider repeated calls to the predictor in online problems such as caching and scheduling. As such, if there are costs associated with the prediction, frequent calls may jeopardize the algorithm's performance, even if the predictions are correct. Recently Im et al. (2022) studied the query minimization for the caching problem in the learning augmented setting, where they consider predictions that are error-prone. Another recent work Antoniadis et al. (2022), studies paging with "succinct" predictions by using a one bit prediction alongside a page request, which is a different model as compared to Im et al. (2022). As mentioned, we study the bahncard problem which is a different problem altogether. Also, the works above do not consider the other aspects such as, when to ask and the usefulness of a costly predictor in the presence of simple prior information. In another prior work, Bamas et al. (2020) does consider the bahncard problem, where they have free, but error-prone predictions. In addition, they assume that the forecasts given by the ML algorithm are complete solutions (i.e., a sequence of purchasing times). In this paper, we have predictions that provide the true total cost of trips taken in intervals of desired lengths, which is a more natural choice of prediction from an ML perspective.

Query minimization has also been considered in relation to other traditional combinatorial optimization problems. The Minimum Spanning Tree (MST) problem has been studied in the setting in which the edge weights are uncertain and the goal is to identify a minimum-weight spanning tree with the fewest number of queries Erlebach et al. (2008); Megow et al. (2017). Other settings have also been considered such as scheduling Dürr et al. (2020), in which queries can be made to reduce processing times. In Singla (2018), the author studies the effect of associating a price on the information required to be collected in order to solve some combinatorial optimization problems such as MST, in the presence of some stochastic information about the edge weights. We remark that our model is different from theirs in the sense that, even in the presence of existing information, they pay costs to gather additional information (even if the distribution is a point distribution). We take a tangential approach in this paper, to try and understand if one needs to ask the predictor at all, given apriori information. Finally, in the bahncard problem, we do not have access to any prior information.

Bandits with costs The idea of associating costs for observations has appeared before in the bandit literature Seldin et al. (2014). In this setting, they have multiple experts and the game goes for multiple rounds and a reward is given for each action taken. In addition, they may pay some cost to observe the rewards for some subset of arms in each round. To deal with the cost of observations, they maintain a probability distribution for the experts and the probability of querying the rewards of arms. In complementary work, there is also the idea of a budget constrain to observe feedback Efroni et al. (2021) in the bandit settings, for instance in applications like recommendation systems. In Bhaskara et al. (2021), they consider the online linear optimization problem with hints and provide an algorithm that achieves $O(\log T)$ regret with $O(\sqrt{T})$ hints (which is tight). Recently, Bhaskara et al. (2022), extend the above work to a probing based model, where the algorithm (is allowed constant number of probes), provides the oracle with a few options asking for the best option and obtain improved regret guarantees for the hints problem. Our work deals with settings that does not have any feedback for the irrevocable actions that are taken and the game ends as soon as we play a strategy.

2 THE TIMING AND VALUE OF A PREDICTION

As a warm-up, we look at the first problem, as to decide, when to ask the predictor. If the predictor is free ($c = 0$), then it is in our best interest to ask at day 0. Now consider a simple example, where $c = 20$ and $b = 100$. If the true number of days is 10, then by simply asking at day 0, our competitive ratio will be 3, as opposed to waiting for 10 days. Thus it is clear, that an improved algorithm might need to wait a certain amount of time before asking, so as to balance out different cases based on the outcome of the predictor. To this end, we can run a simple algorithm as follows. We wait until a day t^* to ask the predictor and simply follow its advice thereafter. Let d^* be the true number of days that is unknown to us before asking the predictor. The algorithm that we call as ToP (Timing of Prediction), is given by:

$$\begin{cases} \text{Buy at day b without asking, if } c \geq \theta^* b \\ \text{Ask predictor on day } t^*+1 \text{ and rent if} \\ \qquad\qquad d^* < t^* + b \text{ and } c < \theta^* b \\ \text{Ask the predictor at day } t^*+1 \text{ and buy immediately if} \\ \qquad\qquad d^* \geq t^* + b \text{ and } c < \theta^* b \end{cases} \tag{1}$$

Here, θ^* is the optimal threshold that can be identified from our algorithm. Now, we can find the optimal t^* yielding the optimal deterministic algorithm for this setting.

Theorem 2.1. *The competitive ratio of the algorithm ToP is* $\max_{\mathcal{I}} \frac{ToP(\mathcal{I})}{OPT(\mathcal{I})} \leq \min\left\{1 + \frac{\sqrt{(c+1)^2+4c(b-1)}+c-1}{2b}, 2\right\}$

For the proof refer to Appendix A.1.

2.1 Value of a Prediction

To understand the value of a prediction in the presence of apriori information, we assume that we are given the mean and the variance of the true number of ski-days. In practice, mean and variance can be obtained from historical data and is easier than estimating the apriori distribution of the true number of days skied. Let X denote the random variable indicating this quantity, where the mean is denoted by $\mathbb{E}[X] = \mu$ and the variance is denoted by $\mathbb{V}ar[X] = \sigma^2 > 0$.

If we are given the apriori distribution, then we can potentially find the best deterministic algorithm by choosing a buying strategy that minimizes the expected cost in the following sense:

$$\text{ALG}^* := \min_{d \geq 0} \mathbb{E}\left[\mathbf{1}\{X \leq d\}X + (d+b)\mathbf{1}\{X > d\}\right]. \tag{2}$$

Similarly, the (expected) optimum cost is given by:

$$\text{OPT} := \mathbb{E}\left[\min\{X, b\}\right]. \tag{3}$$

It is known that in the worst case the competitive ratio for $\frac{\text{ALG}^*}{\text{OPT}} \leq \frac{e}{e-1}$ and this is the lower bound for any randomized algorithm for ski-rental due to Yao's minimax lemma Yao (1977).

Thus we come to the main question in this section:

Given the mean μ and the variance σ^2 of an apriori distribution of the number of ski-days should we pay a cost c to obtain the prediction?

To this end we work with continuous distributions that are supported in $[0, \infty)$ and buying strategies $d \geq 0$, as is common in the literature Karlin et al. (2001); Madry and Panigrahi (2011). As stated before we assume that we have access to the true prediction after paying a cost of c.

Consider, the "algorithm" that simply buys the prediction on day 0 and follows it thereafter, (PRED) pays:

$$\text{PRED} := c + \text{OPT} \tag{4}$$

When the full apriori distribution is known, one can possibly obtain ALG^* and then compare to PRED and choose the better of the two. But this is not trivial when only the mean and the variance are known.

To this end, we consider this simple buy on day 0 or rent forever, which we show is already tight for natural parameters ($\mu = \Theta(b)$ and $\sigma = o(b)$).

$$\text{BoR} := \begin{cases} \text{Buy on day 0}, & \text{if } \mu \geq b \\ \text{Rent forever}, & \text{if } \mu < b \end{cases}$$

We identify a threshold function $f^*(\mu, \sigma, b)$, and show that this is actually the upper bound of BoR. Thus, the threshold function will enable us to decide if we should ask the predictor. That is if c is greater than this threshold value, we use BoR, otherwise we ask the predictor, i.e, use PRED. We call this algorithm VoP (Value of Prediction).

In general, we let any deterministic strategy be denoted by d, i.e, d buys the ski at the start of day d. We denote the cost incurred by strategy d to be $\text{C}(d)$.

Now, we can state the main theorem of this section.

Theorem 2.2. *There exists a threshold function $f^*(\mu, \sigma, b)$, such that the algorithm VoP satisfies, VoP $\leq$ OPT $+ \min(f^*(\mu, \sigma, b), c)$.*

Proof. We first show the upper bounds in Lemma 2.5 and compute the threshold function $f^*(\mu, \sigma, b)$, by considering the set of possible distributions with mean μ and variance σ^2, which are supported at exactly two points, which we will refer to as a *two-point* distribution and show that it suffices to study only this class. □

Remark 2.3. Under some natural parameter choices, i.e, when $\mu = \Theta(b)$ and $\sigma = o(b)$, the upper bounds are tight up to a constant in that, there exists distributions that are supported at exactly two points, such that any randomized algorithm pays (in expectation) at least, OPT $+ \min\left(\frac{1}{7}(f^*(\mu, \sigma, b)), c\right)$. See Lemma 2.6 for details.

In the main body, we consider the simplest case, i.,e $\mu = b$ and $\sigma \leq b$, as a warm-up and highlight the relationship between the cost and the parameters.

Lemma 2.4. *If $\mu = b$ and $\sigma \leq b$, then we have that algorithm VoP satisfies, VoP $\leq$ OPT $+ \min\left(\frac{\sigma}{2}, c\right)$.*

Proof. First we assume that we can work with distributions, that are supported at exactly two points (w.l.o.g) and later show that this is sufficient. One can represent a *two-point* distribution as $\mu - k_1\sigma$ and $\mu + k_2\sigma$, where $k_1 > 0$ and $k_2 > 0$, that has mean μ and variance σ^2.

Let $p_1 > 0$ and $p_2 > 0$, be such that, the following equations hold:

$$p_1(\mu - k_1\sigma) + p_2(\mu + k_2\sigma) = \mu \tag{5}$$
$$p_1(k_1\sigma)^2 + p_2(k_2\sigma)^2 = \sigma^2 \tag{6}$$
$$p_1 + p_2 = 1 \tag{7}$$

Then solving the above set of equations gives us $k_2 = \sqrt{\frac{p_1}{p_2}}$ and $k_1 = \frac{1}{k_2}$. Thus setting $\frac{p_1}{p_2} = l$, we can parameterize all such two-point distributions by l.

Note that BoR always pays b, when $\mu = b$, thus let us define $\mathrm{OPT}(l)$ to be the cost of OPT on the two-point distribution $p_{\mu - \frac{\sigma}{\sqrt{l}}} = \frac{l}{l+1}$ and $p_{\mu+\sqrt{l}\sigma} = \frac{1}{l+1}$.

If $\mu = b$, then , $\mu - \frac{\sigma}{\sqrt{l}} < b$, as $\sigma > 0$. Thus we only require that $\sqrt{l} \geq \frac{\sigma}{\mu}$, since the support point must be non-negative.

To find the upper bound on BoR $-$ OPT, we are required to solve this optimization problem.

$$\max_{\sqrt{l} \geq \frac{\sigma}{b}} \frac{\sigma\sqrt{l}}{l+1} \tag{8}$$

This is because $\mathrm{OPT} = \frac{l}{l+1}(b - \frac{\sigma}{\sqrt{l}}) + \frac{1}{l+1}b$ and BoR pays b in this setting. It is easy to see that the maximum value of the above problem is attained at $\sqrt{l} = 1$ and this is feasible as $\sigma \leq b$ and this means that:

$$\mathrm{BoR} - \mathrm{OPT} \leq \frac{\sigma}{2} \tag{9}$$

Now, to see why it suffices to consider only *two-point* distributions. Note that $\mathrm{OPT} = \mathbb{E}[\mathbf{1}\{X < b\}X] + b\mathbb{E}[\mathbf{1}\{X \geq b\}]$. The difficult situation is when there are support points both above and below b. We can set $\frac{l}{l+1}(\mu - \sigma/\sqrt{l}) = \mathbb{E}[\mathbf{1}\{X < b\}X]$. This can be used to solve for a unique l^*. Now place the second point that is $\mu + \sigma\sqrt{l^*}$ and this leaves OPT, μ and σ unchanged. In addition, our algorithm buys at day 0 and always pays b, independent of the distribution that is considered.

Finally, it is easy to see that when we take the minimum of $\mathrm{BoR} \leq \mathrm{OPT} + \frac{\sigma}{2}$, and $\mathrm{PRED} = \mathrm{OPT} + c$, we get the required upper bound.

□

We remark that this upper bound is tight, when we consider $\sigma = o(b)$. The worst case *two-point* distribution, is $p_{b-\sigma} = \frac{1}{2}$ and $p_{b+\sigma} = \frac{1}{2}$. It is easy to see that buying before $b - \sigma$, costs at least $\frac{\sigma}{2}$ and buying after $b - \sigma$ costs at least $\frac{\sigma}{2}$, when $\sigma = o(b)$, (since $b - \sigma > \sigma$). Now renting pays $\mu = b$ and hence the lower bound applies to all deterministic strategies and it is tight.

From the above case, it is clear that, as long as, $\frac{\sigma}{2} < c$, it is less expensive to manage this uncertainty, than the cost you would have to pay to circumvent it completely and in which case, we do not have to ask the predictor. Now, the dependence on the parameters becomes more complicated in other cases. But we can show the following statement in general.

Lemma 2.5. *The algorithm VoP satisfies VoP $\leq$ OPT $+ \min(f^*(\mu, \sigma, b), c)$.*

Refer to Appendix A.3, for the full proof.

Finally, we end with the lemma that provides the lower bounds for other cases which is tight with a loss of small constant, under certain natural parameter settings, i.e, $\mu = \Theta(b)$ and $\sigma = o(b)$.

Lemma 2.6. *If $\mu = \Theta(b)$ and $\sigma = o(b)$, then, there exists two-point distributions with mean μ and variance σ^2 such that any deterministic strategy d, satisfies $C(d) - OPT \geq \min\left(\frac{1}{7}f^*(\mu, \sigma, b), c\right)$*

The proof of the lower bound is in Appendix A.5.

3 CIRCUMVENTING FREQUENT COSTLY PREDICTIONS

In this section, we consider our final question. That is, how often must we ask for predictions in order to gain sufficient information to solve an instance of the bahncard problem optimally or approximately.

Recall, that in a traditional instance of the bahncard problem, we are given a cost B of purchasing a card, a time horizon of T for which a card is valid, and a discount rate $\beta \in [0, 1)$. Then, at each point $i = 0, \ldots, N-1$ over a finite time horizon lasting N days, a travel request is revealed in the form of a cost c_i of tickets that need to be purchased immediately.

If we have a consecutive set $i, i+1 \ldots, i'$ of days, together we say they form an *interval*. We denote such an interval by $[i, i']$. We will sometimes use $[i, i')$ to denote $[i, i'-1]$. The intervals (i, i') and $(i, t']$ are defined analogously. The traveler needs to decide when to purchase discount cards online. If a bahncard was purchased at some time $i + 1 - T, \ldots, i-1$, then they pay $p_i = \beta c_i$ to travel on day i. Otherwise, they can purchase a card for cost B, and pay a price $p_i = \beta c_i$ for their travels, or opt to pay a price of $p_i = c_i$. The goal is to minimize the total expenditure namely,

$$mB + \sum_{i=1}^{N} p_i,$$

where m is the number of cards purchased. A solution to the problem can be represented as a possibly empty sequence $\tau = (\tau_1, \ldots, \tau_m)$, of time points in $0, \ldots, N-1$ at which we buy our cards.

We wish to gain an intuition about the minimum quantity of information we must extract about the structure of the cost distribution $(c_0, \ldots, c_{N-1})$ to be able to improve on the tight bound of $2 - \beta$ for the online setting without predictions. This approach is motivated by the scenario in which we can use a predictor to estimate our travel behaviour beforehand for some portion of our lifetime. An example of this would be travel behaviour over holidays.

The authors of Bamas et al. (2020) consider a model in which the predictor suggests a complete buying schedule for the cards. It is natural, however to consider a predictor that for a given time interval, for example a duration of one month, can estimate the total cost of the trips taken during that time period. We show in Theorem 3.2 that if we limit ourselves to periods of one day we require $\Omega(N)$ queries to get a $(1+\epsilon)$-approximation, further motivating the prediction model in which we can query longer intervals.

To this end, we will be presented with a predictor who we can provide a length L, where $1 \leq L \leq N$ and a day $t \leq N - L$, and in turn the predictor will inform us of the value of

$$\sum_{i=t}^{t+L} c_i.$$

We will call such a predictor an *interval estimator*.

In section 3.1 we provide lower bounds on the number of predictions required from our interval estimator to determine the optimal solution and a $(1+\epsilon)$-approximation, respectively. As we will see in Theorem 3.1, in order to compute the optimal solution we require $\Omega(N)$ queries. In addition we show in Theorem 3.3, we show that to get a $(1+\epsilon)$-approximation of the optimal offline solution, we require $\Omega(N/T)$ queries from the predictor.

In section 3.2 we give a nearly tight $(1+\epsilon)$-approximation algorithm, where we exploit structural properties of the bahncard problem to avoid querying sections of lifetime.

3.1 Lower Bounds on Number of Predictions

In this section, we present the lower bounds on the query complexity of the bahncard problem.

Theorem 3.1. *Suppose we are given an interval estimator. Then in expectation any randomized algorithm requires $\Omega(N)$ queries to determine the optimal solution.*

Proof. By Yao's Principle, it suffices to give a distribution over a family $\mathcal{X}$ of instances of the bahncard problem, such that in expectation, any deterministic algorithm must query $\Omega(N)$ times to distinguish between instances whose set of optimal solutions are disjoint.

We create such a family $\mathcal{X}$ as follows. Fix some $T \geq 1$ and $N \geq 2T$. Let $B = T$ and $\beta = 0$. For each day $i = 1, \ldots, N-T$ add the following instance to $\mathcal{X}$. The cost of the trips on days i and $i+1$ are 0 and 2 respectively, and the cost of the trips on each remaining day is 1. Call the resulting instance, instance i. Finally, add to $\mathcal{X}$ the instance in which the cost of the trips on each day is 1. We call this final instance the *uniform* instance.

One can construct an optimal solution to the uniform instance by purchasing any number of bahncards beginning on or before day $N - T$ whose validity periods do not intersect. Similarly, one can construct an optimal solution to instance i by purchasing a card at the beginning of day $i+1$ and any number of additional cards so that all cards begin on or before day $N-T$ and have non-intersecting validity periods.

If given the uniform instance, U, in order for any algorithm to be certain it has produced the optimal solution, it must distinguish U from the remaining instances in $\mathcal{X}$. Any given query can distinguish U from at most two other instances in $\mathcal{X}$. Since we have $N-T$ non-uniform instances this requires at least $\frac{N-T}{2}$ queries.

Instead, if we are presented with instance i to distinguish instance i from the rest of $\mathcal{X}$, we must query an interval that ends precisely after day i or one that begins precisely on day $i+1$.

Let Q_A be the random variable that takes on the number of queries required by an algorithm A over the uniform distribution of instances over $\mathcal{X}$. Suppose that we draw an instance X from our probability distribution. Observe that after A has made at most $N/4$ queries, with probability at least $1/2$ A has not determined with certainty which instance X is. Thus the expected number of queries is $\Omega(N)$ as required.

Hence by Yao's principle no randomized algorithm can determine the optimal solution with fewer than $\Omega(N)$ queries, in expectation.

□

The proofs of the following theorems are similar and are provided in the appendix.

The intuition behind Theorem 3.2 is as follows. If we can only query single days then it may take many queries to determine if in any given time period it is worthwhile to buy a bahncard. The proof proceeds by constructing a family of instances, where one instance can be obtained from another by shifting around ticket purchases from various days.

The proof of Theorem 3.3 also uses a similar argument.

Theorem 3.2. *Suppose we are given an interval estimator that only queries single days. Then, in expectation, any randomized $(1+\epsilon)$-approximation algorithm requires $\Omega(N)$ queries to determine the optimal solution, in expectation.*

Theorem 3.3. *Suppose that we are given an interval estimator. Then, in expectation any randomized $(1+\epsilon)$-*

approximation algorithm requires $\Omega(N/T)$ queries to determine the optimal solution.

3.2 A Nearly Tight Algorithm

Here we present an algorithm that returns a $(1+\epsilon)$-approximation for the bahncard problem, when compared to the optimal offline solution. The number of queries our algorithm requires has near optimal dependence on N/T.

The following observations are central to our approach.

(a) If a card $OPT(\mathcal{I})$ has little ticket costs at the beginning or end of its validity period we may shift its purchase time at a small charge.

(b) Suppose we know that we will purchase a card during an interval $I = [s, s']$. Then, if $s + T \geq s'$, there is no need to gather information about costs in $[s', s+T]$ as we will have a valid card during this time period.

We formalize these observations, and then apply a binary-type search on intervals with large cost to pinpoint days that we will restrict our purchases to at a small charge.

As noted, we will make use of the optimal offline algorithm for the bahncard problem presented by the authors of Fleischer (2001), presented in Theorem 3.4. Proof details are provided in the appendix.

Theorem 3.4. *Fleischer (2001) Suppose we are given an instance $\mathcal{I} = ((c_0, \dots c_{N-1}), T, B, \beta)$ of the bahncard problem where $(c_0, \dots c_{N-1})$ is known in advance. Then we can solve $\mathcal{I}$ optimally.*

To aid us in formalizing our approach, we define the Modified Bahncard Problem (MBP). An instance $\mathcal{I}$ of MBP is given by $((c_0, \dots, c_{N-1}), \mathcal{P}, B, \beta)$, where $\mathcal{P}$ is a set containing elements of the form (i, P_i). Each such element (i, P_i) corresponds to a bahncard that can be purchased at time $i \in \{0, \dots, N-1\}$, with validity period P_i. Again, we pay B for each card we purchase and p_i for each day, where we recall that $p_i = \beta c_i$ if there is a valid bahncard during day i and $p_i = c_i$ otherwise. We wish to purchase the set of bahncard in $\mathcal{P}$ that minimizes the cost.

Note that an instance

$$\mathcal{I} = ((c_0, \dots, c_{N-1}), T, B, \beta)$$

of the bahncard is equivalent to the instance

$$\mathcal{I}' = ((c_0, \dots, c_{N-1}), \mathcal{P}, B, \beta)$$

of MBP, where

$$\mathcal{P} = \{(i, T) : i \in 0, \dots, N-1\}.$$

If we have complete information about the ticket costs incurred during the validity periods of each card in $\mathcal{P}$, then we can solve an offline instance of MBP optimally using essentially the same construction as in Theorem 3.4. This is formalized in Observation 3.5. In general for a set of days $D \subseteq N$, we say D is *fully-determined* if we know

$$\sum_{i \in D} c_i.$$

Observation 3.5. Suppose we are given an instance $\mathcal{I} = ((c_0, \dots c_{N-1}), \mathcal{P}, B, \beta)$ of MBP, such that each pair of consecutive points i, i' in

$$\{i : (i, P) \in \mathcal{P}\} \cup \{i + T : (i, P) \in \mathcal{P}\} \cup \{0, N\},$$

satisfy the property that $[i, i')$ is fully-determined. Then we can efficiently solve the instance $\mathcal{I}$.

Throughout, we will refer to multiple MBP instances, so we introduce some notation. For an instance $\mathcal{I}$ of MBP, we will use $OPT(\mathcal{I})$ to denote a fixed optimal solution to $\mathcal{I}$, and *opt*$(\mathcal{I})$ to indicate the cost of such a solution. Note that $OPT(\mathcal{I})$ can be represented by a subset of $\mathcal{P}$.

Consider the following fact, the proof of which can be found in the appendix.

Fact 3.6. Suppose we are given an instance $\mathcal{I} = ((c_0, \dots c_{N-1}), T, B, \beta)$ of the bahncard problem. If we extend the validity period of all bahncards to $2T$, we can restrict our purchase times to integer multiples of T without increasing the value of the optimal solution. Call the resulting modified instance of MBP, $\mathcal{I}'$. During any period I of the lifetime, if $OPT(\mathcal{I}')$ purchases b bahncards during I, then we can assume that $OPT(\mathcal{I})$ purchases at most $2b$ bahncards during I. Furthermore

$$\frac{1}{2} opt(\mathcal{I}) \leq opt(\mathcal{I}') \leq opt(\mathcal{I}). \tag{10}$$

First, we present an approach that requires

$$O\left(\frac{N}{T} + \left(\frac{b}{\gamma}\right)^b \log(T)\right) \tag{11}$$

queries to obtain a $(1+\gamma)$ approximation, where b is an upper bound on the number of bahncards purchased by $OPT(\mathcal{I})$. Taking $\gamma = \epsilon$, then gives the desired result.

We begin by querying the intervals

$$[0, T], [T, 2T], \dots, [(\lceil N/T \rceil - 1)T, N],$$

so that we may apply Observation 3.5 to solve the instance $\mathcal{I}'$ described in Fact 3.6. Suppose that this solution purchases b' cards. Then by Fact 3.6, if we define

$$b = \min\{N/T, 2b'\}, \tag{12}$$

we obtain that $b^* \leq b$, where b^* is the number of cards purchased by $OPT(\mathcal{I}^*)$. We define $g = 2opt(\mathcal{I}')$, and note that by Equation (10) $g \geq opt(\mathcal{I})$.

As b may be $\Omega(N/T)$ we show in the appendix how to modify our approach, by essentially "splitting" our instance into smaller instances. The splitting procedure fixes a card purchase once we know that $OPT(\mathcal{I})$ has accumulated sufficient cost since the last split. The upper bound on the number of queries required in this case is

$$O\left(\frac{N}{T}\log(T)\right),$$

for fixed ϵ.

Next we define any subset S of days in the lifetime of $\mathcal{I}$ to be γ-light if the total ticket costs during S does not exceed $\frac{g\gamma}{b}$, and γ-heavy otherwise. Note that if $b = 0$, then we need not proceed any further, so this quantity is well-defined. The notion of γ-lightness is motivated by point (a).

We construct a subset D of days so that the instance

$$\mathcal{I}^* = ((c_0, \ldots c_{N-1}), \mathcal{P}^*, B, \beta)$$

of MBP where

$$\mathcal{P}^* = \{(i, T) : i \in D\}$$

that has the desirable property that

$$opt(\mathcal{I}^*) \leq (1+\gamma)opt(\mathcal{I}).$$

Furthermore we show that in our construction of D we fully-determine the validity periods of all cards in $\mathcal{P}^*$ using at most the number of queries given in Equation (11).

Suppose that we have a solution including a bahncard ending at day s, and that there is a later date s' such that the ticket costs in $I = [s, s')$ exceed g. If we do not wish to accumulate ticket costs exceeding $opt(\mathcal{I})$ we must purchase an additional card during I. This motivates the construction of the algorithm Construct-Starting-Points(s), which takes as input a point s and suggests at most $\frac{2b}{\gamma} + 1$ starting points for the next card for which to purchase the next card, given that we have a card ending at s.

Construct-Starting-Points(s) outputs a set $D_s \subseteq [s, N)$ and terminates when one of the following conditions is met. Either D_s contains a point s' such that $[s, s')$ has ticket costs exceeding g or D_s contains a point s' such that $s' + T \geq N$. The second condition is included as we may assume that no card is purchased later than s'. The algorithm proceeds as follows. First, we initialize $d = s$, and while neither termination condition is met, we let $t_0 = d, \ldots, t_k = N$ be the maximal subset of $[d, N]$ such that $[t_i, t_{i+1})$ is *fully-determined* for each $i = 0, \ldots k-1$ (Recall that an interval is fully-determined if we know the total cost of the tickets purchased during the interval).

If $[t_0, t_1)$ is a single γ-heavy day then we add t_1 to D_s, and update d to be t_1. If $[t_0, t_k)$ is γ-light then we add t_k to D_s and terminate. Otherwise, take k' to be the largest value in $0, \ldots, k-1$ such that $J = [t_0, t_{k'})$ is γ-light. If $[t_{k'}, t_{k'+1})$ is γ-light or a single heavy day we add $t_{k'}$ to D_s. In addition, we update d to be $t_{k'}$. Otherwise we let m be the midpoint of $[t_{k'}, t_{k'+1})$, and we query $[t_{k'}, m)$. Then we update our list of points to be $t_0, \ldots, t_{k'}, m, t_{k'+1}, \ldots, t_k$ and repeat. Let

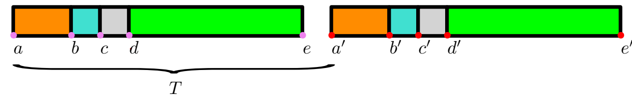

Figure 1: Illustration of the construction of D_a = Construct-Starting-Points(a). In this figure D_a, consists of $\{a, b, c, d, e\}$. During our construction of D we would run Construct-Starting-Points on each point in $\{a', b', c', d', e'\}$. In this depiction $e - a < T$ and hence $[a, e)$ has ticket costs exceeding g. Note that $[a, b]$ and $[d, e]$ may be composed of many minimal fully-determined sub-intervals as specified by the update procedure.

$$D^1 = \text{Construct-Starting-Points}(0),$$

be a set of suggested locations for our first card. We create additional sets $D^2, \ldots D^{b+1}$ as follows. Suppose that $D^1, \ldots, D^j$ have been constructed. We let

$$D^{j+1} = \bigcup_{s \in D^j} \text{Construct-Starting-Points}(s+T)$$

The shift of T days to the right in construction of D^{j+1} is motivated by point (b). After D is complete we ensure with at most one additional query for each element i of D that $[i, i+T)$ is fully determined so that we may apply Observation 3.5 to solve $\mathcal{I}^*$.

We remark that we need not run Construct-Starting-Points to completion for each point s as we may examine intervals of the lifetime repeatedly. However, we have chosen our construction of D for simplicity of exposition.

As a result of our careful construction of D we get Theorem 3.7, whose proof we provide in the appendix.

Theorem 3.7. *The instance $\mathcal{I}$ satisfies*

$$opt(\mathcal{I}^*) \leq (1+\gamma)opt(\mathcal{I}),$$

and D requires at most

$$O\left(\frac{N}{T} + \left(\frac{b}{\gamma}\right)^b \log(T)\right)$$

queries to compute.

We can make our algorithm consistent and robust to errors via the following theorem which we prove in the appendix (C.3). Here we let $ALG(\mathcal{I})$ denote our robust buying schedule. Recall that our prediction algorithm is parameterized by γ, and guarantees a solution of cost at most $(1+\gamma)opt(\mathcal{I})$ under the assumption that the predictions are correct.

Theorem 3.8. *For a fixed $\lambda \in (0,1)$ $ALG(\mathcal{I})$ is a solution with performance guarantee at most $\left(1+\frac{1}{\lambda}\right) opt(\mathcal{I})$. In the case that the predictions are perfect, we pay at most $(1+\max\{2\gamma, \lambda\}) opt(\mathcal{I})$.*

Acknowledgements

This work was supported by the Swiss National Science Foundation project 200021-184656 "Randomness in Problem Instances and Randomized Algorithms."

References

Keerti Anand, Rong Ge, and Debmalya Panigrahi. Customizing ml predictions for online algorithms. In *International Conference on Machine Learning*, pages 303–313. PMLR, 2020.

Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. A regression approach to learning-augmented online algorithms. *Advances in Neural Information Processing Systems*, 34, 2021.

Antonios Antoniadis, Joan Boyar, Marek Eliáš, Lene M Favrholdt, Ruben Hoeksma, Kim S Larsen, Adam Polak, and Bertrand Simon. Paging with succinct predictions. *arXiv preprint arXiv:2210.02775*, 2022.

Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. *arXiv preprint arXiv:2010.11632*, 2020.

Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Logarithmic regret from sublinear hints. *Advances in Neural Information Processing Systems*, 34: 28222–28232, 2021.

Aditya Bhaskara, Sreenivas Gollapudi, Sungjin Im, Kostas Kollias, and Kamesh Munagala. Online learning and bandits with queried hints. *arXiv preprint arXiv:2211.02703*, 2022.

Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, Ali Vakilian, and Nikos Zarifis. Learning online algorithms with distributional advice. In *International Conference on Machine Learning*, pages 2687–2696. PMLR, 2021.

Christoph Dürr, Thomas Erlebach, Nicole Megow, and Julie Meißner. An adversarial model for scheduling with testing. *Algorithmica*, 82(12):3630–3675, 2020.

Yonathan Efroni, Nadav Merlis, Aadirupa Saha, and Shie Mannor. Confidence-budget matching for sequential budgeted learning. *arXiv preprint arXiv:2102.03400*, 2021.

Thomas Erlebach, Michael Hoffmann, Danny Krizanc, Matús Mihal'Ák, and Rajeev Raman. Computing minimum spanning trees with uncertainty. *arXiv preprint arXiv:0802.2855*, 2008.

Rudolf Fleischer. On the bahncard problem. *Theoretical Computer Science*, 268(1):161–174, 2001.

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

Sungjin Im, Ravi Kumar, Aditya Petety, and Manish Purohit. Parsimonious learning-augmented caching. *arXiv preprint arXiv:2202.04262*, 2022.

Anna R Karlin, Mark S Manasse, Larry Rudolph, and Daniel D Sleator. Competitive snoopy caching. *Algorithmica*, 3(1):79–119, 1988.

Anna R Karlin, Mark S Manasse, Lyle A McGeoch, and Susan Owicki. Competitive randomized algorithms for non-uniform problems. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 301–309, 1990.

Anna R Karlin, Claire Kenyon, and Dana Randall. Dynamic tcp acknowledgement and other stories about e/(e-1). In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 502–509, 2001.

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

Thodoris Lykouris and Sergei Vassilvtiskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning*, pages 3296–3305. PMLR, 2018.

Aleksander Madry and Debmalya Panigrahi. The semi-stochastic ski-rental problem. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Allocating online advertisement space with unreliable estimates. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 288–294, 2007.

Nicole Megow, Julie Meißner, and Martin Skutella. Randomization helps computing a minimum spanning tree under uncertainty. *SIAM Journal on Computing*, 46(4): 1217–1240, 2017.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. *Advances in Neural Information Processing Systems*, 31:9661–9670, 2018.

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

Yevgeny Seldin, Peter Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *International Conference on Machine Learning*, pages 280–287. PMLR, 2014.

Sahil Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2523–2532. SIAM, 2018.

Shufan Wang, Jian Li, and Shiqiang Wang. Online algorithms for multi-shop ski rental with machine learned advice. *Advances in Neural Information Processing Systems*, 33, 2020.

Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227. IEEE Computer Society, 1977.

A Missing Proofs in Section 2

A.1 Competitive Ratio of ToP

Proof of Theorem 2.1. Let t^*, be our deterministic strategy and let d^*, be the day chosen by the online adversary (which is unknown to us).

If $c \geq \theta b$, we never ask the predictor and we simply use the standard deterministic strategy to obtain a competitive ratio of 2. Otherwise rent until t^*, and we ask the predictor on day $t^* + 1$, and then continue to rent if $d^* - t^* \leq b$ and buy otherwise.

If $d^* \leq t^*, d^* \leq b$, then we get the optimal solution.

If $d^* > t^*$, and $d^* \leq b$, then our ratio is $\frac{d^*+c}{d^*}$ which is maximized when $d^* = t^* + 1$. Worst case guarantee is $\frac{t^*+1+c}{t^*+1}$.

If $d^* > b$, but $d^* - t^* \leq b$, then our ratio is $\frac{d^*+c}{b}$ which is maximized when $d^* = t^* + b$. Worst case guarantee is $\frac{t^*+b+c}{b}$.

If $d^* > b$, but $d^* - t^* > b$, then our ratio is $\frac{t^*+b+c}{b}$.

We want to minimize the maximum of $\frac{t^*+1+c}{t^*+1} = 1 + \frac{c}{t^*+1}$ and $\frac{t^*+b+c}{b} = 1 + \frac{t^*+c}{b}$. Equating these two, we can solve for the required t^* and substituting this we get the competitive ratio.

We get the competitive ratio to be exactly:

$$1 + \frac{\sqrt{(c+1)^2 + 4c(b-1)} + c - 1}{2b} \tag{13}$$

Thus , for this to be less than 2, we can enforce the above ratio to be less than 2 and this gives us the optimal θ. Now we show that this will give us the optimal deterministic algorithm.

Let (t, θ), be the parameters of any deterministic algorithm $ALG(t, \theta)$, such that if $c < \theta b$, this algorithm asks for the prediction at the start of day t, otherwise buys at the start of day b (the optimal deterministic strategy when there is no prediction, which is 2 competitive). If an algorithm asks for the prediction, then from the aforementioned proof, it is clear that the competitive ratio is minimized with the particular t^*. Now, let us set $c = \theta^{'} b$. Then from Equation 13, the competitive ratio (when the prediction is obtained) is specified by some function $g(b, \theta^{'})$. It is easy to see that this function is increasing in $\theta^{'}$. We claim that setting $g(b, \theta^{'}) = 2$, gives us θ^*, the optimal threshold. Now, consider a different threshold $\theta^{''}$. If $\theta^{''} < \theta^*$, then for all costs, such that $\theta \leq \theta^{''}$, both algorithms will have the same competitive ratio. But for $\theta^{''} < \theta < \theta^*$, $ALG(t^*, \theta^*)$ will have competitive ratio which is less than 2. However, $ALG(t, \theta^{''})$ will be 2 competitive, for any t. A similar argument can be made if $\theta^{''} > \theta^*$ and thus the optimal deterministic algorithm is given by (t^*, θ^*).

□

A.2 Modifying ToP for error-prone predictions

The main modification we make is that, when ToP asks for a prediction and say y is the prediction, we buy at the start of $t^* + \lceil \lambda b \rceil$ from the day of *asking the predictor*, if $y \geq t^* + b$, else we buy at the start of $t^* + \lceil b/\lambda \rceil$, if $y < t^* + b$, exactly as prescribed by Purohit et al. (2018) and similarly define $\eta = |x - y|$, where x is the true number of ski-days.

Following the analysis of Theorem 2.2 in Purohit et al. (2018), we can show that: The robustness bound in this case is: ALG $\leq (\frac{t^*+c}{b} + 1 + 1/\lambda)$OPT and this can be upper bounded by :

For the smoothness, we get the following: ALG $\leq (\frac{t^*+c}{b} + 1 + \lambda)(\text{OPT} + \eta)$ and ALG $\leq (\frac{t^*+c}{b} + 1)\text{OPT} + \frac{\eta}{1-\lambda}$. Now, putting the above two conditions together, we get that: ALG $\leq (\frac{t^*+c}{b} + 1 + \lambda)\text{OPT} + \frac{\eta}{1-\lambda}$. Finally, we get that:

$$\frac{\text{ALG}}{\text{OPT}} \leq \min\{\frac{t^*+c}{b} + 1 + 1/\lambda, \frac{t^*+c}{b} + 1 + \lambda + \frac{\eta}{(1-\lambda)\text{OPT}}\}.$$

Here $1 + \frac{t^*+c}{b}$ is exactly given by the expression in 13.

A.3 Upper Bound Proof

The full threshold function $f^*(\mu, \sigma, b)$, mentioned in the paper is given by:

When $\mu \geq b$

$$f^*(\mu,\sigma,b) = \begin{cases} \frac{1}{2}\left(\sqrt{\sigma^2+(\mu-b)^2}-(\mu-b)\right), & \text{if } \mu \in [b, b+\sqrt{b^2-\sigma^2}] \text{ and } \sigma \leq b \\ \dfrac{b}{1+\frac{\mu^2}{\sigma^2}}, & \text{otherwise} \end{cases} \tag{14}$$

When $\mu < b$,

$$f^*(\mu,\sigma,b) = \begin{cases} \dfrac{1}{2}\left(\sqrt{\sigma^2+(b-\mu)^2}-(b-\mu)\right), & \text{if } \; b-\sqrt{b^2-\sigma^2} \leq \mu < b \text{ and } \sigma \leq b \\ \mu - \dfrac{b}{\frac{\sigma^2}{\mu^2}+1}, & \text{otherwise} \end{cases} \tag{15}$$

Full proof of Lemma 2.5. First we assume that we can work with distributions, that are supported at exactly two points and later show that this is sufficient. One can represent a *two-point* distribution as $\mu - k_1\sigma$ and $\mu + k_2\sigma$, where $k_1 > 0$ and $k_2 > 0$, that has mean μ and variance σ^2.

Let $p_1 > 0$ and $p_2 > 0$, be such that, the following equations hold:

$$p_1(\mu - k_1\sigma) + p_2(\mu + k_2\sigma) = \mu \tag{16}$$
$$p_1(k_1\sigma)^2 + p_2(k_2\sigma)^2 = \sigma^2 \tag{17}$$
$$p_1 + p_2 = 1 \tag{18}$$

Then solving the above set of equations gives us $k_2 = \sqrt{\dfrac{p_1}{p_2}}$ and $k_1 = \dfrac{1}{k_2}$. Thus setting $\dfrac{p_1}{p_2} = l$, we can parameterize all such two-point distributions by l.

First consider $\mu \geq b$.

BoR always pays b, when $\mu \geq b$, thus let us define OPT(l) to be the cost of OPT on the two-point distribution $p_{\mu - \frac{\sigma}{\sqrt{l}}} = \dfrac{l}{l+1}$ and $p_{\mu+\sqrt{l}\sigma} = \dfrac{1}{l+1}$.

To get meaningful upper bounds we are interested in the case when $\mu - \frac{\sigma}{\sqrt{l}} \leq b$, otherwise, OPT pays b and BoR $=$ OPT. Also, the point $\mu - \frac{\sigma}{\sqrt{l}} \geq 0$. Thus we can define the feasible set for the optimization problem as the following:

$$l_f := \left\{ l : \frac{\sigma}{\mu} \leq \sqrt{l} \leq \frac{\sigma}{\mu - b} \right\} \tag{19}$$

If $\mu = b$, then , $\mu - \frac{\sigma}{\sqrt{l}} < b$, as $\sigma > 0$. Thus we only require that $\sqrt{l} \geq \frac{\sigma}{\mu}$.

To find the upper bound on BoR $-$ OPT $:= \max_{l_f}(\text{BoR} - \text{OPT}(l))$, where,

$$(\text{BoR} - \text{OPT}(l)) = \frac{\sigma\sqrt{l} - (\mu - b)l}{l+1} \tag{20}$$

This is because OPT $= \frac{l}{l+1}(\mu - \frac{\sigma}{\sqrt{l}}) + \frac{1}{l+1}b$ and BoR pays b in this setting.

Thus, the max value attained is given by (see Claim A.1), $\max_{l_f}(\text{BoR} - \text{OPT}(l)) =$

$$\begin{cases} \frac{1}{2}\left(\sqrt{\sigma^2 + (\mu-b)^2} - (\mu - b)\right), \text{ if} \\ \qquad\qquad \mu \in [b, b+\sqrt{b^2-\sigma^2}] \text{ and } \sigma \leq b \\ \dfrac{b}{1+\frac{\mu^2}{\sigma^2}}, \quad \text{otherwise} \end{cases} \tag{21}$$

Now let us consider $\mu < b$.

Our algorithm rents as long as $\frac{\sigma}{\mu} < \frac{b-\mu}{\sigma}$ or if ($\frac{\sigma}{\mu} \geq \frac{b-\mu}{\sigma}$ and $\sigma < \mu$).

$$\max_{l_f} \frac{\sigma\sqrt{l} - (b-\mu)}{l+1} \tag{22}$$

Here, the feasible set is considering the following cases: When $\mu + \sqrt{l}\sigma \geq b$ (otherwise OPT pays μ, same as BoR) and such that $\mu - \frac{\sigma}{\sqrt{l}} \geq 0$. Combining the two conditions, we have:

$$l_f := \left\{ l : \sqrt{l} \geq \max\left\{ \frac{\sigma}{\mu}, \frac{b-\mu}{\sigma} \right\} \right\} \tag{23}$$

See Claim A.2, for the details of the above optimization problem.

Finally, we get the maximum value attained, which is:

$$\max_{\sqrt{l} \geq \max\left\{\frac{\sigma}{\mu}, \frac{b-\mu}{\sigma}\right\}} \frac{\sigma\sqrt{l} - (b-\mu)}{l+1} = \begin{cases} \dfrac{1}{2}\left(\sqrt{\sigma^2 + (b-\mu)^2} - (b-\mu)\right), & \text{if } b - \sqrt{b^2-\sigma^2} \leq \mu \leq b \text{ and } \sigma \leq b \\ \mu - \dfrac{b}{\frac{\sigma^2}{\mu^2}+1}, & \text{otherwise} \end{cases} \tag{24}$$

Now, to see why it suffices to consider only *two-point* distributions. Note that $\text{OPT} = \mathbb{E}\left[\mathbf{1}\{X < b\}X\right] + b\mathbb{E}\left[\mathbf{1}\{X \geq b\}\right]$. The difficult situation is when there are support points both above and below b. We can set $\frac{l}{l+1}(\mu - \sigma/\sqrt{l}) = \mathbb{E}\left[\mathbf{1}\{X < b\}X\right]$. This can be used to solve for a unique l^*. Now place the second point that is $\mu + \sigma\sqrt{l^*}$ and this leaves OPT, μ and σ unchanged. In addition, our algorithm either buys at day 0, when $\mu \geq b$, and either rents forever when $\mu < b$. In these cases the cost of the algorithm depends only on μ and b,independent of the distribution that is considered.

Finally, it is easy to see that when we take the minimum of $\text{BoR} \leq \text{OPT} + f^*$, and $\text{PRED} = \text{OPT} + c$, we get the required upper bound. ☐

A.4 Proof of Claims in Section 2

Claim A.1. It holds that:

$$\max_{\frac{\sigma}{\mu} \leq \sqrt{l} \leq \frac{\sigma}{\mu - b}} \frac{\sigma\sqrt{l} - (\mu-b)l}{l+1} \leq \frac{1}{2}\left(\sqrt{\sigma^2 + (\mu-b)^2} - (\mu-b)\right)$$

Proof. Now consider,

$$\max_{\frac{\sigma}{\mu} \leq \sqrt{l} \leq \frac{\sigma}{\mu - b}} \frac{\sigma\sqrt{l} - (\mu-b)l}{l+1}. \tag{25}$$

Set $\sqrt{l} = \frac{\sigma}{\mu - t}$, such that $0 \leq t \leq b$. Then, $\max_{0 \leq t \leq b} \dfrac{b-t}{1 + \frac{(\mu-t)^2}{\sigma^2}}$, Then set $x = \mu - t$, such that $\mu - b \leq x \leq \mu$, we have finally:

$$\max_{\mu - b \leq x \leq \mu} \frac{x - (\mu - b)}{1 + \frac{x^2}{\sigma^2}} \tag{26}$$

When $\mu \geq b$, we have that the unconstrained maximum above gives us exactly $\dfrac{1}{2}\left(\sqrt{\sigma^2 + (\mu-b)^2} - (\mu - b)\right)$, when certain conditions on μ and σ are satisfied, the $x^* = \mu - b + \sqrt{\sigma^2 + (\mu-b)^2}$. This means that the maximum value is given as follows, substituting x^* in the above expression, we have the following:

$$\frac{\sqrt{\sigma^2 + (\mu-b)^2}}{1 + \frac{((\mu-b) + \sqrt{\sigma^2 + (\mu-b)^2})^2}{\sigma^2}} \tag{27}$$

Now, in the above equation, set $y = \sqrt{\sigma^2 + (\mu-b)^2}$ and therefore, $\mu - b = \sqrt{y^2 - \sigma^2}$.Now the maximum value can be written as:

$$\begin{aligned}
&\frac{y}{1 + \dfrac{(\sqrt{y^2 - \sigma^2} + c)^2}{\sigma^2}} && (28)\\
&= \frac{1}{2}\frac{\sigma^2 y}{y^2 + y\sqrt{y^2 - \sigma^2}} && (29)\\
&= \frac{1}{2}\frac{\sigma^2}{y + \sqrt{y^2 - \sigma^2}} && (30)\\
&= \frac{1}{2}(y - \sqrt{y^2 - \sigma^2}) && (31)\\
&= \frac{1}{2}\left(\sqrt{\sigma^2 + (\mu-b)^2} - (\mu - b)\right) && (32)
\end{aligned}$$

Otherwise, the maximum occurs on the boundary point that is $x^* = \mu$ and this gives us $\dfrac{b}{1 + \frac{\mu^2}{\sigma^2}}$, but this maximum value has to be less than the unconstrained max and hence we get the bound above. The reason for this is that the function above is increasing until $\mu - b + \sqrt{\sigma^2 + (\mu-b)^2}$. Suppose, $\mu - b + \sqrt{\sigma^2 + (\mu-b)^2} > \mu$, then the maxima is attained at $x^* = \mu$. □

Claim A.2. It holds that:

$$\max_{\sqrt{l} \geq \max\{\frac{\sigma}{\mu}, \frac{b-\mu}{\sigma}\}} \frac{\sigma\sqrt{l} - (b - \mu)}{l + 1} \leq \frac{1}{2}\left(\sqrt{\sigma^2 + (b-\mu)^2} - (b - \mu)\right)$$

Proof. First let us look at the condition when $\frac{\sigma}{\mu} \geq \frac{b-\mu}{\sigma}$. Then,

$$\mu^2 - \mu b + \sigma^2 \geq 0 \tag{33}$$

Solving for the above gives us that $\mu \in (0, \frac{b - \sqrt{b^2 - 4\sigma^2}}{2}) \bigcup (\frac{b + \sqrt{b^2 - 4\sigma^2}}{2}, b)$, if, $\sigma < b/2$. Also, if $\sigma \geq b/2$, then the above equation is always satisfied.

Let us assume that $\sigma \geq b/2$ and thus $\frac{\sigma}{\mu} \geq \frac{b-\mu}{\sigma}$. Thus the optimization problem becomes:

$$\max_{x \geq \frac{\sigma}{\mu}} \frac{\sigma x - (b - \mu)}{x^2 + 1} \tag{34}$$

Now, we have to note that the above function is increasing in the interval $x \in [\frac{b - \mu - \sqrt{(b-\mu)^2 + \sigma^2}}{\sigma}, \frac{b - \mu + \sqrt{(b-\mu)^2 + \sigma^2}}{\sigma}]$, with the unconstrained maximum at $x^* = \frac{b - \mu + \sqrt{(b-\mu)^2 + \sigma^2}}{\sigma}$. Now, we know that $\frac{\sigma}{\mu} \geq \frac{b-\mu}{\sigma}$, but additionally, if $\frac{\sigma}{\mu} >$

$\frac{b-\mu}{\sigma} + \frac{\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}$, then the maximum occurs at the boundary, i.e, $x^* = \frac{\sigma}{\mu}$. Now the latter condition holds when

$$\mu^2 - 2\mu b + \sigma^2 > 0 \tag{35}$$

This implies that $\mu < b - \sqrt{b^2 - \sigma^2}$, if $\sigma \leq b$. But the above equation is true if $\sigma > b$, regardless of the value of μ. When $\sigma < b/2$, we have the same conditions for $\frac{\sigma}{\mu} > \frac{b-\mu}{\sigma} + \frac{\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}$ and $\frac{\sigma}{\mu} > \frac{b-\mu}{\sigma}$. But in addition, we might have $\frac{b-\mu}{\sigma} > \frac{\sigma}{\mu}$.

Thus we can say that the maximum is attained in the interior point ($x^* = \frac{b-\mu+\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}$), if $\frac{\sigma}{\mu} < \frac{b-\mu}{\sigma}$ or $\frac{b-\mu}{\sigma} \leq \frac{\sigma}{\mu} \leq \frac{b-\mu}{\sigma} + \frac{\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}$.Combining these two we get that, as long as $\frac{\sigma}{\mu} \leq \frac{b-\mu}{\sigma} + \frac{\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}$ is satisfied, we have the maxima at the interior.

But, when we have $\sigma < b/2$, we note the following fact:

$$\left(\frac{b - \sqrt{b^2 - 4\sigma^2}}{2}, \frac{b + \sqrt{b^2 - 4\sigma^2}}{2}\right) \subseteq (b - \sqrt{b^2 - \sigma^2}, b) \tag{36}$$

Thus we get the following cases:

$$\underset{\sqrt{l} \geq \max\{\frac{\sigma}{\mu}, \frac{b-\mu}{\sigma}\}}{\arg\max} \frac{\sigma\sqrt{l} - (b-\mu)}{l+1} = \begin{cases} \frac{b-\mu+\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}, & \text{if } b - \sqrt{b^2-\sigma^2} \leq \mu \leq b \text{ and } \sigma \leq b \\ \frac{\sigma}{\mu}, & \text{otherwise} \end{cases} \tag{37}$$

This gives us the required maximum values. The maximum is calculated using the exact same technique as in Claim A.1, i.e, set $y = \sqrt{(b-\mu)^2 + \sigma^2}$.

Finally, we get the maximum value attained, which is:

$$\max_{\sqrt{l} \geq \max\{\frac{\sigma}{\mu}, \frac{b-\mu}{\sigma}\}} \frac{\sigma\sqrt{l} - (b-\mu)}{l+1} = \begin{cases} \frac{1}{2}\left(\sqrt{\sigma^2 + (b-\mu)^2} - (b-\mu)\right), & \text{if } \ b - \sqrt{b^2-\sigma^2} \leq \mu \leq b \text{ and } \sigma \leq b \\ \mu - \frac{b}{\frac{\sigma^2}{\mu^2}+1}, & \text{otherwise} \end{cases} \tag{38}$$

□

A.5 Lower Bound Proof

Proof of Lemma 2.6. We begin with $\mu = b$. Now consider the distribution supported at points $p_{b-\sigma} = \frac{1}{2}$ and $p_{b+\sigma} = \frac{1}{2}$.

If $d \leq b - \sigma$, we have that

$$\begin{aligned} \mathrm{C}(d) - \mathrm{OPT} &= d + b - (\frac{1}{2}(b-\sigma) + \frac{1}{2}b) && (39)\\ &= d + \frac{\sigma}{2} && (40)\\ &\geq \frac{\sigma}{2} && (41) \end{aligned}$$

If $d > b - \sigma$, then, we have:

$$\begin{aligned} \mathrm{C}(d) - \mathrm{OPT} &= (\frac{1}{2}(b-\sigma) + \frac{1}{2}(d+b)) - (\frac{1}{2}(b-\sigma) + \frac{1}{2}b) && (42)\\ &= \frac{d}{2} && (43)\\ &\geq \frac{b-\sigma}{2} && (44)\\ &\geq \frac{\sigma}{2} && (45) \end{aligned}$$

The last inequality just follows from the fact that $b - \sigma > \sigma$, when $\sigma = o(b)$.

For $\mu > b$. We provide our choices of $\sqrt{l}$, below.

$$\sqrt{l} = \begin{cases} \frac{\sigma}{\mu - b + \frac{1}{2}\sqrt{\sigma^2 + (\mu-b)^2}}, & \text{if } \mu \in (b, b + \sqrt{b^2 - \sigma^2}] \\ \frac{\sigma}{\mu - \sigma}, & \text{if } \mu > b + \sqrt{b^2 - \sigma^2} \end{cases} \tag{46}$$

Let us see the lower bound for the first case:

If $\mu \in (b, b + \sqrt{b^2 - \sigma^2}]$ and $\sigma \leq \mu - b$, then, we use the following distribution: Choose $\sqrt{l} = \frac{\sigma}{\mu - b + \frac{1}{2}\sqrt{\sigma^2 + (\mu-b)^2}}$. This makes one point of the support at $b - \frac{1}{2}\sqrt{\sigma^2 + (\mu-b)^2}$. The above conditions ensure that this is valid, i.e, we have that $b \geq \sqrt{\sigma^2 + (\mu-b)^2}$. If $d \leq b - \frac{1}{2}\sqrt{\sigma^2 + (\mu-b)^2}$:

$$\mathrm{C}(d) - \mathrm{OPT} \geq \frac{0.5\sqrt{\sigma^2 + (\mu-b)^2}}{1 + \frac{((\mu-b) + 0.5\sqrt{\sigma^2 + (\mu-b)^2})^2}{\sigma^2}} \tag{47}$$

$$\geq \frac{0.5 * \sqrt{\sigma^2 + (\mu-b)^2}}{1 + \frac{((\mu-b) + \sqrt{\sigma^2 + (\mu-b)^2})^2}{\sigma^2}} \tag{48}$$

$$= \frac{1}{4}(\sqrt{\sigma^2 + (\mu-b)^2} - (\mu - b)) \tag{49}$$

The above bound follows from steps that are used to simplify Equation (27). If $d > b - \frac{1}{2}\sqrt{\sigma^2 + (\mu-b)^2}$:

$$\mathrm{C}(d) - \mathrm{OPT} \geq \frac{b - 0.5\sqrt{\sigma^2 + (\mu-b)^2}}{1 + \frac{\sigma^2}{((\mu-b) + 0.5\sqrt{\sigma^2 + (\mu-b)^2})^2}} \tag{50}$$

$$\geq \frac{b - 0.5\sqrt{\sigma^2 + (\mu-b)^2}}{1 + \frac{\sigma^2}{(\mu-b)^2}} \tag{51}$$

$$\geq \frac{b - 0.5\sqrt{\sigma^2 + (\mu-b)^2}}{2} \tag{52}$$

$$\geq \frac{1}{4}\sqrt{\sigma^2 + (\mu-b)^2} \tag{53}$$

$$\geq \frac{1}{4}\sqrt{\sigma^2 + (\mu-b)^2} - \frac{1}{4}(\mu - b) + \frac{1}{4}(\mu - b) \tag{54}$$

Also, the renting strategy is lower bounded by buying at day 0. Thus, $\mathrm{C}(d) - \mathrm{OPT} \geq \frac{1}{2} f^*(\mu, \sigma, b)$.

When $\mu > b + \sqrt{b^2 - \sigma^2}$, we will use the following distribution: $p_{\mu - \frac{\sigma}{\sqrt{l}}} = \sigma$, i.e, $\sqrt{l} = \frac{\sigma}{\mu - \sigma}$. Now, we have for different buying strategies: Let $d \leq \sigma$, then:

$$\begin{aligned} C(d) - \text{OPT} &= d + \frac{(b-\sigma)\sigma^2}{\sigma^2 + (\mu - \sigma)^2} && (55)\\ &\geq \frac{(b-\sigma)\sigma^2}{\sigma^2 + \mu^2} && (56)\\ &\geq \frac{1}{2}\frac{b\sigma^2}{\sigma^2 + \mu^2} && (57) \end{aligned}$$

We can use the fact that $b - \sigma > \sigma$. Now, when $d > \sigma/2$, we have:

$$\begin{aligned} C(d) - \text{OPT} &= \frac{d(\mu-\sigma)^2}{\sigma^2 + (\mu-\sigma)^2} \\ &\geq \frac{\sigma(\mu-\sigma)^2}{\sigma^2 + \mu^2} \\ &\geq \frac{\sigma^2 b}{\sigma^2 + \mu^2} \end{aligned} \qquad (58)$$

The last line is due to the fact that if $\mu - \sigma > \sigma$ and $\mu - \sigma > b$, as $\sigma < \mu - b$, both are due to the fact that $\mu = \Theta(b)$, while $\sigma = o(b)$.

Note, that the renting strategy clearly is lower bounded by the buying at day 0 strategy since $\mu > b$ and thus all the above lower bounds carry over. Again, we have that $\text{C}(d) - \text{OPT} \geq \frac{1}{2} f^*(\mu, \sigma, b)$, in this case as well.

For $\mu < b$, if $\sigma = o(b)$ and $\mu = \theta(b)$, we always have that $\mu \geq b - \sqrt{b^2 - \sigma^2}$ and $\sigma < \mu$, i.e, we have that $\frac{\sigma}{\mu} < \frac{b-\mu}{\sigma} < \frac{b-\mu}{\sigma} + \frac{\sqrt{(b-\mu)^2 + \sigma^2}}{\sigma}$. Also, we may assume that $\sigma < b - \mu$.

When $\sigma \leq (b - \mu)$, use $\sqrt{l} = \frac{b-\mu}{\sigma} + 2\frac{\sqrt{(b-\mu)^2+\sigma^2}}{\sigma}$

When renting, we have:

$$\begin{aligned} C(d) - \text{OPT} &= \frac{\mu + \sigma\sqrt{l} - b}{l+1} \\ &= \frac{2\sqrt{(b-\mu)^2 + \sigma^2}}{\frac{(b-\mu+2\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2} + 1} \\ &\geq \frac{1}{4}\frac{2\sqrt{(b-\mu)^2 + \sigma^2}}{\frac{(b-\mu+\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2} + 1} \\ &\frac{1}{4}(\sqrt{(b-\mu)^2+\sigma^2} - (b-\mu)) \end{aligned} \qquad (59)$$

When, $d > \mu - \sigma/\sqrt{l}$, we have:

$$
\begin{aligned}
C(d) - \text{OPT} &= \frac{d}{l+1} \\
&= \frac{1}{l+1}(\mu - \frac{\sigma^2}{b-\mu+2\sqrt{(b-\mu)^2+\sigma^2}}) \\
&= \frac{1}{\frac{(b-\mu+2\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2}+1}(\mu - \frac{\sigma^2(2\sqrt{(b-\mu)^2+\sigma^2}-(b-\mu))}{3(b-\mu)^2+4\sigma^2}) \\
&\geq \frac{1}{\frac{(b-\mu+2\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2}+1}(\mu - \frac{\sigma^2(2\sqrt{(b-\mu)^2+\sigma^2}-(b-\mu))}{3(b-\mu)^2+4\sigma^2}) \\
&\geq \frac{1}{\frac{(b-\mu+2\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2}+1}(\mu + \frac{b-\mu}{7} - \frac{2\sqrt{(b-\mu)^2+\sigma^2}}{7}) \\
&\geq \frac{1}{\frac{(b-\mu+2\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2}+1}(\frac{6b}{7} - \frac{2\sqrt{(b-\mu)^2+\sigma^2}}{7}) \\
&\geq \frac{1}{\frac{(b-\mu+\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2}+1}\frac{4\sqrt{(b-\mu)^2+\sigma^2}}{28} \\
&\frac{1}{7}\left(\frac{\sqrt{(b-\mu^2)+\sigma^2}}{\frac{(b-\mu+\sqrt{(b-\mu)^2+\sigma^2})^2}{\sigma^2}+1}\right) \\
&\geq \frac{1}{14}(\sqrt{(b-\mu)^2+\sigma^2}-(b-\mu))
\end{aligned}
\tag{60}
$$

We used that $\mu > b - \sqrt{b^2-\sigma^2}$, $b \geq \sqrt{(b-\mu^2)+\sigma^2}$ and $\sigma \leq (b-\mu)$. Finally, the reduction to the lower bound is following the steps in Equation (27). The strategy of buying at day d, that is less than the first support point, is lower bounded by the cost of renting since $b > \mu$ and hence we are done.

Finally, we have $\text{C}(d) - \text{OPT} \geq \frac{1}{7} f^*(\mu, \sigma, b)$, combining all the above cases.

□

A.6 Modifying VoP for error-prone predictions

Firstly, observe that, the algorithm that decides the threshold for asking the predictor depends only on b, the mean (μ) and the standard deviation σ and not on the error of the predictor. Recall that the algorithm PRED asks the prediction after paying cost c. Now, we have different buying strategies , denoted by d. Now if the algorithm PRED, after paying a cost c, buys at day d, then it pays the following:

$$\text{PRED}(d) := \mathbb{E}\left[\mathbf{1}\{X \leq d\}X + (d+b)\mathbf{1}\{X > d\}\right]. \tag{61}$$

Let d^*, be the true buying strategy that minimizes the following,

$$\text{PRED}(d^*) := \min_{d \geq 0} \text{PRED}(d). \tag{62}$$

Also recall that

$$\text{OPT} := \mathbb{E}\left[\min\{X, b\}\right].$$

We assume that the prediction gives a buying strategy $\hat{d}$. Then, we consider the following algorithm: If $\hat{d} < b$, then buy at $\max\{\lambda b, \hat{d}\}$, otherwise buy at $\min\{\hat{d}, b/\lambda\}$. Let $|\hat{d} - d^*| = \eta$.

We will compare our algorithm to $\text{PRED}(d^*)$ and note that $\text{PRED}(d^*) \leq \frac{e}{e-1}\text{OPT}$. Considering different cases of operation of our algorithm, the consistency in the worst case is $\text{PRED} \leq c + \max\{\min\{\mu, \lambda b\}, \min\{\lambda\mu, b\}\} + \text{OPT}$ and the worst case robustness is $\text{PRED} \leq c + \frac{\min\{\mu, b\}}{\lambda} + \text{OPT}$.

Now, this implies, we recover the well known consistency and robustness when the true distribution is a point distribution, we get consistency, which is PRED $\leq c + (1+\lambda)$OPT and robustness which is PRED $\leq c + (1+\frac{1}{\lambda})$OPT.

B Missing Proofs in Section 3.1

Proof of Theorem 3.2. Again, we proceed in the same fashion as in Theorem 3.1. Let $\mathcal{X}$ be the following family of instances. Fix T, and take $B = T, N = T, \beta = 0$. Add to $\mathcal{X}$ an instance U with ticket purchases of $2B/3$ on day 0 and no other ticket purchases. We call U the *uniform instance*. Next, for each day $i = 1, \dots, T-1$, add to $\mathcal{X}$ the instance that purchases a ticket of cost $2B/3$ on day i, a ticket of cost $2B/3$ on day 0 and no other tickets. We call such an instance, instance i. Note that the only $(1+\epsilon)$-approximation for the uniform instance purchases no cards, but the only $(1+\epsilon)$-approximation for the remaining instances purchases a card immediately. To distinguish the uniform instance from all other members of $\mathcal{X}$, $N-1$ queries are required. To distinguish instance i from all other instances we must query day i. Hence, we can again represent any deterministic algorithm as a sequence of days we query that terminates once it has determined which instance it has been presented with.

Consider the distribution over $\mathcal{X}$ that selects the uniform instance with probability $1/2$ and the remaining instances with equal probability.

Let Q_A be the random variable that takes on the number of queries required by an algorithm A over the uniform distribution of instances over $\mathcal{X}$. Suppose that we draw an instance X from our probability distribution. Suppose that A makes at most $N/2$ queries. Then with probability at least $3/4$ A has not determined with certainty which instance X is. Thus the expected number of queries is $\Omega(N)$ as required. We again apply Yao's principle to obtain the desired result.

□

Proof of Theorem 3.3. Fix T and and a value $c > 0$. Let $N = c(3T+1)$, select some $B > 0$ sufficiently large and take $\beta = 0$.

Construct the following family $\mathcal{X}$ of instances. For each $j = 1, \dots, c$, there are ticket purchases on day $(j-1)(3T+1)+T$ of cost $2B/3$. There are additional ticket costs of $2B/3$ that fall either on day $(j-1)(3T+1)+2T-1$ or day $(j-1)(3T+1)+2T$ each with probability $1/2$. All other days have ticket costs 0.

Note that by our construction, there is a unique optimal solution for each instance $X \in \mathcal{X}$. In particular, for $j = 1, \dots, c$ let section j be time period $[(j-1)(3T+1), j(3T+1))$. If X has ticket costs on day $(j-1)(3T+1)+2T-1$, then $OPT(X)$ buys one card in section j on day $(j-1)(3T+1)+T$. Otherwise it buys no cards in section j. In addition, any solution that differs from the optimal one in interval j for $j = 1, \dots c$ incurs an additional cost of at least $B/3$. Hence if a solution S differs from $OPT(X)$ in $\Omega(N/T)$ sections it does not give a $(1+\epsilon)$-approximation. To see this note that $opt(\mathcal{X}) \leq \frac{4B}{3}c$ for any instance X. Such a solution S will have cost at least $opt(\mathcal{X}) + \Omega(\frac{NB}{T})$, and thus the approximation ratio is $(1+\Omega(N/T))$.

Note that with any given query an algorithm can determine whether or not to buy a card in at most two sections. Thus by Yao's principle any randomized algorithm requires $\Omega(N/T)$ queries to obtain a $(1+\epsilon)$-approximation .

□

C Missing Proofs in Section 3.2

Proof of Theorem 3.4. We construct $G = (V, E)$ as follows. See Figure 2 for a visualization. The vertex set V is indexed by $\{0, \dots, N\}$. For each day $i = 0, \dots, N-1$ we add such an arc $(i, i+1)$ to E of length c_i. In addition, for each day $i = 0, \dots, N-T$ we create an arc $(i, i+T)$ to E of weight $B + \sum_{j=i}^{i+T-1} \beta c_i$. For days $i = N-T+1, \dots, N-1$ we create an arc (i, N) to E of weight $B + \sum_{j=i}^{N} \beta c_i$. We take s to be 0 and t to be N. If an s,t-path contains an arc $(i, i+1)$ of the first type, then in the corresponding solution to the Bahncard instance there is no valid card during time i. If an s,t-path contains an arc $(i, i+T)$ or (i, N) of the second type, then the corresponding solution purchases a card at time i. Given a shortest s,t-path in G, it is easy to reconstruct an optimal solution to $\mathcal{I}$.

We note here that we can add extra arcs to our construction, so that we need not assume the validity periods of our cards are disjoint. □

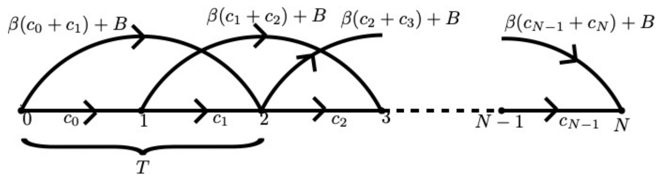

Figure 2: Construction of the DAG used to solve for the optimal offline solution for a Bahncard instance. In this instance we have $T = 2$.

Proof of Observation 3.5. Let V be the same vertex set as in the proof of Theorem 3.4. Instead only add arcs $(i, i+P)$ of weight $B + \sum_{i'=i}^{i+P-1} \beta c_i$, for each $(i, P) \in \mathcal{P}$. In the case that $i + P$ extends past N, we would add the arc (i, N) instead. Then since we do not know the costs c_i for each day we instead of adding arcs of the form $(i, i+1)$ for each day we add arcs (i, i') of cost $\sum_{d=i}^{i'} c_d$ for each minimal fully-determined sub-interval in our lifetime. $\square$

Proof of Fact 3.6. Equation (10) follows from the observation that one can construct a solution S to $\mathcal{I}$ with a valid card during each day that $OPT(\mathcal{I}')$ has a valid card by purchasing at most double the number of cards. Also, given $OPT(\mathcal{I})$ we can easily construct a solution S' to $\mathcal{I}'$ that purchased the same number of cards, and has valid cards whenever $OPT(\mathcal{I})$ has valid cards.

To see that $OPT(\mathcal{I})$ can be assumed to purchase at most $2b$ cards, let $J = [a, b)$ be any maximal interval in time where at any point in time $t \in (a, b)$ there exists either a purchase made by $OPT(\mathcal{I}')$ or by $OPT(\mathcal{I})$ at time τ such that $t \in (\tau, \tau + T)$. Suppose that during J $OPT(\mathcal{I}')$ has b_J valid cards, but $OPT(\mathcal{I})$ has $a_J = 2b_J + i$ cards for $i > 0$. Then there must exist at least $2Ti$ days in J during which $OPT(\mathcal{I})$ has a valid Bahncard but $OPT(\mathcal{I}')$ whose total ticket costs together exceed pi, for $p = \frac{B}{1-\beta}$. But then we can find an improved solution for instance $\mathcal{I}$ by purchasing at most p additional cards and shifting the original purchases of $OPT(\mathcal{I}')$ during J, which is a contradiction. $\square$

Proof of Theorem 3.7. Recall that b^* is the number of cards purchased by $OPT(\mathcal{I}^*)$. We take b^* iterations to modify $OPT(\mathcal{I})$ into a solution S of $\mathcal{I}^*$ at a loss of at most $\frac{g\gamma}{b}$ at each iteration. Since $b^* \leq b$, this will give us the desired approximation guarantee. Recall b from Equation 10.

Let $\tau_1, \ldots, \tau_{b^*}$ be the purchase times of $OPT(\mathcal{I})$. We construct solutions $S_0 = OPT(\mathcal{I}), S_1, \ldots, S_{b^*} = S$ such that for $j = 1, \ldots, b^*$ the solution S_j makes purchases at $\tau'_1, \ldots, \tau'_j, \tau_{j+1}, \ldots \tau_{b^*}$, where $\tau'_1 \in D^1, \ldots \tau'_j \in D^j$. We will prove inductively that the cost of S_j is at most

$$OPT(\mathcal{I}) + j\frac{g\gamma}{b} \leq (1+\gamma)OPT(\mathcal{I}), \tag{63}$$

which gives us our desired result.

Clearly the statement holds for $j = 0$. Suppose now that $j > 1$ and the statement holds for S_{j-1}. Then take τ'_j to be the largest of $\tau'_{j-1} + T$ and the earliest point in D^j at least τ_j. In the former case S_j is at least as good at S_{j-1} and hence Equation (63) holds. In the latter case there must exist $d, d' \in$ Construct-Starting-Points$(\tau'_{j-1} + T) \subseteq D^j$ such that $\tau_j \in [d, d')$. To see this if e is the latest point in Construct-Starting-Points$(\tau'_{j-1} + T)$, and $e < N$, then $[\tau'_{j-1} + T, e)$ has ticket costs exceeding g and hence $OPT(\mathcal{I})$ must have purchased a card valid during this time period, the earliest of which must be τ_j. Hence $[\tau_j, \tau'_j)$ is γ-light and we obtain Equation (63). Note that it is possible that $\tau_j = \tau'_j$, in which case the statement holds trivially.

To see the bound on the number of queries we note that if q, q', q'' are consecutive elements in Construct-Starting-Points(s) for some $s \in [0, N)$, then by construction $[q, q'']$ is γ-heavy. Thus it must be the case that for any set $D_s =$ Construct-Starting-Points(s), we have that $|D_s| - 1 \leq \frac{2b}{\gamma}$. As a result there will be $O\left(\left(\frac{2b}{\gamma}\right)^j\right)$ points in D^j, for each $j = 1, \ldots, b$. In this case our query bound will be $O\left(\left(\frac{2b}{\gamma}\right)^b \log(T)\right)$. $\square$

Figure 3: A visual representing two instances $\mathcal{I}_1, \mathcal{I}_2$ both having time horizon $T + \epsilon T$. The ticket costs for $\mathcal{I}_1$ and $\mathcal{I}_2$ are represented by the green and red circles respectively. A circle represents ticket costs of B, otherwise there are no ticket costs.

C.1 Modification of Algorithm for Large b

If our estimate b of b^* is very large, we can instead take a constant $\tilde{b}$, and assume that $OPT(\mathcal{I})$ purchases a card every time we see that $OPT(\mathcal{I}')$ purchases a multiple of $\tilde{b}$ cards. We formalize this notion below.

Definition C.1. Suppose we have an instance $\mathcal{I} = ((c_0, \ldots c_{N-1}), T, B, \beta)$ of the Bahncard problem, and times $0 = i_0 \leq i_1, \ldots, \leq i_k \leq i_{k+1} = N$. If we split $\mathcal{I}$ at $i_1, \ldots, i_k$ we obtain instances $\mathcal{I}_0, \ldots, \mathcal{I}_k$, where instance $\mathcal{I}_j = ((c_{i_j}, \ldots c_{i_{j+1}-1})), T, B, \beta)$.

In Lemma C.2, we bound the increase in cost obtained by splitting an instance.

Lemma C.2. *Suppose we are given an instance $\mathcal{I}$ of the Bahncard problem and we split $\mathcal{I}$ at k timepoints $i_1, \ldots, i_k$ to obtain Bahncard instances $\mathcal{I}_0, \ldots, \mathcal{I}_k$. Then,*

$$\sum_{j=0}^{k} opt(\mathcal{I}_j) \leq opt(\mathcal{I}) + kB$$

Proof. For each $j = 0, \ldots, k$ we construct a solution S_j for $\mathcal{I}_j$ such that total costs paid by $S_0, \ldots, S_k$ together is at most the claimed bound. Suppose that $OPT(\mathcal{I})$ purchases a card at time i such that $i_j \leq i < i_{j+1}$, then S_j will also purchase a card at time i. Suppose now that $OPT(\mathcal{I})$ purchases a card at time i such that $i_j - T + 1 \leq i < i_j$, then S_j will purchase a card at time i_j. Note that these solutions together use at most k more cards than $OPT(\mathcal{I})$, and have valid Bahncards during all times when $OPT(\mathcal{I})$ has one. One can observe that the validity periods of the cards purchased may intersect, but this can be fixed by shifting purchase times while only improving cost. □

To obtain a $(1 + \epsilon)$-approximation algorithm we can select $\tilde{b}, \gamma$ such that

$$\epsilon = \frac{1}{\tilde{b}} + \gamma + \frac{\gamma}{\tilde{b}}$$

and split $\mathcal{I}$ after each point in time that $OPT(\mathcal{I}')$ sees $\tilde{b}$ cards. Formally, we define $i_1, \ldots, i_k$ as follows. Let $i_0 = 0$ and while there are more than least $\tilde{b}$ cards purchased by $OPT(\mathcal{I}')$ after the most recently defined i_j we let i_{j+1} be the point in time where $OPT(\mathcal{I}')$ purchases its $\tilde{b} + 1^{st}$ card after i_j. Let $\mathcal{I}_0, \ldots, \mathcal{I}_k$ be the instances that result from splitting at $i_1, \ldots, i_k$.

Then we solve each instance separately to get a $(1 + \epsilon)$-approximation using $O\left(\left(\frac{\tilde{b}}{\gamma}\right)^{\tilde{b}} \frac{N}{\tilde{b}T} \log(T)\right)$ queries.

C.2 A Note on Simpler Approaches

Simpler algorithms may work when the ticket costs are (say) uniformly distributed. But consider the algorithm that makes at most $N/(\varepsilon T)$ queries. We claim that this does not suffice in general (e.g. Figure 3). Suppose for simplicity that $\beta = 0$, meaning that if we have a valid Bahncard we do not need to pay any additional costs to obtain tickets. Note that the instance $\mathcal{I}_1$ has an optimal solution which purchases a card at time t and pays B. On the other hand, $\mathcal{I}_2$ has an optimal solution which buys no cards and pays $2B$. However querying intervals of length ϵT does not allow us to distinguish between these two instances.

C.3 Making the Bahncard Algorithm Robust to Errors, Proof of Theorem 3.8

This section is devoted to relaxing the assumption that our interval estimator from Section 3 returns the true ticket costs of the requested interval, while still producing a high quality solution. Suppose now that our interval estimator can make errors.

Formally, given an instance $\mathcal{I}$ of the Bahncard problem, if we present the predictor with a length L, where $1 \leq L \leq N$ and day $t \leq N - L$ we obtain a value x that is a prediction of the quantity

$$\sum_{i=t}^{t+L} c_i.$$

However, in this setting x may differ from this quantity by an arbitrary amount. For a subset I of our time horizon, let x_I, denote the total predicted ticket costs over days in I.

We call a predictor satisfying Equation (64)

$$x_{I_1} + x_{I_2} = x_{I_1 \cup I_2} + x_{I_1 \cap I_2}. \tag{64}$$

for any two subsets I_1, I_2 of the time horizon. *internally consistent*, and assume that our predictor has this natural property. With this assumption we can then use the algorithm from Section 3 to obtain a sequence of buying times $\tau_1, \ldots \tau_{k_{pred}}$, for a given instance $\mathcal{I}$ which we denote by $PRED(\mathcal{I})$. The goal is to develop an algorithm that takes $PRED(\mathcal{I})$ as input, and outputs a sequence of buying times $t_1, \ldots, t_k$, which we denote by $ALG(\mathcal{I})$ whose solution cost is close to that of $(1+\gamma)opt(\mathcal{I})$ if our prediction has few errors. Otherwise the solution cost should be close to that produced by the best online algorithm without predictions if the prediction is error-prone.

To quantify how well an algorithm achieves the above goals it is standard to identify a parameter $\lambda \in (0, 1)$ that signifies how much we trust our predictor Purohit et al. (2018); Bamas et al. (2020). As λ approaches 0, we place full trust in our prediction. On the contrary, as λ approaches 1, we do not trust our predictor.

Our algorithm uses the techniques from Purohit et al. (2018) in the ski-rental setting to the repeated horizon setting, to make our algorithm robust. Their paper characterizes the error as the absolute difference between the actual number of ski days, x, and the predicted number of ski days, y. They break down the problem into a few cases depending on the values of x and y. With this breakdown, they give an algorithm that provides a good performance guarantee in terms of robustness and consistency.

We say that a given point t in our lifetime is a *predicted discount time*, or discount time if there exists a buying time $\tau \in PRED(\mathcal{I})$ such that $t \in [\tau, \tau + T)$. Otherwise, we say that t is a *predicted full-price* time or full-price time. Recall that an interval was γ-light if the total ticket predicted ticket purchases during the interval did not exceed $\frac{g\gamma}{b}$. Recall, that under the assumption that the prediction is perfect, b is the upper-bound on the number of Bahncards purchased by $OPT(\mathcal{I})$, that we computed in Equation 12. Under this same assumption $opt(\mathcal{I}) \leq g \leq 2opt(\mathcal{I})$.

Note that $p = \frac{B}{(1-\beta)}$ is the total number of ticket costs that we need to incur during any given period of length T to break even, when we purchase a Bahncard for the same period.

We build $ALG(\mathcal{I})$ as follows. At each point in time t, we maintain a counter r_t that is 0 whenever there is a valid Bahncard at time t. Otherwise, r_t is the minimum of the total tickets purchased since the last Bahncard expired and the number of tickets purchased in the previous T days. For a time t, let $t(r)$, denote the earlest time point that tickets purchased at $t(r)$ contribute to r_t. If $ALG(\mathcal{I})$ purchases a card at time t, we define the interval $[t(r), t)$ be the *counting period* of t. The union of the counting period and the validity period of a purchase of $ALG(\mathcal{I})$ will be referred to as the *relevance period* of the purchase.

We purchase a Bahncard at time t if either (i) t is contained in a discount interval and $r_t \geq p\lambda$, or (ii) t is contained in a full price interval and $r_t \geq p/\lambda$.

Note that the solutions $OPT(\mathcal{I}), ALG(\mathcal{I})$ and $PRED(\mathcal{I})$ are sets of card purchases. However, for simplicity we will often abuse notation and say that A purchases a card at time t, if $t \in A$ and $A \in \{OPT(\mathcal{I}), ALG(\mathcal{I}), PRED(\mathcal{I})\}$. Alternatively, we will say that A has a valid card during $[t, t+T)$.

Proof of Theorem 3.8. First, we partition the timeline into segments, called *chains*. Informally, each chain will correspond to an interval of time for which every point in the interval corresponds to the relevance period of a Bahncard purchase in $ALG(\mathcal{I})$ or the validity period of a Bahncard purchase in $OPT(\mathcal{I})$.

Formally, consider each maximal sub-interval $I = [t_s, t_e)$ of the life-time such that for all internal points $t' \in (t_s, t_e)$ we have that there is either a time $\tau^* \in OPT(\mathcal{I})$ such that $t' \in (\tau^*, \tau^* + T)$ or a time $t \in ALG(\mathcal{I})$ such that $t' \in (t(r), t+T)$.

We make the following observation about our set of chains. If there is any point in time t not contained in a chain I, created in the above process, then both $ALG(\mathcal{I})$ and $OPT(\mathcal{I})$ pay full-price during this time, and make no card purchases. Hence we can restrict our attention to points in time contained in one of these chains.

Fix a chain I, and let ALG_I and OPT_I denote the cost that one would pay on Bahncards and tickets by following buying schedules $ALG(\mathcal{I})$ and $OPT(\mathcal{I})$, respectively over I. We consider three main cases:

Case 1: $I = [\tau^*, \tau^* + T)$**, where** τ^* **is a card purchased by** $OPT(\mathcal{I})$

By construction the relevance period of a card t in $ALG(\mathcal{I})$ does not intersect with I.

Observe that $\tau^* + T$ must be a full-price time point and by construction of our algorithm its costs do not exceed p/λ, and we obtain Equation (65).

$$\frac{ALG_I}{OPT_I} \leq \frac{p/\lambda}{p} \leq \frac{1}{\lambda}. \tag{65}$$

Case 2: I **contains the relevance period of a purchase** $t \in ALG(\mathcal{I})$**, but is disjoint from the validity period of any purchase of** $OPT(\mathcal{I})$**.**

By our construction of chains, I must contain the relevance period of exactly one card purchase $t \in ALG(\mathcal{I})$. It must be that t is a discount time point, for otherwise $OPT(\mathcal{I})$ must intersect with the counting period of t.

Let $I_1 = [t(r), t), I_2 = [t, t+T)$. By our choice of algorithm, OPT_I must accumulate ticket costs of at least λp during I_1, thus in the worst case we can obtain Equation (66).

$$\begin{aligned}\frac{ALG_I}{OPT_I} &\leq \frac{B + \lambda p + \beta c_{I_2}}{\lambda p + c_{I_2}} \\ &\leq 1 + \frac{(1-\beta)}{\lambda}\end{aligned} \tag{66}$$

Case 3: I **contains purchases from both** $ALG(\mathcal{I})$ **and** $OPT(\mathcal{I})$**.**

Observe that if m is the number of card purchases in the chain that $ALG(\mathcal{I})$ makes, and n is the number of card purchases in the chain that $OPT(\mathcal{I})$ makes, then $n \geq m-1$, since the length of the validity period of any card purchased by $OPT(\mathcal{I})$ is at most the relevance period of any card purchased by $ALG(\mathcal{I})$.

If $n \geq m$ we conclude that

$$\begin{aligned}\frac{ALG_I}{OPT_I} &\leq \frac{mB + np/\lambda}{np} \\ &\leq 1 + \frac{1}{\lambda}\end{aligned} \tag{67}$$

Next, we assume that $n = m - 1$. In this case we know that the earliest purchase in the chain is made by $ALG(\mathcal{I})$ at time t, a discount time-point. It follows that during the counting period of t there are λp in ticket costs, and the optimal buying schedule accumulated these ticket costs, as it does not have a valid Bahncard during this time. As a result we obtain Equation (68).

$$\begin{aligned}\frac{ALG_I}{OPT_I} &\leq \frac{mB + \lambda p + (m-1)p/\lambda}{(m-1)p + \lambda p} \\ &\leq 1 + \frac{1/\lambda + 1}{1 + \lambda} \\ &= 1 + \frac{1}{\lambda}\end{aligned} \tag{68}$$

The above bounds hold for any errors made by our predictor. Next, we argue consistency. Assume that the predictions are perfect. In this case, if the number of cards b (Recall b from Equation (12)) is 0, then our algorithm will also never buy and we recover the optimal solution. Otherwise, whenever we have a predicted purchase at time τ, we could have accumulated up to $\frac{g\gamma}{b}$ in ticket costs since the latest point in time between the end of the previous card purchase t by $ALG(\mathcal{I})$ and the beginning of the latest purchase τ^* of $OPT(\mathcal{I})$ most τ. In this case we pay a factor of $\max\{\frac{g\gamma}{b}, \lambda p\}$ for each such purchase τ^*. Since $g \leq 2opt(\mathcal{I})$ when the predictions are perfect, we obtain the result. $\square$