Online Algorithms for Rent-or-Buy with Expert Advice

Sreenivas Gollapudi [*1] Debmalya Panigrahi [*2]

Abstract

We study the use of predictions by multiple experts (such as machine learning algorithms) to improve the performance of online algorithms. In particular, we consider the classical rent-or-buy problem (also called ski rental), and obtain algorithms that provably improve their performance over the adversarial scenario by using these predictions. We also prove matching lower bounds to show that our algorithms are the best possible, and perform experiments to empirically validate their performance in practice.

1. Introduction

Uncertainty plays a central role in many scenarios where an optimizer is faced with the decision between two alternatives with very different costs. The first alternative has a recurring small cost ("rent") while the second alternative presents a large cost upfront ("buy") but nothing thereafter. While long term use justifies the large cost to buy, renting is the preferred option for short term use. The uncertainty arises in the length of use, which is typically not known in advance. These decisions arise in our everyday lives, such as in deciding whether to buy a house or rent. The same question arises in much larger contexts, such as in a corporate decision of whether to invest in a new data center or rent space in an existing one. In optimization, such problems constitute the *rent-or-buy* question, and are modeled as the widely-studied *ski rental* problem (Karlin et al., 1994; Lotker et al., 2008; Khanafer et al., 2013; Kodialam, 2014).

Two popular paradigms for dealing with uncertainty are online algorithms (Borodin & El-Yaniv, 1998) that are designed to work without knowing the input to the problem in advance, and machine learning that makes future predictions by fitting a model to prior data. Recent work has begun to incorporate machine learned predictions into the design of online algorithms (Medina & Vassilvitskii, 2017; Lykouris & Vassilvitskii, 2018; Kumar et al., 2018; Mitzenmacher, 2018) to improve their performance. The goal is to incorporate the ML predictions in a manner that improves the performance of the online algorithm if the predictions are accurate (a design goal called *consistency*), but not degrade it significantly if the predictions are inaccurate (a design goal called *robustness*). Note that these properties are ensured by the algorithm *without any knowledge of the quality of the predictions*.

While the previous studies focused on using prediction inputs from a *single* ML algorithm or expert, we study the more general setting where we get predictions from *multiple* experts. This is often the case in practice, where different ML algorithms use a variety of models and techniques to arrive at different sets of predictions for the future. Indeed, the problem of combining predictions from multiple experts to obtain a policy that matches the performance of the best expert has been extensively studied in the context of online learning (Jacobs et al., 1991; Chen et al., 1999; Hansen, 1999; Masoudnia & Ebrahimpour, 2014). In this paper, we study the use of multiple predictions to improve the performance of online algorithms, namely for the classical ski rental problem. Our goal is to match the performance of the best expert, while also ensuring that the algorithm does not degrade significantly compared to the worst-case performance of the best online algorithm if all the experts have large prediction errors.

The ski rental problem. In the ski rental problem, a skier needs to decide between buying skis at cost b and renting them at the cost of 1 per day. It is easy to see that if the ski season last more than b days, then the skier should buy skis at the start of the season; else, she should rent skis throughout the season. But, the skier does not know the length of the ski season in advance, and only learns it once the season ends. It is well-known that the best deterministic strategy for the skier is to rent for b days and buy after that if the ski season continues longer. This algorithm achieves a competitive ratio of 2.[1] Our goal is to use experts (e.g., ML algorithms) that provide estimates of the length of the season to improve the performance of this online algorithm. The

[*]Equal contribution [1]Google Research [2]Department of Computer Science, Duke University. Correspondence to: Debmalya Panigrahi <debmalya@cs.duke.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1]The competitive ratio of an online algorithm is the worst case ratio of the algorithm's cost to the optimal cost.

ski-rental problem (Karlin et al., 1994; Lotker et al., 2008; Khanafer et al., 2013; Kodialam, 2014) and its many variations such as TCP acknowledgment (Karlin et al., 2003), the parking permit problem (Meyerson, 2005), and snoopy caching (Karlin et al., 1988) model the rent or buy question that is at the heart of decision making in many different settings, and have consequently been extensively studied in the literature.

Consistency and Robustness. Following (Lykouris & Vassilvitskii, 2018; Kumar et al., 2018), we use the notions of consistency and robustness to evaluate our algorithms. We say that an algorithm alg is α-consistent if $\mathsf{alg} \le \alpha \cdot \mathsf{opt}$ provided at least one of the k experts provides the correct prediction. (Note that the algorithm does not know the identity of the correct predictor.) More generally, if the best expert has a prediction error of Δ, i.e., the absolute difference between her prediction and the actual outcome is Δ, then an α-consistent algorithm ensures $\mathsf{alg} \le \alpha \cdot (\mathsf{opt} + \Delta)$. For robustness, we use the standard notion of competitive ratio: i.e., an algorithm is β-robust if for all outcomes, $\mathsf{alg} \le \beta \cdot \mathsf{opt}$. We call α and β the consistency factor and robustness factor respectively. Observe that the classical online algorithm that does not use predictions has consistency and robustness factors of 2. Our goal is to improve the consistency factor without degrading the robustness factor significantly.

Related work. Our study borrows the concepts of robustness and consistency from (Lykouris & Vassilvitskii, 2018) and motivation for a thorough understanding of the ski-rental problem with predictions from (Kumar et al., 2018). The former considered the online caching problem with predictions. It extends the Marking algorithm to incorporate predictions ensuring both robustness and consistency. The latter extends the models to analyze non-clairvoyant scheduling using predictions of the job lengths. As noted above, we differ from these studies in one significant way by considering predictions from multiple experts which makes the problem considerably more challenging. Moreover, we obtain matching upper and lower bounds for our setting, thereby deriving the precise values of the optimal consistency and robustness factors.

Other well-studied models of computation under uncertainty include robust optimization (e.g., (Kouvelis & Yu, 2013)) and stochastic optimization (e.g., (Bubeck & Slivkins, 2012; Mirrokni et al., 2012; Mahdian et al., 2012)). While the former focuses on providing provable guarantees for solutions to particular realizations of uncertain input, the latter generally aims to provide provably good algorithms for stochastic input or input from some known distributions. In contrast, in our study, we make no assumptions on the input.

Our Contributions. Our main contribution in this paper is to develop algorithms that achieve consistency and robustness for the ski rental problem in the presence of predictions provided by multiple experts.

- We first consider the idealized scenario where the best expert makes the correct prediction. For every value of k, we precisely obtain the best deterministic consistency ratio achievable in this setting by giving matching upper and lower bounds. We also show that similar techniques lead to tight results for randomized algorithms as well. (Section 2)
- Next, we extend the above analysis to show that a slightly modified version of this algorithm also achieves the best consistency ratio in the more realistic scenario of non-zero prediction errors. (Section 3)
- We then incorporate robustness into our algorithm. We slightly modify the algorithm such that it continues to have a consistency ratio that almost matches the optimal value, but also guarantees robustness in the form of a worst-case competitive ratio that is only marginally worse than 2, which is the best ratio in the absence of expert advice.
- Finally, we evaluate these algorithms experimentally and show that for natural models of prediction error, our algorithms achieve near-optimal competitive ratios. We also demonstrate the benefits of using multiple experts over a single expert, and empirically prescribe the "right" number of ML predictions to use in this setting.

The Benefit of Using Multiple Experts: Lowering Prediction Error. We close this section by justifying the use of multiple experts in our setting. In particular, we show that even the use of two experts significantly reduces the prediction error compared to a single expert, thereby compensating for the slightly weaker consistency bounds. To illustrate the dependence of the prediction error on the number of experts, let us consider a simple setting where the prediction of an input parameter made by each individual expert has an independent, additive Gaussian noise given by the standard normal variate $\mathcal{N}(0, 1)$. If there is only a single expert, then the prediction error is given by the half-normal distribution

$$f(x) = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) \quad \text{for } 0 \le x < \infty$$

whose mean is $\sqrt{\frac{2}{\pi}}$. Now, consider the setting of two experts, and let X_1 and X_2 be their respective prediction errors, which are independent half normal variates. Since the algorithm competes with the best expert, the overall prediction error of this ensemble of two experts is given by

$\min(X_1, X_2)$ which is distributed as follows:

$$f(x) = 2\sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) \left(\int_{y=x}^{\infty} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{y^2}{2}\right) dy\right) dx.$$

The mean prediction error is then given by

$$2\sqrt{\frac{2}{\pi}} \int_{x=0}^{\infty} x \exp\left(-\frac{x^2}{2}\right) \left(1 - \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right) dx,$$

which evaluates to $\frac{2(\sqrt{2}-1)}{\sqrt{\pi}}$. Therefore, for independent standard Gaussian noise, the mean prediction error decreases by a factor of $\frac{\sqrt{2}}{2(\sqrt{2}-1)} \approx 1.707$ when a single expert is supplemented with a second independent expert. Adding more experts decreases the mean prediction error further. A simple analysis, which we also empirically verify in our experiments later, shows that the prediction error rapidly decreases up to around $3-5$ experts and then decreases slowly thereafter (to the eventual limiting value of 0). This suggests that it might be sufficient to use $3-5$ ML predictors in many practical scenarios for lowering the prediction error to a small value.

2. Ideal Prediction Scenario: An Expert with Zero Prediction Error

In this section, we design algorithms for the ski-rental problem in the idealized scenario where the *best* expert predicts the input correctly. This is an easy case for the single expert scenario treated previously in the literature: since the prediction of the solitary expert must be correct, the problem is equivalent to the corresponding offline problem where the input is known to the algorithm. But, as the number of experts increases, this increases the uncertainty in the input since the algorithm has to choose from a larger set of different predictions, only one of which is correct. Note that the identity of the correct expert is unknown to the algorithm. Hence, the consistency ratio of the algorithm gradually worsens from 1 for a single expert to the eventual limiting value of 2 with an infinite number of experts (which is equivalent to the worst-case online setting as discussed earlier). Our goal is to find the best algorithm if there are k experts providing advice, for any finite k.

Even if there are only two experts, the situation is already somewhat complicated. In this case, the algorithm does not know which of the experts is making the accurate prediction. Let us call their respective predictions a_1 and a_2. There are three possible scenarios:

- Both $a_1 \geq b$ and $a_2 \geq b$: In this case, irrespective of which expert is correct, the algorithm has a unique optimal strategy, that of buying at time 0. Clearly, $\mathsf{alg} = \mathsf{opt}$.
- Both $a_1 < b$ and $a_2 < b$: Again, irrespective of which of expert is correct, the algorithm has a unique optimal strategy of always renting. As in the previous case, we get $\mathsf{alg} = \mathsf{opt}$.
- $a_1 < b$ but $a_2 \geq b$: This is the interesting case, since the two experts are providing predictions that would make the algorithm behave differently. The first expert is advising the algorithm to always rent while the second expert is suggesting that the algorithm should buy at time 0. Our first observation is that neither of these two strategies, by themselves, yields a bounded consistency factor. If the algorithm decides to rent always, and the second expert is correct, then $\mathsf{alg} = a_2$ while $\mathsf{opt} = b$, which has an unbounded ratio. On the other hand, if the algorithm decides to buy at time 0 and the first expert is correct, then $\mathsf{alg} = b$ and $\mathsf{opt} = a_1$, which also has an unbounded ratio.

  To get some intuition for our algorithm, let us consider two extreme cases. For both cases, assume that $a_2 >> b$, say $a_2 > 2b$. First, consider the scenario where $a_1 = b - \epsilon$ (think of $\epsilon > 0$ as a small number compared to b). In this case, if the algorithm decides to rent till a_1, and the second expert turns out to be correct, then $\mathsf{alg} \geq \min(a_1 + b, a_2) = 2b - \epsilon$ irrespective of the algorithm's strategy after time a_1. Since $\mathsf{opt} = b$, the consistency ratio is ≈ 2 and the algorithm fails to take advantage of the learned advice it receives from the two experts. Therefore, in this case, the algorithm should buy before a_1. Since either a_1 or a_2 is corrects, once the algorithm decides to buy before a_1, there is no incentive for it to buy at any time other than 0. Hence, in this case, the algorithm should buy at time 0, which yields a consistency ratio of $\frac{b}{a_1}$. Now, consider the second extreme case of $a_1 = \epsilon$. In this case, if the algorithm decides to buy at time 0, then $\mathsf{alg} = b$. But, $\mathsf{opt} = \epsilon$ if the first expert turns out to the correct, which makes the consistency ratio unbounded. Therefore, in this case, the algorithm should buy at time a_1 rather than at time 0, which has a consistency ratio of $\frac{b+a_1}{b}$.

  Our general strategy is to balance these two ratios arising in the two extreme situations. More precisely, suppose x is the solution to

  $$\frac{b}{x} = \frac{b+x}{b}. \tag{1}$$

  The algorithm follows different strategies based on the value of a_1: if $a_1 \geq x$, then the algorithm buys at time 0, whereas if $a_1 < x$, then the algorithm rents till time a_1 and then buys if the input is larger than a_1. Note that the algorithm ignores the precise value of a_2, as long as $a_2 \geq b$. The following theorem is an easy consequence of the above case analysis; hence, we omit the proof for brevity.

Theorem 1. *For the case of zero prediction error, the above algorithm achieves a consistency ratio of $\phi = \frac{\sqrt{5}+1}{2} = 1.618\ldots$ for two experts.*

Remark: ϕ is also known as the *golden ratio*, and arises in a surprisingly wide range of settings in nature and mathematics!

Interestingly, this consistency ratio is also the best achievable by any deterministic algorithm for two experts. To see this, consider an instance of the problem where $a_1 = (\phi - 1) \cdot b$ and $a_2 = 2b$. In this case, the choices available to the algorithm are to buy at time 0 or at time a_1. (Any other choice is strictly dominated by these two choices: if the algorithm buys at a time between 0 and a_1, that is strictly worse than buying at time 0; and, if the algorithm buys at a time after a_1, that is strictly worse than buying at time a_1.) Now, consider the following adversary strategy: if the algorithm buys at time 0, then, the correct prediction is a_1, and if the algorithm buys at time a_1, then the correct prediction is a_2. A simple calculation now shows that the consistency ratio in either of these cases is ϕ.

Theorem 2. *For the case of zero prediction error, no deterministic algorithm can achieves a consistency ratio that is strictly better than $\phi = \frac{\sqrt{5}+1}{2} = 1.618\ldots$ for two experts.*

Our main result in this section is to generalize the algorithm above to k experts. The algorithm first partitions the segment of the number line $[0, b)$ into k disjoint segments using a set of $k-1$ *breakpoints* $x_1, x_2, \ldots, x_{k-1}$. These breakpoints are derived as the solution to the following system of equations (note that b is constant, so values of x_i are derived as fractions of b):

$$\frac{b}{x_1} = \frac{b + x_1}{x_2} = \frac{b + x_2}{x_3} = \ldots = \frac{b + x_{k-2}}{x_{k-1}} = \frac{b + x_{k-1}}{b}. \tag{2}$$

These equations are obtained as the natural generalization of Eq. (1) to $k > 2$. To obtain a closed form solution for this system of equations, let us denote $y_i := x_i/b$. Then,

$$\frac{1}{y_1} = \frac{1 + y_1}{y_2} = \frac{1 + y_2}{y_3} = \ldots = \frac{1 + y_{k-2}}{y_{k-1}} = 1 + y_{k-1}. \tag{3}$$

Solving these equations yields:

$$y_t = \sum_{i=1}^{t} y_1^i \quad \text{for } 2 \le t \le k-1 \tag{4}$$

where y_1 is given by

$$\sum_{i=1}^{k} y_1^i = 1. \tag{5}$$

```
$x_0 = 0$, $x_k = b$
for $i = 1$ to $k - 1$ do
    $x_i = b \cdot y_i$ (Eqs. (4) and (5))
    if there is no prediction in $S_i = [x_{i-1}, x_i)$ then
        Rent till $x_{i-1}$ and buy at $x_{i-1}$ if the input exceeds $x_{i-1}$; exit
    end if
end for
Rent forever
```

Figure 1. The algorithm for k experts with zero error

The algorithm is now defined in terms of the solutions to these equations, and is given in Fig. 1. The algorithm partitions the range $[0, b)$ into segments $[x_{i-1}, x_i)$ given by the above equations. If every segment has a prediction, then the algorithm rents forever; else, the algorithm rents till the beginning of the first interval that does not contain any prediction and buys at that time if the input is longer.

Before we analyze this algorithm, let us match this description of the algorithm with the one we previously gave for two experts: the intuition for $k = 2$ will be crucial in our analysis of the algorithm for general k. For $k = 2$, the system of equations Eq. (2) reduces to a single equation, namely Eq. (1) that we described earlier. Suppose the solution to this equation is x; then, our algorithm for general k creates two segments $[0, x)$ and $[x, b)$, and uses the following rules:

- If neither expert predicts a value in $[0, x)$, then the algorithm buys at time 0.
- If at least one expert predicts a value in $[0, x)$ but neither predicts one in $[x, b]$, then the algorithm rents till time x and buys at x if the input is larger than x.
- If each segment $[0, x)$ and $[x, b)$ has exactly one expert's advice in it, then the algorithm always rents.

First, we reconcile the superficial dissimilarities of this algorithm with the one we presented earlier for two experts.

- If both predictions are smaller than b, then the previous algorithm always rents while the new algorithm only does so if both segments are occupied. But, if only the first segment $[0, x)$ is occupied, then the fact that one of the experts' predictions is correct implies the input must be smaller than x. Thus, in this case, the algorithm ends up renting always. The only case where the two algorithms differ in their strategy is when both experts predict a value in $[x, b)$. In this case, the previous algorithm always rents but the new algorithm buys at time 0. It turns out that while the former strategy is optimal, the latter achieves a consistency ratio no worse than b/x, which is the overall consistency ratio as well. Hence, either strategy can be used in this case.

- Finally, we consider the case that one of the experts predicts a value smaller than b and the other experts predicts a value greater than b. The algorithmic choice then depends on the value of the first (smaller) prediction. If it is larger than x, then both algorithms buy at time 0. However, there is a difference between the algorithms when the value of the smaller prediction is less than x. In this case, the previous algorithm buys at the first prediction, while our current algorithm buys at time x. Although the strategies differ, they achieve an identical consistency ratio of $1+x$ since the prediction can be arbitrarily close to x.

Thus, we have established that in spite of superficial dissimilarities, the algorithm for k experts essentially follows the same strategy as the algorithm for the special case of two experts. Next, we use this intuition to derive the competitive ratio of the general algorithm.

Theorem 3. *For the case of zero prediction error, the above algorithm achieves a competitive ratio of η_k for k experts, where η_k is the positive real root of the following equation:*

$$\eta_k = \sum_{r=0}^{k-1} \eta_k^{-r}. \tag{6}$$

Remark. The solution to Eq. 6 is referred to as the k-acci constant in mathematics. It is an increasing function of k, starting at 1 for $k = 1$ and converging to 2 in the limit of k going to ∞. It derives its name from the fact that it is the limit of the ratio of two consecutive terms in the k-acci sequence, which is a generalization of the well-known Fibonacci sequence where the last k numbers are added to obtain the next number of the sequence. (So, setting $k = 2$ yields the Fibonacci sequence, and the corresponding Fibonacci constant is the golden ratio ϕ that we encountered in Theorem 1.)

Proof of Theorem 3. We do a case analysis. If all the segments are occupied and the algorithm chooses to rent always, it is clear that $\mathsf{alg} = \mathsf{opt}$ since all the predictions are smaller than b. So, let us consider the scenario where the first unoccupied segment is S_i and the algorithm rents till time x_{i-1} and buys at x_{i-1} if the input is larger. We consider two cases. First, suppose the input is smaller than x_{i-1}. Then, clearly $\mathsf{alg} = \mathsf{opt}$ since the input is smaller than b. Next, suppose the algorithm buys at time x_{i-1}. In this case, the input exceeds x_{i-1}. But, note that since S_i is unoccupied, and one of the predictions is correct, the input cannot terminate in S_i. Hence, it must be the case that the input $\geq x_i$, which is the start of the next segment. It follows that $\mathsf{alg} = b + x_{i-1}$ while $\mathsf{opt} \geq x_i$, which implies that the consistency ratio is at most $\frac{b+x_{i-1}}{x_i}$. Now, we note that Eq. (6) is obtained by rewriting Eq (5) in terms of the ratio $\eta_k = b/x_1 = 1/y_1$. Therefore, η_k given by the solution to Eq. (2) satisfies $\eta_k = \frac{b+x_{i-1}}{x_i}$ for all values of i in Eq. (2). It follows that the consistency ratio of the algorithm is given by η_k in Eq. (6), thereby proving the theorem. ☐

Next, we show that the above consistency ratio is the best achievable by deterministic algorithms for every value of k, by showing matching lower bounds.

Theorem 4. *For the case of zero prediction error and k experts, no deterministic algorithm can achieve a consistency ratio that is strictly better than η_k in Eq. (6).*

Proof. The main idea in the construction of this lower bound is the same as the lower bound sketched for the special case of two experts in Theorem 2. Namely, we create an instance where the choices available to the algorithm precisely realize each of the ratios given in Eq. (2). Suppose there are k experts, and the prediction of the ith expert, for $1 \leq i \leq k-1$, is given by $x_i = b \cdot y_i$, where y_i satisfies Eqs. (4) and (5). The prediction of the kth expert is $2b$. In other words, all the predictions, except that of the last expert, are precisely at the breakpoints that we used to define the segments in the algorithm. First, note that if the algorithm chooses to buy at a time that is strictly inside any of the segments, i.e., neither at a breakpoint nor at time 0, then its solution is strictly dominated by an alternative strategy of buying at the breakpoint at the beginning of the segment (or at time 0 for the first segment). This is because the actual input cannot terminate within a segment since one of the predictions must be correct. So, if the algorithm decides to buy at time x_{i-1}, i.e., at the beginning of the ith segment for any $1 \leq i \leq k$, then the adversary uses the following strategy: For $1 \leq i \leq k$, the ith expert is correct and the actual sequence is of length x_i for $1 \leq i \leq k-1$ and of length $2b$ for $i = k$. Note that this choice of the adversary realizes one of the ratios in Eq. (2) as the consistency ratio, and hence, the consistency ratio of the algorithm cannot be better than η_k given by Eq. (6). This completes the proof of the lower bound. ☐

2.1. Randomized Algorithms

Although this paper primarily focuses on deterministic algorithms, we make a digression here and briefly discuss randomized algorithms for this problem. Consider the case of $k = 2$, where one of the predictions is $> b$ and the other prediction is $x \leq b$. A randomized algorithm is simply a probability distribution over $[0, b]$ that defines when to buy. First, note that this distribution is discrete wlog, concentrated at the values 0 and x. This is because there is no benefit to an algorithm to buy between 0 and x – it might as well buy at 0 – or between x and 1 – it might as well buy at x. Let the probability of buying at 0 be p; then the probability of buying at x is $1-p$. There are two possible outcomes: either the actual sequence ends at x or extends beyond b. The value of opt in these two cases are respectively x and b. Correspondingly, the expected cost of alg in these two cases are respectively $p \cdot b + (1-p)x$ and $p \cdot b + (1-p)(b+x)$. Setting

$$\frac{p \cdot b + (1-p)x}{x} = \frac{p \cdot b + (1-p)(b+x)}{b}$$

and solving for p gives $p = \frac{x^2}{x^2-x+1}$, which is the probability distribution used by the algorithm to decide when to buy. Then, the ratio $\frac{\mathsf{alg}}{\mathsf{opt}} = \frac{b}{x^2-x+1}$ is maximized at $x = b/2$ for the range $x \in [0, b]$. Correspondingly, the maximum value of this ratio is $4/3$. The next theorem follows.

Theorem 5. *For the case of zero prediction error, the above randomized algorithm achieves a competitive ratio of* $4/3$ *for* 2 *experts.*

This theorem can be generalized further to more than 2 experts by using similar techniques as above, and can also be shown to be tight in that no randomized algorithm can perform better. We defer these results to the full version of the paper due to space constraints.

3. Experts with Non-Zero Prediction Errors

In the previous section, we described an algorithm that obtains the optimal consistency ratio for k expert predictions, under the assumption that one of the predictions is correct. In this section, we consider the more realistic scenario where none of the experts is guaranteed to provide an exactly correct prediction. Naturally, the performance of our algorithm will depend on how good the predictions are. Namely, we define the *prediction error*, denoted Δ, to be the smallest absolute difference between the prediction and the actual outcome among all experts.

We slightly modify the partitioning of the interval $[0, b)$ from the previous section by defining a new set of breakpoints $z_0, z_1, \ldots, z_k$ as follows. First, $z_0 = 0$ and $z_k = b$. These breakpoints are derived as the solution to the following system of equations:

$$\frac{b}{z_1} = \frac{b + \frac{z_1+z_2}{2}}{z_2} = \ldots = \frac{b + \frac{z_{k-2}+z_{k-1}}{2}}{z_{k-1}} = \frac{b + \frac{z_{k-1}+b}{2}}{b}. \quad (7)$$

The algorithm is now defined in terms of the solutions to these equations, and is given in Fig. 2. The algorithm partitions the range $[0, b)$ into segments $[z_{i-1}, z_i)$ given by the above equations. If every segment has a prediction, then the algorithm rents forever; else, the algorithm rents till

```
$z_0 = 0$, $z_k = b$
for $i = 1$ to $k - 1$ do
  $z_i$ is given by Eq. (7)
  if there is no prediction in $[z_{i-1}, z_i)$ then
    if $i = 1$ then
      Buy at time 0; exit
    else
      Rent till $\frac{z_{i-1}+z_i}{2}$ and buy at $\frac{z_{i-1}+z_i}{2}$ if the
      input exceeds $\frac{z_{i-1}+z_i}{2}$; exit
    end if
  end if
end for
Rent forever
```

Figure 2. The algorithm for k experts with non-zero error

the *middle* of the first interval that does not contain any prediction and buys at that time if the input is longer. The only exception is that if the first interval does not contain a prediction, then the algorithm buys at time 0.

Note that b is a constant; hence, the values of z_i are obtained from Eq. (7) as fractions of b. Let us denote the ratio obtained as the solution to Eq. (7) by γ_k. The proof of the next theorem is a case analysis similar to Theorem 3, and is hence deferred to the full version of the paper due to space constraints.

Theorem 6. *For prediction error* Δ *and* k *experts, the algorithm given in Fig. 1 satisfies*

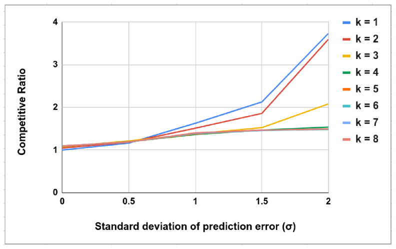
$$\mathsf{alg} \leq \gamma_k \cdot (\mathsf{opt} + \Delta). \quad (8)$$

We close this section by stating that the guarantee provided by Theorem 6 is the best possible in terms of the ratio of $\frac{\mathsf{alg}}{\mathsf{opt}+\Delta}$ for any deterministic algorithm. The proof, which follows the same strategy as Theorem 4 is deferred to the full version due to space constraints.

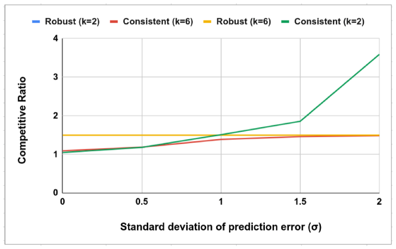
Theorem 7. *For prediction error* Δ *and* k *experts, if a deterministic algorithm provides a guarantee* $\mathsf{alg} \leq \zeta \cdot (\mathsf{opt} + \Delta)$*, then* $\zeta \geq \gamma_k$*.*

4. A Robust and Consistent Algorithm

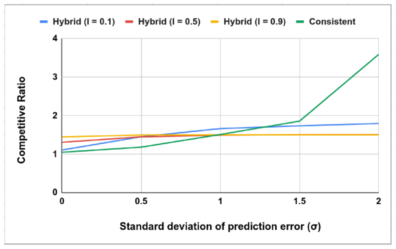
Although the algorithm in Fig. 1 obtains the best achievable guarantee of alg in terms of opt and Δ, the competitive ratio alg/opt can be unbounded for $\Delta > 0$. For instance, if there is a single expert whose prediction is larger than b, the algorithm buys at time 0. However, if the actual input is $\epsilon > 0$, then $\mathsf{alg} = b$ and $\mathsf{opt} = \epsilon$, resulting in a competitive ratio of b/ϵ which is unbounded since ϵ can be arbitrarily small. Contrast this with the best online algorithm in the absence of any expert advice (let us call this the "no ad-



(a) As the number of experts increases, the competitive ratio of the consistent algorithm improves significantly even for large errors in the predictions



(b) The relative performance of the robust and consistent algorithms



(c) The trade-off between robustness and consistency is clearly shown as the prediction error increases

Figure 3. The relative performance of the consistent, robust, and hybrid algorithms. All results are averaged over 10000 trials

vice" algorithm). This algorithm rents till time b and then buys, yielding an $\mathsf{alg}/\mathsf{opt}$ ratio of at most 2 in all scenarios. From a conceptual perspective, this implies that while our algorithm in Fig. 1 takes advantage of good expert predictions and obtains a competitive ratio better than 2 in this case, it suffers a worse competitive ratio than the no advice algorithm if the predictions turns out to be inaccurate. In this section, we desire the best of both worlds: *an algorithm that does much better than the no advice algorithm on being provided good expert predictions, but does not do much worse than the no advice algorithm if the predictions are of poor quality*. Following (Lykouris & Vassilvitskii, 2018; Kumar et al., 2018), we call this property of not degrading significantly compared to the no advice algorithm *robustness* of the algorithm. For brevity, we will call this the *hybrid* algorithm.

In this section, we will denote the consistency factor and robustness factor by α and β respectively. To explore algorithms that are simultaneously consistent and robust, let us first consider the setting of a single expert $k = 1$. (This analysis is the same as (Kumar et al., 2018) since it is for a single expert, but is nevertheless useful for extending to the more general case of multiple experts later.) In this case, there are two possibilities:

- The prediction is $< b$: In this case, our algorithm from Fig. 1 would always rent, which is clearly not robust, e.g., in a situation where the input actually turned out to be unboundedly large. But, this is easy to fix. We can simply use the no advice algorithm instead, which rents till time b and then buys if the input is $\geq b$. In this case, the algorithm continues to have a consistency factor $\alpha = 1$, while the robustness factor $\beta = 2$.
- The prediction is $\geq b$: This is the trickier situation. Here, our algorithm from Fig. 1 would buy at time 0 and achieve $\alpha = 1$, but have an unbounded β. To make the algorithm more robust, let us instead rent till time $\lambda \cdot b$ for some "meta parameter" $\lambda \in [0, 1]$. If the expert is correct, then $\mathsf{alg} = (1 + \lambda)b$ and $\mathsf{opt} = b$, yielding $\alpha = 1 + \lambda$. (Compare this to the best value of α, which is 1.) On the other hand, to estimate the robustness parameter, note that the worst case scenario for the algorithm is that the input ends immediately after the algorithms buys at time $\lambda \cdot b$. In this case, $\mathsf{alg} = (1 + \lambda)b$ and $\mathsf{opt} = \lambda \cdot b$, thereby yielding $\beta = 1 + 1/\lambda$. It is not hard to show that this is indeed the best tradeoff achievable between α and β, but we omit the proof for brevity and move on to the case of multiple experts.

Now, even if the number of experts is > 1, the main modification that ensures robustness is that the algorithm should not buy too early. In particular, the following lemma shows that the robustness parameter only depends on the earliest time that the algorithm buys at.

Lemma 1. *For any algorithm, if the earliest time that it buys is λb across all possible expert predictions, then the robustness factor $\beta = 1 + 1/\lambda$.*

Proof. If we fix the time when an algorithm buys (say bx), then the worst outcome is realized when the input ends immediately after the algorithm buys. In this case $\mathsf{alg} = b + bx$ and $\mathsf{opt} = bx$ yielding a competitive ratio of $1 + 1/x$. Clearly, the robustness factor is then decided by the worst competitive ratio which is achieved when the algorithm buys the earliest, i.e., at time $\lambda \cdot b$. The lemma follows. $\square$

In our algorithm, we will never buy in the interval $[0, \lambda b)$, thereby ensuring a robustness factor of $1 + 1/\lambda$. Given this robustness factor, what is the best consistency factor α that we can achieve? Recall that we would ideally like to have $\alpha \simeq \eta_k$ for k experts, which is the best achievable value by a non-robust algorithm. We use the same ideas as in Section 2. Namely, we create breakpoints x_1 through x_{k-1}

```
$x_0 = \lambda b$, $x_k = b$
**for** $i = 1$ to $k-1$ **do**
  $x_i = b \cdot y_i$ (Eqs. (10) and (11)
  **if** there is no prediction in $[x_{i-1}, x_i)$ **then**
    Rent till $x_{i-1}$ and buy at $x_{i-1}$ if the input exceeds $x_{i-1}$; **exit**
  **end if**
**end for**
Rent forever
```

Figure 4. The hybrid algorithm for k experts

satisfying

$$\frac{b+\lambda b}{x_1} = \frac{b+x_1}{x_2} = \frac{b+x_2}{x_3} = \ldots = \frac{b+x_{k-2}}{x_{k-1}} = \frac{b+x_{k-1}}{b}. \tag{9}$$

Denoting $y_i := x_i/b$, we have

$$y_t = \frac{y_1(1+y_{t-1})}{1+\lambda} \quad \text{for } 2 \le t \le k-1 \tag{10}$$

where y_1 is given by

$$\sum_{i=1}^{k-1} \left(\frac{y_1}{1+\lambda}\right)^i + \frac{y_1^k}{(1+\lambda)^{k-1}} = 1. \tag{11}$$

Using the above equations, we now define the algorithm in Fig. 4, which is similar to the algorithm in Fig. 1 except that it never buys before time λb.

The next theorem, which establishes the consistency parameter of the algorithm, follows the proof strategy in Theorem 3 and hence, the proof is deferred to the full version.

Theorem 8. *The consistency ratio of the algorithm in Fig. 4 is given by $\tilde{\eta}_k$ for k experts, where $\tilde{\eta}_k$ is the positive real root of the following equation:*

$$\tilde{\eta} = \sum_{r=0}^{k-1} \tilde{\eta}^{-r} + \lambda \cdot \tilde{\eta}^{-k}. \tag{12}$$

Next, we show that the above consistency ratio is tight for deterministic algorithms by showing a matching lower bound for every value of k. Again, the lower bound construction follows the same structure as in Theorem 4 and hence, the proof is deferred to the full version.

Theorem 9. *For any deterministic algorithm with a robustness ratio $\beta \le 1 + 1/\lambda$ and k experts, the best consistency ratio achievable is given by $\tilde{\eta}_k$ from Eq. (12).*

5. Experiments

We test the efficacy of our algorithms via simulations. We set the buying cost $b = 1$. (The actual value of b is unimportant because we can scale all values by b.) We choose the actual outcome x to be a value uniformly drawn from $[0, 2b]$. We vary the number of experts from 1 to 8 and set their associated predictions to $x + \epsilon$ where ϵ is drawn from a normal distribution of mean 0 and standard deviation σ. To verify consistency and robustness of our algorithms, we vary σ from 0 to 2. Finally, for the algorithm in Fig. 4, we consider values of 0.1, 0.5, and 0.9 for the meta parameter λ. We label the algorithm defined in Figure 1 *consistent*; it's extension to handle non-zero prediction errors (see Section 3) as *robust*; and the robust and consistent algorithm in Section 4 as *hybrid*. Figure 3 illustrates the relative performance of our algorithms. We make three observations.

First, using more experts is better. The advantage of using more experts is clearly illustrated in Figure 3(a). As the number of experts increases, the consistent algorithm tends to perform significantly better and does not even require explicit robustness adjustments.

Second, the robust algorithm is useful when the prediction error is large. The trade-off between the robust and consistent algorithms is shown in Figure 3(b). As the consistent algorithm tries to mimic the best expert, it's performance is closely tied to the best expert's performance. On the other hand, the robust algorithm does well in this range as it does not depend on the expert's advice at all. This is clearly observed in Figure 3(b). The competitive ratio of the robust algorithm stays constant even as the prediction error increases.

Third, in the absence of any knowledge of the prediction errors, the hybrid algorithm with the robust parameter λ can help achieve good performance by choosing a suitable value of λ. The performance of the hybrid algorithm for two values of $\lambda \in \{0.1, 0.5, 0.9\}$ is shown in Figure 3(c). Even as the consistent algorithm outperforms the hybrid algorithm for all values of λ for small prediction errors, the hybrid algorithm does better as the prediction error increases.

6. Conclusions

In this paper, we initiate the study of improving the worst-case performance of online algorithms by incorporating predictions made by *multiple* experts (typically ML algorithms). In particular, we study the well-known ski rental problem in this framework. We develop algorithms that smoothly trade off between consistency and robustness, and also obtain tight upper and lower bounds on the consistency ratio as a function of the number of experts k. We believe this study, along with the others before it, will raise many interesting questions on incorporating learned advice (e.g., from ML algorithms) into classical optimization problems to improve the performance of online algorithms beyond worst-case bounds.

Acknowledgments

D. Panigrahi was partially supported by NSF grants CCF 1527084 and CCF 1535972, an NSF CAREER Award CCF 1750140, and the Joint Indo-US Networked Center for Algorithms under Uncertainty.

References

Borodin, A. and El-Yaniv, R. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

Bubeck, S. and Slivkins, A. The best of both worlds: Stochastic and adversarial bandits. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012*, pp. 42.1–42.23, 2012.

Chen, K., Xu, L., and Chi, H. Improved learning algorithms for mixture of experts in multiclass classification. *Neural Networks*, 12(9):1229–1252, 1999.

Hansen, J. V. Combining predictors: comparison of five meta machine learning methods. *Information Sciences*, 119(1-2):91–105, 1999.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural Computing*, 3 (1):79–87, 1991.

Karlin, A. R., Manasse, M. S., Rudolph, L., and Sleator, D. D. Competitive snoopy caching. *Algorithmica*, 3: 77–119, 1988.

Karlin, A. R., Manasse, M. S., McGeoch, L. A., and Owicki, S. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.

Karlin, A. R., Kenyon, C., and Randall, D. Dynamic TCP acknowledgment and other stories about e/(e-1). *Algorithmica*, 36(3):209–224, 2003.

Khanafer, A., Kodialam, M., and Puttaswamy, K. P. N. The constrained ski-rental problem and its application to online cloud cost optimization. In *Proceedings of the INFOCOM*, pp. 1492–1500, 2013.

Kodialam, R. Competitive algorithms for an online rent or buy problem with variable demand. *SIAM Undergraduate Research Online*, 7:233–245, 2014.

Kouvelis, P. and Yu, G. *Robust Discrete Optimization and its Applications*. Springer Science & Business Media, 2013.

Kumar, R., Purohit, M., and Svitkina, Z. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 9684–9693, 2018.

Lotker, Z., Patt-Shamir, B., and Rawitz, D. Rent, lease or buy: Randomized algorithms for multislope ski rental. In *Proceedings of the 25th Annual Symposium on the Theoretical Aspects of Computer Science (STACS)*, pp. 503–514, 2008.

Lykouris, T. and Vassilvitskii, S. 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.

Mahdian, M., Nazerzadeh, H., and Saberi, A. Online optimization with uncertain information. *ACM TALG*, 8(1): 2:1–2:29, 2012.

Masoudnia, S. and Ebrahimpour, R. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2): 275–293, 2014.

Medina, A. M. and Vassilvitskii, S. Revenue optimization with approximate bid predictions. In *Proceedings of the NeurIPS*, pp. 1856–1864, 2017.

Meyerson, A. The parking permit problem. In *Proc. of 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 274–284, 2005.

Mirrokni, V. S., Gharan, S. O., and Zadimoghaddam, M. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 1690–1701, 2012.

Mitzenmacher, M. A model for learned bloom filters and optimizing by sandwiching. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 462–471, 2018.