The Secretary Problem with Predicted Additive Gap

Alexander Braun * **Sherry Sarkar** †

Abstract

The secretary problem is one of the fundamental problems in online decision making; a tight competitive ratio for this problem of $1/\mathrm{e} \approx 0.368$ has been known since the 1960s. Much more recently, the study of algorithms with predictions was introduced: The algorithm is equipped with a (possibly erroneous) additional piece of information upfront which can be used to improve the algorithm's performance. Complementing previous work on secretary problems with prior knowledge, we tackle the following question:

What is the weakest piece of information that allows us to break the $1/\mathrm{e}$ barrier?

To this end, we introduce the secretary problem with predicted additive gap. As in the classical problem, weights are fixed by an adversary and elements appear in random order. In contrast to previous variants of predictions, our algorithm only has access to a much weaker piece of information: an *additive gap* c. This gap is the difference between the highest and k-th highest weight in the sequence. Unlike previous pieces of advice, knowing an exact additive gap does not make the problem trivial. Our contribution is twofold. First, we show that for any index k and any gap c, we can obtain a competitive ratio of 0.4 when knowing the exact gap (even if we do not know k), hence beating the prevalent bound for the classical problem by a constant. Second, a slightly modified version of our algorithm allows to prove standard robustness-consistency properties as well as improved guarantees when knowing a range for the error of the prediction.

The full version with proofs can be found at `https://arxiv.org/abs/2409.20460` [Braun and Sarkar, 2024].

1 Introduction

The secretary problem is a fundamental problem in online decision making: An adversary fixes non-negative, real-valued weights $w_1 \geq w_2 \geq \cdots \geq w_n$ which are revealed online in random order. The decision maker is allowed to accept (at most) one element. At the time of arrival of an element, the decision maker is required to make an immediate and irrevocable acceptance decision. The goal is to maximize the weight of the selected element. A tight guarantee[3] of $1/\mathrm{e}$ is known since the seminal work of Lindley [1961] and Dynkin [1963] (also see Ferguson [1989] or Freeman [1983]) and can be achieved with a very simple threshold policy.

In the modern era, the assumption of having no prior information on the weights is highly pessimistic. To go beyond a worst case analysis, researchers have recently considered the setting where we have some sort of learned prediction that our algorithm may use up front. This setting spawned the recent and very successful field of algorithms with predictions. Antoniadis et al. [2020] and Dütting et al. [2021] studied the secretary problem with a prediction of the largest weight in the sequence, and

*Institute of Computer Science, University of Bonn. Email: alexander.braun@uni-bonn.de

†Carnegie Mellon University. Email: sherrys@andrew.cmu.edu

[3]In the literature, there are two variants of the secretary problem: *probability-maximization* (maximize the probability of selecting w_1) and *value-maximization* (maximize selected weight). Throughout the paper, we consider the latter of the two variants.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

resolve this setting with an algorithm which yields a nice robustness-consistency trade-off. Fujii and Yoshida [2023] consider the secretary problem with an even stronger prediction: A prediction for every weight in the sequence.

However, predicting the largest element or weight can sometimes be difficult or unfavorable. For example in retail, past data may only contain information about prices and not the true values of buyers [Kleinberg and Leighton, 2003, Leme et al., 2023]; or for data privacy reasons (see e.g. Asi et al. [2023]), only surrogates for largest weights are revealed in history data. This motivates to advance our understanding of the following question:

What is the weakest piece of information we can predict that still allows us to break the $1/e$ barrier?

Stated another way, is there a different parameter we can predict, one that does not require us to learn the best value, but is still strong enough to beat $1/e$? This brings us to the idea of predicting the *gap* between the highest and k-th highest weight, or in other words, predicting how valuable w_k is with respect to w_1. Coming back to data privacy for example, such a parameter does successfully anonymize the largest weight in the sequence.

From a theoretical perspective, for some special cases of gaps, previous work directly implies improved algorithms. For example, if we know w_1 and w_2 have the same weight, using an algorithm for the two-best secretary problem [Gilbert and Mosteller, 1966] directly leads to a better guarantee. More generally, if we know the *multiplicative gap* w_1/w_2, we observe that we can generalize the optimal algorithm for $w_1 = w_2$ (see e.g. Gilbert and Mosteller [1966], Buchbinder et al. [2014]). However, if we instead only know $w_n/w_1 = 0$, this does not help at all. The only insight is that $w_n = 0$; we have no insight on the range of values taken. This essentially boils down to the classical secretary problem and the best competitive ratio again is $1/\mathrm{e}$. So while the multiplicative gap advises about the relative values without needing to know w_1 entirely, it is not strong enough in general to beat $1/\mathrm{e}$. In this paper, we consider instead predicting an *additive gap* $w_1 - w_k$.

The additive gap between w_1 and w_k can be viewed as interpolating between two previously studied setups: when $w_1 - w_k$ gets small, we get closer towards the k-best secretary problem, and when $w_1 - w_k$ is very large, the additive gap acts as a surrogate prediction of w_1, the prediction setting in Antoniadis et al. [2020] and Dütting et al. [2021]. As we will see, even though the additive gap is much weaker than a direct prediction for w_1, it strikes the perfect middle ground: it is strong enough to beat $1/e$ by a constant for any possible value of the gap $w_1 - w_k$ (and even if we do not know what k is upfront). In addition, in contrast to pieces of advice studied in the literature so far, knowing an exact additive gap does not make the problem trivial to solve.

1.1 Our Results and Techniques

Our contribution is threefold. First, in Section 3, we show the aforementioned result: knowing an exact additive gap allows us to beat the competitive ratio of $1/\mathrm{e}$ by a constant.

Theorem 1 (Theorem 4, simplified form)**.** *There exists a deterministic online algorithm which achieves an expected weight of* $\mathbf{E}[\mathrm{ALG}] \geq 0.4 \cdot w_1$ *given access to a single additive gap* c_k *for* $c_k = w_1 - w_k$ *and some* k*.*

Still, getting an exact gap might be too much to expect. Hence, in Section 4, we introduce a slight modification in the algorithm to make it robust with respect to errors in the predicted gap while simultaneously outperforming the prevalent competitive ratio of $1/\mathrm{e}$ by a constant for accurate gaps.

Theorem 2 (Theorem 5, simplified form)**.** *There exists a deterministic online algorithm which uses a predicted additive gap and is simultaneously* $(1/\mathrm{e} + O(1))$*-consistent and* $O(1)$*-robust.*

The previous Theorem 2 does not assume any bounds on the error of the predicted additive gap used by our algorithm. In particular, the error of the prediction might be unbounded and our algorithm is still constant competitive. However, if we know that the error is bounded, we can do much better.

Theorem 3 (Theorem 6, simplified form)**.** *There exists a deterministic online algorithm which achieves an expected weight of* $\mathbf{E}[\mathrm{ALG}] \geq 0.4 \cdot w_1 - 2\epsilon$ *given access to a bound* ϵ *on the error of the predicted gap.*

Our algorithms are inspired by the one for classical secretary, but additionally incorporate the gap: Wait for some time to get a flavor for the weights in the sequence, set a threshold based on the past observations and the gap, pick the first element exceeding the threshold.

At first glance, this might not sound promising: In cases when the gap is small, incorporating the gap in the threshold does not really affect the best-so-far term. Hence, it may seem that beating $1/\mathrm{e}$ is still hard. However, in these cases, even though the threshold will be dominated by the best-so-far term most of the time, the gap reveals the information that the best value and all other values up to w_k are not too far off. That is, even accepting a weight which is at least w_k ensures a sufficient contribution.

Our analyses use this fact in a foundational way: Either the gap is large in which case we do not consider many elements in the sequence for acceptance at all. Or the gap is small which implies that accepting any of the k highest elements is reasonably good. For each of the cases we derive lower bounds on the weight achieved by the algorithm.

Since we do not know upfront which case the instance belongs to, we optimize our initial waiting time for the worse of the two cases. In other words, the waiting time cannot be tailored to the respective case but rather needs to be able to deal with both cases simultaneously. This introduces some sort of tension: For instances which have a large gap, we would like the waiting time to be small. By this, we could minimize the loss which we incur by waiting instead of accepting high weighted elements at the beginning of the sequence. For instances which have a small gap, the contribution of the gap to the algorithm's threshold can be negligible. This results in the need of a longer waiting time at the beginning to learn the magnitude of weights reasonably well. We solve this issue by using a waiting time which balances between these two extremes: It is (for most cases) shorter than the waiting time of $1/\mathrm{e}$ from the classical secretary algorithm. Still, it is large enough to gain some information on the instance with reasonable probability.

As a corollary of our main theorem, we show that we can beat the competitive ratio of $1/\mathrm{e}$ even if we only know the gap $w_1 - w_k$ but do not get to know the index k. In particular, this proves that even an information like "There is a gap of c in the instance" is helpful to beat $1/\mathrm{e}$, no matter which weights are in the sequence and which value c attains.

Complementing theoretical results, we run simulations in Section 6 which strengthen our theoretical findings. First, we show that for instances in which the classical secretary algorithm achieves a nearly tight guarantee of $1/\mathrm{e}$, our algorithm can almost always select the highest weight. In addition, we further investigate the robustness-consistency trade-off of our algorithm. In particular, as it will turn out, underestimating is not as much of an issue as overestimating the exact gap.

1.2 Additional Related Work

Implications of related results on the additive gap. In the two-best secretary problem, we can pick at most one element but win when selecting either the best or second best element. For this problem, the competitive ratio is upper bounded by approximately 0.5736 [Buchbinder et al., 2014, Chan et al., 2015] (the authors provide an algorithm which matches this bound, so the guarantee is tight). As our setting with $w_1 - w_2 = 0$ can be viewed as a special case, this yields a hardness result; the best any algorithm can perform with the exact additive gap provided upfront is approximately 0.5736.

A non-exhaustive list of related work on secretary problems. Since the introduction of the secretary problem in the 1960s, there have been a lot of extensions and generalizations of this problem with beautiful algorithmic ideas to solve them Kleinberg [2005], Babaioff et al. [2007, 2018], Feldman et al. [2018], Korula and Pál [2009], Mahdian and Yan [2011], Kesselheim et al. [2018], Rubinstein [2016]. Beyond classical setups, recent work by Kaplan et al. [2020] and Correa et al. [2021] studies the secretary problem with sampled information upfront. Here, some elements are revealed upfront to the algorithm which then tries to pick the best of the remaining weights. Guarantees are achieved with respect to the best remaining element in the sequence. In addition, there are also papers bridging between the secretary problem and the prophet inequality world, e.g. Correa et al. [2020] or Correa et al. [2019] and many more [Bradac et al., 2020, Kesselheim and Molinaro, 2020, Argue et al., 2022].

Algorithms with machine learned advice. In the introduction, we already scratched the surface of the field on algorithms with predictions. Here, the algorithm has access to some machine learned advice upfront and may use this information to adapt decisions. Initiated by the work of Lykouris and Vassilvitskii [2021] and Purohit et al. [2018], there have been many new and interesting results in completely classical problems within the last years, including ski rental Wei and Zhang [2020], online bipartite matching Lavastida et al. [2021], load balancing Ahmadian et al. [2023], and many more (see e.g. Im et al. [2021], Zeynali et al. [2021], Almanza et al. [2021]). Since this area is developing

very fast, we refer the reader to the excellent website Algorithms-with-Predictions for references of literature.

As mentioned before, also the secretary problem itself has been studied in this framework. Antoniadis et al. [2020] consider the secretary problem when the machine learned advice predicts the weight of the largest element w_1. Their algorithm's performance depends on the error of the prediction as well as some confidence parameter by how much the decision maker trusts the advice. In complementary work, Dütting et al. [2021] give a bigger picture for secretary problems with machine learned advice. Their approach is LP based and can capture a variety of settings. They assume that the prediction is one variable for each weight (e.g. a 0/1-variable indicating if the current element is the best overall or not). Fujii and Yoshida [2023] assume an even stronger prediction: Their algorithm is given a prediction for every weight in the sequence. In contrast, we go into the opposite direction and deal with a less informative piece of information.

Our work also fits into the body of literature studying weak prediction models, previously studied for e.g. paging [Antoniadis et al., 2023], online search [Angelopoulos, 2021], to just mention a few. In these, several different directions for weak prediction models were considered. For example, the setting in scheduling or caching where the number of predictions is smaller than the input size [Im et al., 2022, Benomar and Perchet, 2024].

2 Preliminaries

In the secretary problem, an adversary fixes n non-negative, real-valued weights, denoted $w_1 \geq w_2 \geq \cdots \geq w_n$. For each of the elements, there is an *arrival time*[4] $t_i \overset{\text{iid}}{\sim} \text{Unif}[0,1]$. Weight w_i is revealed at time t_i and we immediately and irrevocably need to decide if we want to accept or reject this element. Overall, we are allowed to accept at most one element with the objective of maximizing the selected weight. We say that an algorithm is α-*competitive* or achieves a *competitive ratio* of α if $\mathbf{E}[\text{ALG}] \geq \alpha \cdot w_1 = \alpha \cdot \max_i w_i$, where the expectation is taken over the random arrival times of elements (and possible internal randomness of the algorithm).
In addition to the random arrival order, we assume to have access to a single prediction $\hat{c}_k$ for one *additive gap* together with its index k. The additive gap for some index $2 \leq k \leq n$ is $c_k \coloneqq w_1 - w_k$. We say that an algorithm has access to an *exact* or *accurate* gap if $\hat{c}_k = c_k$ (as in Section 3). When the algorithm gets a predicted additive gap $\hat{c}_k$ which might not be accurate (as in Section 4 or Section 5), we say that $\hat{c}_k$ has *error* $\eta = |\hat{c}_k - c_k|$. We call an algorithm ρ-*robust* if the algorithm is ρ-competitive regardless of error η and we say the algorithm is ψ-*consistent* if the algorithm is ψ-competitive when $\eta = 0$. To fix notation, for any time $\tau \in [0,1]$, we denote by $\text{BSF}(\tau)$ (read *best-so-far* at time τ) the highest weight which did appear up to time τ. In other words, $\text{BSF}(\tau) = \max_{i:t_i \leq \tau} w_i$. Also, when clear from the context, we drop the index k at the gap and only call it c or $\hat{c}$ respectively.

3 Knowing an Exact Gap

Before diving into the cases where the predicted gap may be inaccurate in Section 4 and Section 5, we start with the setup of getting a precise prediction for the gap. That is, we are given the exact gap $c_k = w_1 - w_k$ for some $2 \leq k \leq n$. We assume that we get to know the index k as well as the value of c_k, but neither w_1 nor w_k.

Our algorithm takes as input the gap c as well as a waiting time τ. This gives us the freedom to potentially choose τ independent of k if required. As a consequence, we could make the algorithm oblivious to the index k of the element to which the gap is revealed. We will use this in Corollary 1.

Algorithm 1 Secretary with Exact Additive Gap

Input: Additive gap c, time $\tau \in [0,1]$	
Before time τ:	Observe weights w_i
At time τ:	Compute $\text{BSF}(\tau) = \max_{i:t_i \leq \tau} w_i$
After time τ:	Accept first element with $w_i \geq \max(\text{BSF}(\tau), c)$

[4]Note that this setting is equivalent to drawing a random permutation of the n elements and revealing elements in this order one by one.

This algorithm beats the prevalent competitive ratio of $1/e \approx 0.368$ by a constant.

Theorem 4. *Given any additive gap $c_k = w_1 - w_k$, for $\tau = 1 - (1/k+1)^{1/k}$, Algorithm 1 achieves a competitive ratio of* $\max\left(0.4, 1/2\,(1/k+1)^{1/k}\right)$.

Note that as k tends towards n and both become large, the competitive ratio approaches $1/2$.

We split the proof of Theorem 4 in the following two lemmas. Each of them gives a suitable bound on the performance of our algorithm for general waiting times τ in settings when w_k is small or large.

The first lemma gives a lower bound in cases when w_k is small in comparison to w_1.

Lemma 1. *If $w_k < \frac{1}{2}w_1$, then* $\mathbf{E}[\mathrm{ALG}] \geq (1-\tau)\left(\frac{1}{2} + \frac{1}{2(k-1)}\right) \cdot w_1$.

The second lemma will be used to give a bound when w_k is large compared to w_1.

Lemma 2. *If $w_k \geq \frac{1}{2}w_1$, then the following two bounds hold:*

(i) $\mathbf{E}[\mathrm{ALG}] \geq \frac{k+1}{2k}\left(1 - \tau - (1-\tau)^{k+1}\right) \cdot w_1$ *and*

(ii) $\mathbf{E}[\mathrm{ALG}] \geq \left(\frac{3}{2}\tau \ln\left(\frac{1}{\tau}\right) - \frac{1}{2}\tau(1-\tau)\right) \cdot w_1$.

As a consequence, $\mathbf{E}[\mathrm{ALG}]$ *is also at least as large as the maximum of the two bounds.*

The proofs of the lemmas as well as their combination to prove Theorem 4 can be found in the full version [Braun and Sarkar, 2024]. From a high level perspective, the two lemmas give a reasonable bound depending of we either exclude a lot of elements in the algorithm (Lemma 1) or if the largest k elements ensure a sufficient contribution (Lemma 2).

As a corollary of the proof of Theorem 4, we also get a lower bound on the weight achieved by the algorithm if we are only given the gap, but not the element which obtains this gap. That is, we are given c_k but not the index k.

Corollary 1. *If the algorithm only knows c_k, but not k, setting $\tau = 0.2$ achieves* $\mathbf{E}[\mathrm{ALG}] \geq 0.4 \cdot w_1$.

The full version [Braun and Sarkar, 2024] contains a proof of Corollary 1 as well. It mainly relies on the fact that some lower bound we obtained in the proof of Theorem 4 holds for any choice $\tau \in [0,1]$. Also, the algorithm itself only uses the gap to contribute to the threshold. The index k is only used to compute τ. As a consequence, when choosing $\tau = 0.2$ independent of k, the algorithm is oblivious to the exact value of k, but only depends on the gap c_k. For this choice of τ, we can show that $\alpha \geq 0.4$.

As a consequence, very surprisingly, even if we only get to know *some* additive gap c_k and not even the index k, we can outperform the prevalent bound of $1/e$. Also, observe that this is independent of the exact value that c_k attains and holds for any small or large gaps.

As mentioned before, Algorithm 1 is required to get the exact gap as input. In particular, once the gap we use in the algorithm is a tiny bit larger than the actual gap c_k, we might end up selecting no element at all.

Example 1. We get to know the gap to the smallest weight $c_n = w_1 - w_n$ and the smallest weight w_n in the sequence satisfies $w_n = 0$. Let the gap which we use in Algorithm 1 be only some tiny $\delta > 0$ too large. In other words, we use $c_n + \delta$ as a gap in the algorithm. Still, this implies that our threshold $\max(\mathrm{BSF}(\tau), c_n + \delta)$ after the waiting time satisfies

$$\max(\mathrm{BSF}(\tau), c_n + \delta) \geq c_n + \delta = w_1 + \delta > w_1 \geq w_2 \geq \cdots \geq w_n \ .$$

As a consequence, we end up selecting no weight at all and have $\mathbf{E}[\mathrm{ALG}] = 0$.

This naturally motivates the need to introduce more robust deterministic algorithms in this setting. In Section 4, we will show that a slight modification in the algorithm and its analysis allows to obtain robustness to errors in the predictions while simultaneously outperforming $1/e$ for accurate gaps.

4 Robustness-Consistency Trade-offs

Next, we show how to slightly modify our algorithm in order to still beat $1/e$ when getting the correct gap as input, but still be constant competitive in case the predicted gap is inaccurate. The modification

leads to Algorithm 2: Initially, we run the same algorithm as before. After some time $1-\gamma$, we will lower our threshold in order to hedge against an incorrect prediction.

Algorithm 2 Robust-Consistent Algorithm

Input: Predicted gap $\hat{c}$, times $\tau \in [0,1)$, $\gamma \in [0, 1-\tau)$	
Before time τ:	Observe weights w_i
At time τ:	Compute $\mathrm{BSF}(\tau) = \max_{i:t_i \leq \tau} w_i$
Between time τ and time $1-\gamma$:	Accept first element with $w_i \geq \max(\mathrm{BSF}(\tau), \hat{c})$
After time $1-\gamma$:	Accept first element with $w_i \geq \mathrm{BSF}(\tau)$

Note that by $\gamma \in [0, 1-\tau)$, we ensure that $\tau < 1-\gamma$, i.e. the waiting time τ is not after time $1-\gamma$ and hence, the algorithm is well-defined. Now, we can state the following theorem which gives guarantees on the consistency and the robustness of Algorithm 2. We will discuss afterwards how to choose τ and γ in order to outperform the classical bound of $1/\mathrm{e}$ by a constant for accurate predictions while ensuring to be constant-robust at the same time.

Theorem 5. *Given a prediction $\hat{c}_k$ for the additive gap c_k, define*

- $\alpha_1 \coloneqq 1-\gamma-\tau+\tau\ln\left(\frac{1}{1-\gamma}\right)$ *and*
  $\alpha_2 \coloneqq \frac{1}{2}\left((1+\gamma)(1-\tau-\gamma)+\tau\ln\left(\frac{1}{\tau}\right)+\tau\ln\left(\frac{1}{1-\gamma}\right)\right)$,
- $\alpha_3 \coloneqq \frac{k+1}{2k}\left(1-\tau-(1-\tau)^{k+1}\right)$ *and* $\alpha_4 \coloneqq \frac{3}{2}\tau\ln\left(\frac{1}{\tau}\right)-\frac{1}{2}\tau(1-\tau)$.

Then, Algorithm 2 is (i) $\min\left(\min\left(\alpha_1,\alpha_2\right),\max\left(\alpha_3,\alpha_4\right)\right)$*-consistent and (ii)* $\left(\tau\cdot\ln\left(\frac{1}{1-\gamma}\right)\right)$*-robust.*

Observe that if we do not trust the prediction at all, we could set $\tau = 1/\mathrm{e}$ and $1-\gamma = 1/\mathrm{e}$. Doing so, we do not use the prediction in our algorithm. Still, for these choices, we recover the guarantee from classical secretary of $1/\mathrm{e}$. In other words, we can interpret γ as a trust parameter for the prediction which also mirrors our risk appetite. If we do not trust the prediction at all or if we are highly risk averse, we can set $1-\gamma \approx \tau$. If we are willing to suffer a lot in case of an inaccurate prediction (or if we have high trust in the prediction), we will set $\gamma \approx 0$.

Theorem 5 yields a trade-off between robustness and consistency. In particular, for a fixed level of robustness, we can choose the optimal values for τ and $1-\gamma$ for the bounds in Theorem 5 to obtain the plot in Figure 1. Observe that when not focusing on robustness (i.e. choosing robustness being equal to zero), we can achieve a consistency approximately matching the upper bound of 0.5736 described in Section 1.2.

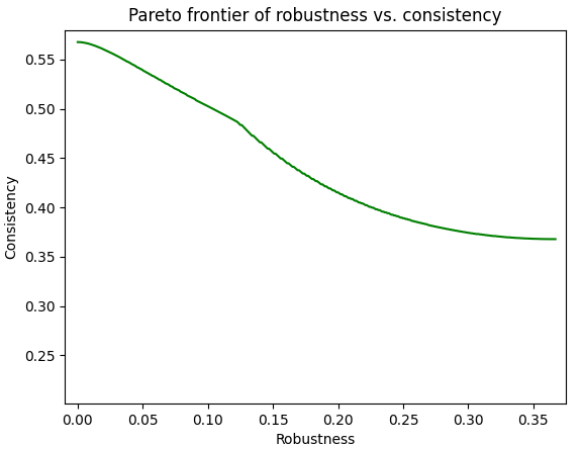

Figure 1: Choosing the optimal parameters τ and $1-\gamma$ for our analysis in Theorem 5: For a given level of robustness, what is the best consistency we can obtain with our analysis.

The proof of Theorem 5 can be found in the full version [Braun and Sarkar, 2024]. Concerning robustness, we can only obtain a reasonable contribution by accepting the best weight. Therefore, we derive a lower bound on the probability of accepting the highest weight via Algorithm 2. Concerning consistency, we can perform a case distinction whether w_k is small or large. Crucially, one is required to take the drop in the threshold after time $1 - \gamma$ into account.

For example, when using a waiting time $\tau = 0.2$ as in Corollary 1 independent of the index k and a value of $\gamma = 0.6$, i.e. $1 - \gamma = 0.4$, we get the following: Algorithm 2 is approximately 0.383-consistent and 0.183-robust (also see Figure 2).

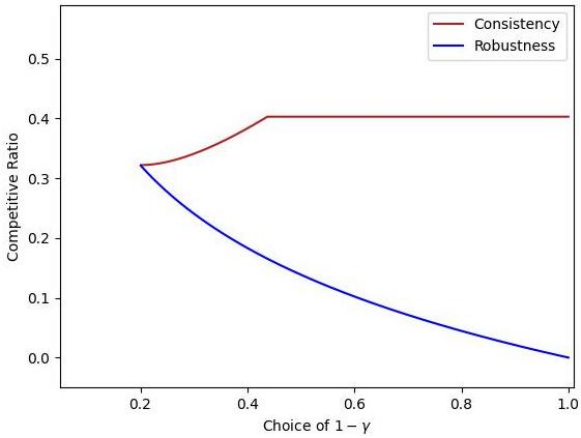

Figure 2: Trade-off between robustness and consistency as a function of the time $1 - \gamma$ for fixed choice of $\tau = 0.2$.

In particular, we can outperform the prevalent bound of $1/e$ by a constant if the predicted gap is accurate while ensuring to be constant competitive even if our predicted gap is horribly off. Of course, when being more risk averse, one could also increase the robustness guarantee for the cost of decreasing the competitive ratio for consistent predictions.[5]

We highlight that these guarantees as well as Theorem 5 hold independent of any bounds on the error of the predicted gap. However, it is reasonable to assume that we have some bounds on how inaccurate our predicted gap is (for example, if our predicted gap is learned from independent random samples). We show in Section 5 that we can achieve much better competitive ratios when we know a range for the error.

5 Improved Guarantees for Bounded Errors

Complementing the previous sections where we had either access to the exact gap (Section 3) or no information on a possible error in the prediction (Section 4), we now assume that the error is bounded[6]. That is, we get to know some $\widetilde{c}_k \in [c_k - \epsilon; c_k + \epsilon]$ which is ensured to be at most an ϵ off. Also, the bound ϵ on the error is revealed to us. Still, the true gap c_k remains unknown.

Our algorithm follows the template which we discussed before. Still, we slightly perturb $\widetilde{c}_k$ to ensure that the threshold is not exceeding w_1. This algorithm allows to state an approximate version of Theorem 4 for the same lower bounds of α as in the exact gap case.

[5]We can also slightly improve the trade-off by allowing randomization: Flip a (biased) coin and either choose our algorithm from Section 3 or the classical secretary algorithm which achieves a guarantee of $1/e$. Our deterministic approach is approximately as good as this randomized variant if we use an unbiased coin and even better compared to biasing the coin towards the algorithm using the prediction. When using a bias towards the classical algorithm, the randomized algorithm has a better trade-off than our deterministic approach.

[6]In order to distinguish a bounded error from a possibly unbounded one, we use $\widetilde{c}$ instead of $\hat{c}$ for the predicted gap in this section.

Algorithm 3 Secretary with Bounded Prediction Error

Input: Approximate gap $\widetilde{c}$, time $\tau \in [0,1]$, error bound ϵ
Before time τ: Observe weights w_i
At time τ: Compute $\mathrm{BSF}(\tau) = \max_{i:t_i \leq \tau} w_i$
After time τ: Accept first element with $w_i \geq \max(\mathrm{BSF}(\tau), \widetilde{c} - \epsilon)$

Theorem 6. *Given any prediction of the gap $\widetilde{c}_k \in [c_k - \epsilon; c_k + \epsilon]$, where $c_k = w_1 - w_k$, Algorithm 3 satisfies* $\mathbf{E}[\mathrm{ALG}] \geq \alpha \cdot w_1 - 2\epsilon$. *For $\tau = 1 - \left(\frac{1}{k+1}\right)^{1/k}$, $\alpha \geq \max\left(0.4, \frac{1}{2}\left(\frac{1}{k+1}\right)^{1/k}\right)$ and for $\tau = 0.2$, $\alpha \geq 0.4$.*

As a consequence, the guarantees from the exact gap case in Section 3 carry over with an additional loss of 2ϵ. Also, the results when not knowing the index k carry over. In particular, this nicely complements the robustness result from Theorem 5 as follows: Once we can bound the error in a reasonable range, even not knowing the gap exactly does not cause too much of an issue. The proof of Theorem 6 can be found in the full version [Braun and Sarkar, 2024].

6 Simulations

In order to gain a more fine-grained understanding of the underlying habits, we run experiments[7] with simulated weights and compare our algorithms among each other and to the classical secretary algorithm[8].

In Section 6.1, we compare our Algorithm 1 to the classical secretary algorithm. To this end, we draw weights i.i.d. from distributions and execute our algorithm and the classical one. As it will turn out, instances which are hard in the normal secretary setting (i.e. when not knowing any additive gap) become significantly easier with additive gap; we can select the best candidate with a much higher probability. We also demonstrate that for some instances, knowing the gap has a smaller impact, though our Algorithm 1 still outperforms the classical one.

Second, in Section 6.2, we turn towards inaccurate gaps and compare Algorithm 1 developed in Section 3 to the robust and consistent variant of Algorithm 2 from Section 4. As a matter of fact, we will see that underestimating the exact gap is not as much of an issue as an overestimation. In particular, underestimating the gap implies a smooth decay in the competitive ratio while overestimating can immediately lead to a huge drop.

6.1 The Impact of Knowing the Gap

We compare our algorithm with additive gap to the classical secretary algorithm (see e.g. [Dynkin, 1963]) with a waiting time of $1/\mathrm{e}$.

6.1.1 Experimental Setup

We run the comparison on three different classes of instances:

(i) *Pareto*: We first draw some $\theta \sim \mathrm{Pareto}(5/n, 1)$. Afterwards, each weight w_i is determined as follows: Draw $Y_i \sim \mathrm{Unif}[0, \theta]$ i.i.d. and set $w_i = Y_i^{\left(n^{1.5}\right)}$ (for more details on Pareto distributions and secretary problems, see e.g. Ferguson [1989]).

(ii) *Exponential*: Here, all $w_i \sim \mathrm{Exp}(1)$.

(iii) *Chi-Squared*: Draw $w_i \sim \chi^2(10)$. That is, each w_i is drawn from a chi-squared distribution which sums over ten squared i.i.d. standard normal random variables.

[7] All experiments were implemented in Python 3.9 and executed on a machine with Apple M1 and 8 GB Memory.

[8] As the piece of information we use as a prediction is fairly different to the pieces which were used in the literature before, we will not compare our algorithms to other algorithms from the secretary problem with predictions literature.

For each class of instances, we average over 5000 iterations. In each iteration, we draw $n = 200$ weights i.i.d. from the respective distribution together with 200 arrival times which are drawn i.i.d. from Unif$[0, 1]$. The benchmark is the classical secretary algorithm with a waiting time of $\tau = 1/e$: Set the largest weight up to time τ as a threshold and accepts the first element afterwards exceeding this threshold. Algorithm 1 is executed with waiting times $\tau = 0.2$ as well as $\tau = 1 - \left(1/k+1\right)^{1/k}$.

6.1.2 Experimental Results

When weights are sampled based on the procedure explained in (i), we observe an interesting phenomenon (see Figure 3). For the classical secretary algorithm, we achieve approximately the tight guarantee of $1/e$. Our algorithm, however, achieves a competitive ratio of approximately 0.8 for $\tau = 0.2$. When having a waiting time depending on k, we improve the competitive ratio for large k while suffering a worse ratio for small k. This can be explained as follows. Weights which are distributed according to (i) almost always have a very large gap between the highest and second highest weight. Hence, no matter which gap we observe, it will always be sufficiently large to exclude all elements except the best one. Therefore, we only incur a loss if we do not accept anything (which happens if and only if the best element arrives before the waiting time). As a consequence, for $\tau = 0.2$, we observe the ratio of 0.8 (which is the probability of the highest weight arriving after time τ). For the waiting times depending on k, the waiting time turns out to be larger for smaller k and vice versa. The improvement in the competitive ratio for large k comes from the reduced waiting time and hence a smaller probability of facing an arrival of w_1 during the waiting period.

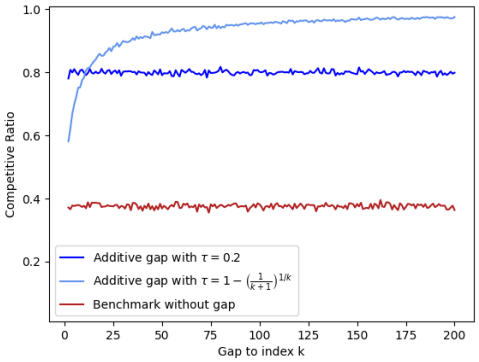

Figure 3: Competitive ratios for weights based on (i). On the x-axis, we have the index k from 2 to n. The y-axis shows the competitive ratios.

Interestingly, this shows that there are instances for which the classical secretary algorithm almost obtains its tight guarantee of $1/e$ while these instances become easy when knowing an additive gap. As a side remark: One might wonder if it is always true that the index k does not play a pivotal role when using a constant waiting time $\tau = 0.2$. In the full version [Braun and Sarkar, 2024], we show that this is not the case for exponentially distributed weights as in (ii) or Chi-Squared distributed ones as in (iii).

6.2 Dealing with Inaccurate Gaps

In order to get a better understanding concerning inaccuracies in the gap, we run a simulation with different errors.

6.2.1 Experimental Setup

Again, we average over 5000 iterations. In each iteration, we set $n = 200$, draw arrival times as before and weights as follows:

(iv) *Exponential*: Here, all $w_i \sim \text{Exp}(1)$.

(v) *Exponential with superstar*: Here, $w_i \sim \text{Exp}(1)$ for $n - 1$ weights and we add a superstar element with weight $100 \cdot \max_i w_i$.

We compare Algorithm 1 to Algorithm 2 both with waiting time $\tau = 0.2$. In addition, Algorithm 2 will drop the gap from the threshold after a time of $1 - \gamma = 0.95$, in other words $\gamma = 0.05$.

The comparison is done for three different gaps: A small one where $k = 2$, i.e. the gap between the largest and second largest element, $k = n/2$ and $k = n$, i.e. the gap to the smallest element. Given a multiplication factor σ for the error, we feed our algorithm with a predicted gap $\hat{c}_k = \sigma \cdot c_k$ for σ going from zero to three in step size of 0.1. In other words, for $\sigma = 1$, we get an accurate gap, for

$\sigma < 1$, we underestimate the gap, for $\sigma > 1$ we overestimate the gap and for $\sigma = 0$, the algorithms are equivalent to the classical secretary algorithms with waiting time τ.

6.2.2 Experimental Results

For exponentially distributed weights (see Figure 4), we can observe that underestimating the gap does not cause too many issues. In particular, when highly underestimating the gap (i.e. $\sigma < 0.5$), both algorithms achieve a competitive ratio of approximately 0.65, similar to an algorithm not knowing any gap. For an accurate gap, $\sigma = 1$, larger gaps are more helpful as they block more elements from being considered. Still, $\sigma > 1$ introduces a transition. For $\sigma > 1$ and gaps between the best and a small element (e.g. $k = 100$ or $k = 200$), overestimating the gap reduces the selection probability of *any* weight of Algorithm 1 to zero: The predicted gap is simply too large and even exceeds w_1. Still, Algorithm 2 is robust in a sense that we still achieve a competitive ratio of approximately 0.15. This constant depends on our choice of γ. As mentioned before, there is the natural trade-off: Increasing γ for an improved robustness and suffer a decrease in the competitive ratio for $\sigma = 1$.

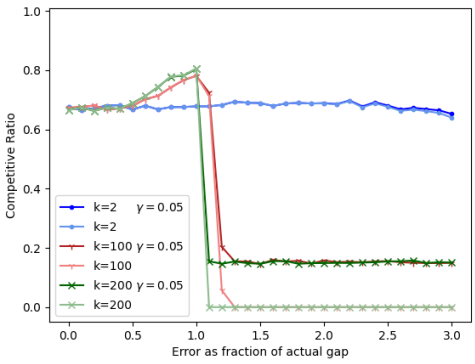

Figure 4: Competitive ratios for weights based on (iv). The x-axis shows σ, where the predicted gap $\hat{c}_k$ used by the algorithms satisfies $\hat{c}_k = \sigma \cdot c_k$ for $\sigma \in [0, 3]$.

Interestingly, for the gap between the best and second best element, both algorithms are much more robust. This can be explained as the gap is small in this case anyway, so overestimating by a factor of three does not cause too much issues yet. One would require to overestimate by a much larger factor here to see a significant difference in the performance of both algorithms.

In the full version [Braun and Sarkar, 2024], we show what happens when shifting our perspective towards the more adversarial setting of exponential weights with one additional superstar as listed in (v). Here, the drop when overestimating is even more significant. Still, Algorithm 2 achieves the desired constant competitive ratio even if gaps are fairly inaccurate. However, the trade-off between consistency and robustness plays a much more important role in the choices of τ and γ here.

7 Conclusion and Future Directions

As we have seen, a single simple piece of information of the form "There is a gap of c in the instance" helps to improve the competitive ratio for the secretary problem. In addition, our algorithm can be made robust against inaccurate predictions without sacrificing too much in the competitive ratio.

Our results directly impose some open questions for future research. First, our guarantees seem to be not tight. Can we achieve a better competitive ratio for any gap? Or is there a matching hardness result? As a second open question, the gaps that we consider are of the form $w_1 - w_k$ for some k. As a generalization, one could consider arbitrary gaps $w_i - w_j$ for some $1 \leq i < j \leq n$. Can we do something in this regime? (as sketched in the full version [Braun and Sarkar, 2024], we can for e.g. $w_2 - w_3 = 0$).
Also, going beyond the single selection problem is interesting, for example by considering the multi-selection variant. For this, we give a reasonable starting point in the full version [Braun and Sarkar, 2024].

Acknowledgments and Disclosure of Funding

The authors would like to thank Thomas Kesselheim for very helpful discussions on the secretary problem as well as suggestions on different models of advice. Also, many thanks to the anonymous reviewers for helpful feedback and remarks concerning previous versions of this paper. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing. Alexander Braun has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project No. 437739576.

References

S. Ahmadian, H. Esfandiari, V. Mirrokni, and B. Peng. Robust load balancing with machine learned advice. *J. Mach. Learn. Res.*, 24:44:1–44:46, 2023. URL http://jmlr.org/papers/v24/22-0629.html.

Algorithms-with-Predictions. Website with list of papers on algorithms with predictions. URL https://algorithms-with-predictions.github.io.

M. Almanza, F. Chierichetti, S. Lattanzi, A. Panconesi, and G. Re. Online facility location with multiple advice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4661–4673. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/250473494b245120a7eaf8b2e6b1f17c-Paper.pdf.

S. Angelopoulos. Online search with a hint. In J. R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 51:1–51:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPICS.ITCS.2021.51. URL https://doi.org/10.4230/LIPIcs.ITCS.2021.51.

A. Antoniadis, T. Gouleakis, P. Kleer, and P. Kolev. Secretary and online matching problems with machine learned advice. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7933–7944. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/5a378f8490c8d6af8647a753812f6e31-Paper.pdf.

A. Antoniadis, J. Boyar, M. Eliás, L. M. Favrholdt, R. Hoeksma, K. S. Larsen, A. Polak, and B. Simon. Paging with succinct predictions. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 952–968. PMLR, 2023. URL https://proceedings.mlr.press/v202/antoniadis23a.html.

C. Argue, A. Gupta, M. Molinaro, and S. Singla. *Robust Secretary and Prophet Algorithms for Packing Integer Programs*, pages 1273–1297. 2022. doi: 10.1137/1.9781611977073.53. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977073.53.

H. Asi, V. Feldman, T. Koren, and K. Talwar. Private online prediction from experts: Separations and faster rates. In G. Neu and L. Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 674–699. PMLR, 2023. URL https://proceedings.mlr.press/v195/asi23a.html.

M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In M. Charikar, K. Jansen, O. Reingold, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*, volume 4627 of *Lecture Notes in Computer Science*, pages 16–28. Springer, 2007. doi: 10.1007/978-3-540-74208-1_2. URL https://doi.org/10.1007/978-3-540-74208-1_2.

M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. Matroid secretary problems. *J. ACM*, 65 (6), nov 2018. ISSN 0004-5411. doi: 10.1145/3212512. URL https://doi.org/10.1145/3212512.

Z. Benomar and V. Perchet. Non-clairvoyant scheduling with partial predictions. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jJLcXGB2uA.

D. Bradac, A. Gupta, S. Singla, and G. Zuzic. Robust Algorithms for the Secretary Problem. In T. Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:26, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-134-4. doi: 10.4230/LIPIcs.ITCS.2020.32. URL https://drops.dagstuhl.de/opus/volltexte/2020/11717.

A. Braun and S. Sarkar. The secretary problem with predicted additive gap, 2024. URL https://arxiv.org/abs/2409.20460.

N. Buchbinder, K. Jain, and M. Singh. Secretary problems via linear programming. *Math. Oper. Res.*, 39(1):190–206, 2014. doi: 10.1287/moor.2013.0604. URL https://doi.org/10.1287/moor.2013.0604.

T. H. Chan, F. Chen, and S. H. Jiang. Revealing optimal thresholds for generalized secretary problem via continuous LP: impacts on online K-item auction and bipartite K-matching with random arrival order. In P. Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1169–1188. SIAM, 2015. doi: 10.1137/1.9781611973730.78. URL https://doi.org/10.1137/1.9781611973730.78.

J. Correa, P. Dütting, F. Fischer, and K. Schewior. Prophet inequalities for i.i.d. random variables from an unknown distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 3–17, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367929. doi: 10.1145/3328526.3329627. URL https://doi.org/10.1145/3328526.3329627.

J. R. Correa, A. Cristi, B. Epstein, and J. A. Soto. Sample-driven optimal stopping: From the secretary problem to the i.i.d. prophet inequality. *CoRR*, abs/2011.06516, 2020. URL https://arxiv.org/abs/2011.06516.

J. R. Correa, A. Cristi, L. Feuilloley, T. Oosterwijk, and A. Tsigonias-Dimitriadis. The secretary problem with independent sampling. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms,SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2047–2058. SIAM, 2021. doi: 10.1137/1.9781611976465.122. URL https://doi.org/10.1137/1.9781611976465.122.

P. Dütting, S. Lattanzi, R. Paes Leme, and S. Vassilvitskii. Secretaries with advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, page 409–429, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385541. doi: 10.1145/3465456.3467623. URL https://doi.org/10.1145/3465456.3467623.

E. B. Dynkin. The optimum choice of the instant for stopping a markov process. *Soviet Math. Dokl*, 4:627–629, 1963.

M. Feldman, O. Svensson, and R. Zenklusen. A simple $O(\log \log(\text{rank}))$-competitive algorithm for the matroid secretary problem. *Math. Oper. Res.*, 43(2):638–650, 2018. doi: 10.1287/moor.2017.0876. URL https://doi.org/10.1287/moor.2017.0876.

T. S. Ferguson. Who solved the secretary problem? *Statistical Science*, 4(3):282–289, 1989. ISSN 08834237.

P. R. Freeman. The secretary problem and its extensions: A review. *International Statistical Review / Revue Internationale de Statistique*, 51(2):189–206, 1983. ISSN 03067734, 17515823.

K. Fujii and Y. Yoshida. The secretary problem with predictions. *CoRR*, abs/2306.08340, 2023. doi: 10.48550/ARXIV.2306.08340. URL https://doi.org/10.48550/arXiv.2306.08340.

J. P. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313):35–73, 1966. ISSN 01621459.

S. Im, R. Kumar, M. Montazer Qaem, and M. Purohit. Online knapsack with frequency predictions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2733–2743. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/161c5c5ad51fcc884157890511b3c8b0-Paper.pdf.

S. Im, R. Kumar, A. Petety, and M. Purohit. Parsimonious learning-augmented caching. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9588–9601. PMLR, 2022. URL https://proceedings.mlr.press/v162/im22a.html.

H. Kaplan, D. Naori, and D. Raz. *Competitive Analysis with a Sample and the Secretary Problem*, pages 2082–2095. 2020. doi: 10.1137/1.9781611975994.128. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611975994.128.

T. Kesselheim and M. Molinaro. Knapsack Secretary with Bursty Adversary. In A. Czumaj, A. Dawar, and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-138-2. doi: 10.4230/LIPIcs.ICALP.2020.72. URL https://drops.dagstuhl.de/opus/volltexte/2020/12479.

T. Kesselheim, K. Radke, A. Tönnis, and B. Vöcking. Primal beats dual on online packing lps in the random-order model. *SIAM Journal on Computing*, 47(5):1939–1964, 2018. doi: 10.1137/15M1033708. URL https://doi.org/10.1137/15M1033708.

R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, page 630–631, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0898715857.

R. D. Kleinberg and F. T. Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 594–605. IEEE Computer Society, 2003. doi: 10.1109/SFCS.2003.1238232. URL https://doi.org/10.1109/SFCS.2003.1238232.

N. Korula and M. Pál. Algorithms for secretary problems on graphs and hypergraphs. In *Proceedings of the 36th Internatilonal Collogquium on Automata, Languages and Programming: Part II*, ICALP '09, page 508–520, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 9783642029295. doi: 10.1007/978-3-642-02930-1_42. URL https://doi.org/10.1007/978-3-642-02930-1_42.

T. Lavastida, B. Moseley, R. Ravi, and C. Xu. Learnable and Instance-Robust Predictions for Online Matching, Flows and Load Balancing. In P. Mutzel, R. Pagh, and G. Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 59:1–59:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-204-4. doi: 10.4230/LIPIcs.ESA.2021.59. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2021.59.

R. P. Leme, B. Sivan, Y. Teng, and P. Worah. Pricing query complexity of revenue maximization. In N. Bansal and V. Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 399–415. SIAM, 2023. doi: 10.1137/1.9781611977554.CH17. URL https://doi.org/10.1137/1.9781611977554.ch17.

D. V. Lindley. Dynamic programming and decision theory. *Journal of The Royal Statistical Society Series C-applied Statistics*, 10:39–51, 1961.

T. Lykouris and S. Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4):24:1–24:25, 2021. doi: 10.1145/3447579. URL https://doi.org/10.1145/3447579.

M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 597–606, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306911. doi: 10.1145/1993636.1993716. URL https://doi.org/10.1145/1993636.1993716.

M. Purohit, Z. Svitkina, and R. Kumar. Improving online algorithms via ML predictions. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9684–9693, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html.

A. Rubinstein. Beyond matroids: Secretary problem and prophet inequality with general constraints. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 324–332, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341325. doi: 10.1145/2897518.2897540. URL https://doi.org/10.1145/2897518.2897540.

A. Wei and F. Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

A. Zeynali, B. Sun, M. Hajiesmaili, and A. Wierman. Data-driven competitive algorithms for online knapsack and set cover. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12): 10833–10841, May 2021. doi: 10.1609/aaai.v35i12.17294. URL https://ojs.aaai.org/index.php/AAAI/article/view/17294.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Proofs for the results and statements made in the abstract and introduction are given in the main body and appendix.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the algorithms in the paper are discussed in the main body at several positions. In addition, improved approaches are given which overcome these limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Assumptions are mentioned clearly with the respective statements. Proofs are provided mainly in the appendix due to space constraints.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides all information such that the simulations can easily be reproduced. Algorithms should be easily implementable within a few lines, sample sizes and distributions are stated.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will make the code and data publicly available after the review process.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: There was no training, neither a testing phase. Simulations only contain samples from distributions and the performance of our algorithms.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: See answer of Question 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Hardware settings are provided as a footnote in the main body of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The goal of our paper is to gain a better understanding of different kinds of predictions compared to the ones studied in the literature before. There are no further societal consequences beyond those for algorithms with predictions.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not pose any risks. There are no further consequences beyond those for theoretical results on algorithms with predictions.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use any code or data from previous work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The code for the simulations in the paper is straight-forward. We will make a well-documented version publicly available after the review process.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.