No-Regret Learning in Partially-Informed Auctions

Wenshuo Guo [1] **Michael I. Jordan** [1 2] **Ellen Vitercik** [3]

Abstract

Auctions with partially-revealed information about items are broadly employed in real-world applications, but the underlying mechanisms have limited theoretical support. In this work, we study a machine learning formulation of these types of mechanisms, presenting algorithms that are no-regret from the buyer's perspective. Specifically, a buyer who wishes to maximize his utility interacts repeatedly with a platform over a series of T rounds. In each round, a new item is drawn from an unknown distribution and the platform publishes a price together with incomplete, "masked" information about the item. The buyer then decides whether to purchase the item. We formalize this problem as an online learning task where the goal is to have low regret with respect to a myopic oracle that has perfect knowledge of the distribution over items and the seller's masking function. When the distribution over items is known to the buyer and the mask is a SimHash function mapping $\mathbb{R}^d$ to $\{0,1\}^\ell$, our algorithm has regret $\tilde{\mathcal{O}}((Td\ell)^{1/2})$. In a fully agnostic setting when the mask is an arbitrary function mapping to a set of size n and the prices are stochastic, our algorithm has regret $\tilde{\mathcal{O}}((Tn)^{1/2})$.

1. Introduction

Selling mechanisms play a crucial role in economic theory and have a wide range of applications across many industries (Post et al., 1995; Milgrom, 2004; Edelman et al., 2007; Milgrom, 2010; Arnosti et al., 2016). Under the canonical mechanism design model, buyers choose whether or not to buy items for sale based on their true values for those items. This fundamental model, however, assumes that the buyers know exactly how much they value the items for sale, which is often not the case.

One of the overriding reasons that a buyer may not know their true values is information asymmetry: the seller may purposefully obfuscate information about an item for sale. For example, the seller may hide information about the item in the hopes of better revenue (Gershkov, 2009). Alternatively, information about the item may be private, and thus the seller may wish to protect this sensitive information by only revealing partial information about the item. For instance, in online advertising auctions, bids represent how much advertisers are willing to pay to display their ad to a particular user. Historically, advertisers have bid based on uniquely identifying information about users, but there has been a growing effort to protect users' privacy by obfuscating this sensitive information (Juels, 2001; Guha et al., 2011; Epasto et al., 2021).

In these scenarios, the buyer only has partial information about the item for sale but still must decide whether to make a purchase. This raises the question: **how should a buyer determine their purchase strategy with only incomplete item information?**

We study posted-price auctions—a fundamental mechanism family that is appealingly interpretable—with incomplete item information. In particular, the seller reveals obfuscated ("masked") information about the item using a fixed, unknown masking function. We study an online setting where, at each round, a fresh item is drawn from an unknown distribution (for example, a distribution over users visiting a webpage). The seller sets a price and the buyer chooses whether to buy the item based on the incomplete information that the seller provides. We propose no-regret learning algorithms for the buyer that achieve sub-linear regret compared to an oracle buyer who has perfect knowledge of the item distribution as well as the seller's masking function.

1.1. Our results

We study no-regret learning with incomplete item information in two settings:

1. First, we propose an algorithm for a setting where the item distribution is known to the buyer and the mask

[1]Department of Electrical Engineering & Computer Sciences, University of California, Berkeley, USA [2]Department of Statistics, University of California, Berkeley, USA [3]Department of Management Science & Engineering and Department of Computer Science, Stanford University, USA. Correspondence to: Wenshuo Guo <wguo@cs.berkeley.edu>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Item distribution	Prices	Masking function h	Regret
Known	Adversarial	SimHash $h : [0,1]^d \to \{0,1\}^\ell$	$\mathcal{O}(\sqrt{Td\ell \log(T\ell/\delta)})$ (Theorem 3.5)
Unknown	Stochastic	Arbitrary $h : \mathcal{X} \to [n]$	$\mathcal{O}(\sqrt{T(n \log T/n + \log 1/\delta)})$ (Theorem 4.4)
Unknown	Adversarial	Arbitrary $h : \mathcal{X} \to [n]$	$\tilde{\mathcal{O}}(T^{2/3} n^{1/3})$ (Remark 4.5)

Table 1. Summary of regret bounds which hold with probability at least $1 - \delta$.

is a SimHash function mapping $[0,1]^d$ to $\{0,1\}^\ell$. In other words, each item is defined by d real-valued features and the seller reveals ℓ bits about the item to the buyer, as defined by a function that is unknown to the buyer. This model has been studied from an applied perspective in the context of ad auctions (Epasto et al., 2021). We provide an algorithm with regret $\mathcal{O}(\sqrt{Td\ell \log(T\ell/\delta)})$.

2. Next, we study a setting where the masking function is an arbitrary mapping from the set of all items, denoted $\mathcal{X}$, to a finite set of size n. We propose an online learning algorithm with regret $\mathcal{O}(\sqrt{T(n \log T/n + \log 1/\delta)})$ when the prices are stochastic, where T is the length of the horizon.

In the first setting where the masking function is a SimHash function mapping $[0,1]^d$ to $\{0,1\}^\ell$, the domain of the masking function is of size $n = 2^\ell$, so our regret bound of $\tilde{\mathcal{O}}((Td\ell)^{1/2})$ is exponentially better than the latter regret bound.

We summarize these results in Table 1.

1.2. Related work

This work draws on several threads of research on designing auctions with incomplete information, learning to bid, and privacy-preserving simple auctions.

Auction design with incomplete information. Auctions with incomplete value information have attracted much research attention. Several prior works have explored auction design where the information about the item may be incomplete to the buyer or the seller (Ganuza, 2004; Eső & Szentes, 2007; Bergemann & Pesendorfer, 2007; Arefeva & Meng, 2021; Roesler & Szentes, 2017; Li & Shi, 2017; Bergemann et al., 2021; Li & Jewitt, 2017). In particular, Ganuza (2004) studied the incentives of the auctioneer to release signals about the item to the buyers that refine their private valuations before a second-price auction. In their model, the seller reveals a noisy item feature vector which is an unbiased estimator of the true one. Bergemann & Pesendorfer (2007) considered single-item multi-bidder auctions where the seller decides how accurately the bidders can learn their valuations. However, all of this prior work is limited to offline settings; they did not explore online purchase strategies that the buyer can adopt. To the best of our knowledge, this work constitutes the first analysis of no-regret learning algorithms in partially-informed posted-price auctions.

Learning to bid with an unknown value. Instead of focusing the problem from the side of the auctioneer who aims to maximize revenue over repeated rounds, another active line of research studies bidding strategies for the bidders when they do not know their values (Dikkala & Tardos, 2013; Weed et al., 2016; Feng et al., 2018; Balseiro & Gur, 2019). Feng et al. (2018) considered a single-item multi-bidder setting where the bidder learns to bid via partial feedback, and provide algorithms with regret rates against the best fixed bid in hindsight. Dikkala & Tardos (2013) explored a setting where bidders need to experiment in order to learn their valuations. The key difference between this work and ours is that our algorithms exploit the available "partial" information instead of having zero knowledge about the item being sold. This partial information model allows us to trade off between the amount of information revealed about the item and the regret. Moreover, we compete with the best purchase policy, rather than the best fixed bid in hindsight.

Private auctions. Our theoretical model is motivated by recent work on designing privacy-enhanced auctions for practical usage. An important application of these auctions is online advertising, where the items being auctioned are user queries and the auctioner must trade off between user privacy and revenue maximization (De Corniere & De Nijs, 2016; Rafieian & Yoganarasimhan, 2021; Epasto et al., 2021; Guha et al., 2011; Juels, 2001). Epasto et al. (2021) present a detailed exploration of Chrome's Federated Learning of Cohorts (FLoC) API, where user information is masked. Our contribution is to provide a formal treatment of such auctions, including providing algorithms that have theoretical guarantees in the setting of arbitrary masking functions that are unknown to the buyer.

2. Preliminaries

We begin by defining formally the setting we study for auctions with partial information.

2.1. Setup

We consider a setting where there is a single seller and a single buyer. There is a distribution $\mathcal{P}$ over items which are elements of an abstract set $\mathcal{X}$. The buyer has a bounded valuation, $v^*(\mathbf{x}) \in [0, H]$ for every item $\mathbf{x} \in \mathcal{X}$ and some $H \in \mathbb{R}_+$[1]. The seller and buyer interact over a series of T rounds. At each round $t \in [T]$, the seller draws an item $\mathbf{x}_t \overset{iid}{\sim} \mathcal{P}$ and sets a price $p_t(\mathbf{x}_t) \in [0, H]$ for the item. The seller does not reveal $\mathbf{x}_t$ to the buyer, but rather reveals some partial information $h(\mathbf{x}_t) \in \mathcal{Y}$ where $h : \mathcal{X} \to \mathcal{Y}$ is a fixed "masking" function that maps to an abstract set $\mathcal{Y}$.

We assume that the price function p_t does not give any more information about $\mathbf{x}_t$ than that is provided by the masking function h. In other words, if $h(\mathbf{x}) = h(\mathbf{x}')$, then $p_t(\mathbf{x}) = p_t(\mathbf{x}')$, which implies

$$\mathbb{E}[v^*(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})] = \mathbb{E}[v^*(\mathbf{x}) \mid h(\mathbf{x})]. \tag{1}$$

The buyer uses a strategy $s_t : \mathcal{Y} \times \mathbb{R} \to \{0, 1\}$ to decide whether to buy the item given the partially revealed item information and the price. Here $s_t(h(\mathbf{x}_t), p_t(\mathbf{x}_t)) = 1$ if and only if the buyer buys the item. Letting $b_t = s_t(h(\mathbf{x}_t), p_t(\mathbf{x}_t))$, the buyer's utility is $u_t = b_t(v^*(\mathbf{x}_t) - p_t(\mathbf{x}_t)) \in [-H, H]$. We summarize this process below.

Model Online model

for $t = 1, 2, \ldots, T$ **do**
  Seller selects a price function $p_t : \mathcal{X} \to [0, H]$.
  Item $\mathbf{x}_t$ is sampled from $\mathcal{P}$.
  Seller publishes item information $h(\mathbf{x}_t)$ and a price $p_t(\mathbf{x}_t)$.
  Buyer decides whether or not to buy: $b_t = s_t(h(\mathbf{x}_t), p_t(\mathbf{x}_t)) \in \{0, 1\}$.
  Buyer obtains reward $u_t = (v^*(\mathbf{x}_t) - p_t(\mathbf{x}_t)) \cdot b_t$.
  if b_t **then** {item is purchased}
    Buyer observes $\mathbf{x}_t$.
  end if
end for

Example 2.1 (Advertising auctions)**.** We instantiate our model in the context of advertising auctions, taking inspiration from Epasto et al. (2021). The seller is a platform and the buyer is an advertiser. Each item $\mathbf{x} \in \mathcal{X}$ describes a user who visits the platform. For example, $\mathcal{X} = \mathbb{R}^d$ might denote features that uniquely identify each user. On round t, the advertiser has a value $v^*(\mathbf{x}_t)$ for the opportunity to show the user $\mathbf{x}_t$ an ad. In order to preserve user privacy, the platform does not reveal $\mathbf{x}_t$ to the advertiser, but rather some summary $h(\mathbf{x}_t)$. For example, Epasto et al. (2021) study a setting where h is a SimHash function, so $\mathbf{x}_t \in \mathbb{R}^d$ and $h(\mathbf{x}_t) \in \mathbb{R}^\ell$ for some $\ell < d$. The platform sets a price $p_t(\mathbf{x}_t)$ which the advertiser pays to show the user an ad.

We study this problem from the perspective of the buyer: how should they select the strategy s_t at each round to maximize their utility? We study two settings: a setting where the distribution $\mathcal{P}$ is unknown to the buyer and the prices are stochastic (Section 4) and a model where $\mathcal{P}$ is known to the buyer with adversarial prices (Section 3).

2.2. Regret and the optimal strategy

We measure the regret of the buyer in our online model with regard to the optimal strategy s^* of a myopic buyer[2] who has perfect knowledge of the distribution $\mathcal{P}$ and the masking function h, but not the realized item $\mathbf{x}_t$. To make this dependence on the environment clear, in any single round, we use the notation $s^*(h(\mathbf{x}), p(\mathbf{x}), h, \mathcal{P})$ to denote the optimal strategy (we drop the subscript t for simplicity). More formally, s^* maximizes the expected utility:

$$\operatorname*{argmax}_{s \in \mathcal{S}} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{P}}[(v^*(\mathbf{x}) - p(\mathbf{x}))s(h(\mathbf{x}), p(\mathbf{x}), h, \mathcal{P}) \mid h(\mathbf{x})],$$

where $\mathcal{S}$ represents the set of all decision functions $s(\cdot) : \mathcal{Y} \times \mathbb{R} \times \{h(\cdot), \mathcal{P}\} \to \{0, 1\}$.

Definition 2.2. The buyer's (expected) regret with respect to the optimal strategy s^*, denoted R_T, is defined as

$$\mathbb{E}\left[\sum_{t=1}^{T} \Big(v^*(\mathbf{x}_t) - p_t(\mathbf{x}_t)\Big) s^*\left(h(\mathbf{x}_t), p_t(\mathbf{x}_t), h(\cdot), \mathcal{P}\right) - \Big(v^*(\mathbf{x}_t) - p_t(\mathbf{x}_t)\Big) s_t\left(h(\mathbf{x}_t), p_t(\mathbf{x}_t)\right)\right]. \tag{2}$$

In the following proposition, we identify the form of the optimal strategy s^*. The proof is in Appendix A.

Proposition 2.3. *The strategy s^* that maximizes $\mathbb{E}_{\mathbf{x} \sim \mathcal{P}}[(v^*(\mathbf{x}) - p(\mathbf{x}))s^*(h(\mathbf{x}), p(\mathbf{x}), h, \mathcal{P}) \mid h(\mathbf{x})]$ is*

$$s^*(h(\mathbf{x}), p(\mathbf{x}), h, \mathcal{P}) = \mathbb{I}\left(\mathop{\mathbb{E}}_{x \sim \mathcal{P}}[v^*(\mathbf{x}) \mid h(\mathbf{x})] > p(\mathbf{x})\right).$$

Even when the buyer has no information about the distribution $\mathcal{P}$ (Section 4), we show that he can guarantee low regret with respect to s^* with either stochastic or adversarial prices, in polynomial per-round runtime. When the distribution $\mathcal{P}$ is known (Section 3), we provide an algorithm with exponentially better regret.

3. Known Item Distribution

First, we focus on a specific class of masking functions, SimHash, motivated by recent practical applications in ad

[1] The assumption that there is a bound on the maximum amount that the agents value the item is widely made in prior research. In our main application—ad auctions—the value of an impression is typically very cheap, so this assumption is mild.

[2] A myopic buyer optimizes his utility separately in each round.

auctions (Epasto et al., 2021). Here, $\mathcal{X}$ is a feature space $[0,1]^d$ and the the masking function h is a SimHash function that is unknown to the buyer. In other words, there are ℓ unknown vectors $\mathbf{w}_1, \ldots, \mathbf{w}_\ell \in \mathbb{R}^d$ such that the masking function, denoted as $h_{\mathbf{w}}$ with $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_\ell)$, is $h_{\mathbf{w}}(\mathbf{x}) = (\mathrm{sgn}(\mathbf{w}_1 \cdot \mathbf{x}), \ldots, \mathrm{sgn}(\mathbf{w}_\ell \cdot \mathbf{x}))^\top$.

We consider the setting where the distribution of the items is known to the buyer (for example, via historical data). We provide an algorithm that achieves a regret $\tilde{\mathcal{O}}(\sqrt{Td\ell})$ even under adversarial prices. Since the masking function maps to a set of size $n = 2^\ell$, the regret only depends logarithmically on n. As we detail in the subsequent Section 4, this algorithm achieves exponentially better regret compared to the algorithm we present where the distribution $\mathcal{P}$ over items is unknown and the masking function is arbitrary.

Algorithm 1 Explore-then-Commit (Known Distribution)

Input: horizon T**, distribution** $\mathcal{P}$**,** $d, \ell \in \mathbb{N}_+$**, and** $\delta \in (0,1)$**.**
Compute $t' = \sqrt{4Td\ell \log(\ell/\delta)}$.
for $t = 1, 2, \ldots, t'$ **do** {Exploration phase}
  Receive $h(\mathbf{x}_t)$, price $p_t(\mathbf{x}_t)$ where $\mathbf{x}_t \overset{iid}{\sim} \mathcal{P}$.
  Make decision $b_t = 1$ and observe $\mathbf{x}_t$.
end for
Use linear programming to compute $\hat{\mathbf{w}} = (\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_\ell)$ such that $h_{\hat{\mathbf{w}}}(\mathbf{x}_i) = h_{\mathbf{w}}(\mathbf{x}_i)$ for all $i \in [t']$.
for $t = t'+1, t'+2, \ldots, T$ **do** {Exploitation phase}
  Receive $h_{\mathbf{w}}(\mathbf{x}_t)$, price $p_t(\mathbf{x}_t)$ where $\mathbf{x}_t \overset{iid}{\sim} \mathcal{P}$.
  Obtain an estimate $\hat{Z}_t$ of $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\hat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))]$ using the Integration Algorithm by Lovász & Vempala (2006).
  Make decision $b_t = \mathbb{I}(\hat{Z}_t \geq p_t(\mathbf{x}_t))$.
end for

Algorithm 1 begins with an exploration phase of length $t' = \tilde{O}(\sqrt{Td\ell})$, during which the buyer buys the item in each round. The algorithm then uses linear programming to solve for separators $\hat{\mathbf{w}} = (\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_\ell)$, $\hat{\mathbf{w}}_j \in \mathbb{R}^d$ for all $j \in [\ell]$, such that $\mathrm{sgn}(\mathbf{w}_j \cdot \mathbf{x}_i) = \mathrm{sgn}(\hat{\mathbf{w}}_j \cdot \mathbf{x}_i)$ for all $j \in [\ell]$ and $i \in [t']$. During the rest of the rounds $t \in \{t'+1, \ldots, T\}$, the algorithm exploits. Since the optimal strategy is to buy if $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid h_{\mathbf{w}}(\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x}_t)] \geq p(\mathbf{x}_t)$ (Prop. 2.3), the algorithm uses $h_{\mathbf{w}}(\mathbf{x}_t)$ and $\hat{\mathbf{w}} = (\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_\ell)$ to compute an estimate of $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid h_{\mathbf{w}}(\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x}_t)]$.

The intuition behind the estimate is the following. Letting $h_{\mathbf{w}}^{-1}(\mathbf{x}_t) = \{\mathbf{x} : h_{\mathbf{w}}(\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x}_t)\}$ (a convex polytope), we have that $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid h_{\mathbf{w}}(\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x}_t)] = \mathbb{E}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right]$. Since the buyer does not know $\mathbf{w}$, we cannot compute the set $h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))$, but we can compute the set $h_{\hat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))$ using the estimated separators that we have obtained after the exploration phase. Even still, the conditional expectation $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\hat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right]$ may be challenging to compute in high dimensions. Therefore, we use a sampling algorithm by Lovász & Vempala (2006) to compute an estimate Z_t of $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\hat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))]$. The buyer buys the item if $Z_t \geq p_t(\mathbf{x}_t)$.

To compute the estimate Z_t, Lovász & Vempala (2006) require that if π is the density function of $\mathcal{P}$, then $v^*(\mathbf{x})\pi(\mathbf{x})$ is log-concave and "well-rounded." Many well-studied distributions are log-concave, including the normal, exponential, uniform, and beta distributions, among many others. Moreover, every concave function that is nonnegative on its domain is log-concave. If v^* and π are log-concave, then $v^*(\mathbf{x})\pi(\mathbf{x})$ is also log-concave. For example, the Cartesian product of single-dimensional log-concave distributions (exponential, logistic, extreme value, Laplace, and beta distributions, among many others) is log-concave. Log-concavity has also been widely-assumed in prior works in machine learning and high-dimensional statistics (e.g., Bagnoli & Bergstrom, 2006; Saumard & Wellner, 2014).

The function $v^*(\mathbf{x})\pi(\mathbf{x})$ is well-rounded if for any $\mathcal{A} \subseteq \mathcal{X}$, the distribution defined by $f_\pi(\mathcal{A}) = \frac{\int_{\mathcal{A}} v^*(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}}{\int_{\mathcal{X}} v^*(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}}$ is neither too spread out nor too concentrated. We include the formal definition in Appendix B (Def. B.1). Every log-concave function can be brought to a well-rounded position by an affine transformation of the space in polynomial time (Lovász & Vempala, 2006).

Regret analysis. We now prove that the regret of Algorithm 1 is $\tilde{O}(\sqrt{Td\ell})$. To do so, we must contend with two sources of error: the fact that we use the learned linear separators $\hat{\mathbf{w}}$ instead of $\mathbf{w}$ and the estimation error introduced by the sampling algorithm.

We begin our analysis by proving that for any $\mathbf{y} \in \{0,1\}^\ell$ in the image of $h_{\mathbf{w}} : \mathcal{X} \to \{0,1\}^\ell$, the agent's expected value conditioned on $\mathbf{x} \in h_{\hat{\mathbf{w}}}^{-1}(\mathbf{y})$ is close to its true expected value conditioned on $\mathbf{x} \in h_{\mathbf{w}}^{-1}(\mathbf{y})$. In this section, we use the notation $\epsilon = \frac{\ell}{t'}\left(d \ln \frac{2et'}{d} + \ln \frac{2\ell}{\delta}\right)$.

Lemma 3.1. *For any* $\mathbf{y} \in \{0,1\}^\ell$*, with probability at least* $1-\delta$ *over* $\mathbf{x}_1, \ldots, \mathbf{x}_{t'} \sim \mathcal{P}$*,*

$$\left|\mathbb{E}\left[v^*(\mathbf{x}) | \mathbf{x} \in h_{\hat{\mathbf{w}}}^{-1}(\mathbf{y})\right] - \mathbb{E}\left[v^*(\mathbf{x}) | \mathbf{x} \in h_{\mathbf{w}}^{-1}(\mathbf{y})\right]\right| \leq H\epsilon.$$

Proof. We can decompose the set $h_{\hat{\mathbf{w}}}^{-1}(\mathbf{y})$ as

$$h_{\hat{\mathbf{w}}}^{-1}(\mathbf{y}) = \left(h_{\hat{\mathbf{w}}}^{-1}(\mathbf{y}) \cap h_{\mathbf{w}}^{-1}(\mathbf{y})\right) \cup \left(h_{\hat{\mathbf{w}}}^{-1}(\mathbf{y}) \setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right).$$

We can therefore write

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right]\\ =&\,\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\cap h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\\ &\cdot \Pr\left[\mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\cap h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\\ +&\,\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\\ &\cdot \Pr\left[\mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right].\end{aligned}$$

Similarly, we can write

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\mathbf{w}}^{-1}(\mathbf{y})\right]\\ =&\,\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\cap h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\\ &\cdot \Pr\left[\mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\cap h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\\ +&\,\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\mathbf{w}}^{-1}(\mathbf{y})\setminus h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right)\right]\\ &\cdot \Pr\left[\mathbf{x}\in \left(h_{\mathbf{w}}^{-1}(\mathbf{y})\setminus h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right)\right].\end{aligned}$$

Matching terms, we have that

$$\begin{aligned}&\left|\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right]-\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\mathbf{w}}^{-1}(\mathbf{y})\right]\right|\\ =&\left|\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\right.\\ &\cdot \Pr\left[\mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\\ &-\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\mathbf{w}}^{-1}(\mathbf{y})\setminus h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right)\right]\\ &\left.\cdot \Pr\left[\mathbf{x}\in \left(h_{\mathbf{w}}^{-1}(\mathbf{y})\setminus h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right)\right]\right|.\end{aligned}$$

We know that $\mathbf{x} \in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)$ if and only if $h_{\widehat{\mathbf{w}}}(\mathbf{x}) = \mathbf{y}$ and $h_{\mathbf{w}}(\mathbf{x}) \neq \mathbf{y}$, which means that $h_{\widehat{\mathbf{w}}}(\mathbf{x}) \neq h_{\mathbf{w}}(\mathbf{x})$. The following claim bounds $\Pr[h_{\widehat{\mathbf{w}}}(\mathbf{x}) = \mathbf{y} \text{ and } h_{\mathbf{w}}(\mathbf{x}) \neq \mathbf{y}] \leq \Pr[h_{\widehat{\mathbf{w}}}(\mathbf{x}) \neq h_{\mathbf{w}}(\mathbf{x})]$.

Claim 3.2. *With probability* $1-\delta$, $\Pr_{\mathbf{x}\sim\mathcal{P}}[h_{\widehat{\mathbf{w}}}(\mathbf{x}) \neq h_{\mathbf{w}}(\mathbf{x})] \leq \epsilon$.

Proof of Claim 3.2. For a fixed $i \in [\ell]$, by the standard PAC learning generalization bound in the realizable setting (e.g., Theorem 4.8 by Anthony & Bartlett (2009)), we have that with probability $1-\frac{\delta}{\ell}$,

$$\Pr[\operatorname{sgn}(\mathbf{w}_i\cdot\mathbf{x}) \neq \operatorname{sgn}(\widehat{\mathbf{w}}_i\cdot\mathbf{x})] \leq \frac{1}{t'}\left(d\ln\frac{2et'}{d}+\ln\frac{2\ell}{\delta}\right).$$

Therefore, with probability $1-\delta$,

$$\begin{aligned}&\Pr_{\mathbf{x}\sim\mathcal{P}}[h_{\widehat{\mathbf{w}}}(\mathbf{x}) \neq h_{\mathbf{w}}(\mathbf{x})]\\ =&\Pr[\exists i\in[\ell] \text{ such that } \operatorname{sgn}(\mathbf{w}_i\cdot\mathbf{x}) \neq \operatorname{sgn}(\widehat{\mathbf{w}}_i\cdot\mathbf{x})] \leq \epsilon,\end{aligned}$$

as claimed. □

By Claim 3.2 and the fact that $v^*(\mathbf{x}) \in [0,H]$, we therefore know that $\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right]\cdot \Pr_{\mathbf{x}\sim\mathcal{P}}\left[\mathbf{x}\in \left(h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\setminus h_{\mathbf{w}}^{-1}(\mathbf{y})\right)\right] \in [0,H\epsilon]$. By a symmetric argument, $\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in \left(h_{\mathbf{w}}^{-1}(\mathbf{y})\setminus h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right)\right]\cdot \Pr_{\mathbf{x}\sim\mathcal{P}}\left[\mathbf{x}\in \left(h_{\mathbf{w}}^{-1}(\mathbf{y})\setminus h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right)\right] \in [0,H\epsilon]$. Therefore, the lemma statement holds. □

Lemma 3.1 guarantees that $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\widehat{\mathbf{w}}}^{-1}(\mathbf{y})\right]$ is a good approximation of $\mathbb{E}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\mathbf{w}}^{-1}(\mathbf{y})\right]$ —which is the key quantity needed to compute the optimal policy (see Prop. 2.3). However, this estimate may be difficult to compute when $\mathbf{x}$ is high dimensional, despite the fact that $\mathcal{P}$ is known. The integration algorithm of Lovász & Vempala (2006) allows us to estimate it in polynomial time, as we summarize in the following lemma.

Lemma 3.3. *Suppose that $v^*(\mathbf{x})\pi(\mathbf{x})$ is log-concave and well-rounded. Then for any $\mathbf{y}\in\{0,1\}^\ell$, with probability at least $1-\delta$, we can compute a constant A in polynomial time such that $\left|A-\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x}\in h_{\mathbf{w}}^{-1}(\mathbf{y})\right]\right| \leq H\epsilon$.*

Proof. By Lemma 3.1, with probability at least $1-\delta/2$,

$$\begin{aligned}&\left|\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathcal{P}}\left[v^*(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}}\in h_{\widehat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))\right]\right.\\ &\left.-\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathcal{P}}\left[v^*(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}}\in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))\right]\right| \leq \epsilon H.\end{aligned}$$

By definition

$$\begin{aligned}&\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathcal{P}}[v^*(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}}\in h_{\widehat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))]\\ =&\int_{\tilde{\mathbf{x}}\in h_{\widehat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))} v^*(\tilde{\mathbf{x}})\pi(\tilde{\mathbf{x}})d\tilde{\mathbf{x}}.\end{aligned}$$

Then, by Lovász & Vempala (2006, Theorem 1.3), in polynomial runtime with probability of at least $1-\delta/2$, we can compute a constant A, such that

$$\begin{aligned}&\left|A-\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathcal{P}}\left[v^*(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}}\in h_{\widehat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))\right]\right|\\ \leq&\ \epsilon\cdot\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathcal{P}}\left[v^*(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}}\in h_{\widehat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))\right] \leq \epsilon H.\end{aligned}$$

By the triangle inequality and a union bound, we have that with probability of at least $1-\delta$, we can compute a value A such that

$$\left|A-\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathcal{P}}\left[v^*(\tilde{\mathbf{x}}) \mid \tilde{\mathbf{x}}\in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}))\right]\right| \leq 2\epsilon H,$$

which completes the proof. □

Lemma 3.4. *In each round $t\in\{t'+1,\ldots,T\}$ of the exploitation phase in Algorithm 1, with probability at least $1-\delta$, the expected instantaneous regret incurred in round t is at most*

$$\frac{2H\ell}{t'}\left(d\ln\frac{2et'}{d}+\ln\frac{\ell\cdot 2^{\ell+2}}{\delta}\right).$$

Proof. Given $h_{\mathbf{w}}(\mathbf{x}_t)$ and price $p(\mathbf{x}_t)$, denote the estimated value of $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\widehat{\mathbf{w}}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right]$ obtained using the sampling algorithm (Alg 1, step 1) as $A(h_{\mathbf{w}}(\mathbf{x}_t))$. For simplicity of notation, we denote the decision of the oracle policy as s^* and the decision of the learned policy as s_t:

$$s^* = \mathbb{I}\left(\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right] > p(\mathbf{x}_t)\right),$$
$$s_t = \mathbb{I}\Big(A(h_{\mathbf{w}}(\mathbf{x}_t)) > p(\mathbf{x}_t)\Big).$$

Now we bound the expected instantaneous regret in round t:

$$\mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[(v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s^* - (v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s_t\right]$$
$$= \mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[(v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\,(s^* - s_t)\right].$$

Let Δ denote the difference $\Delta = s^* - s_t$, so $\Delta \in \{-1, 0, 1\}$ is a random variable that depends on $\mathbf{x}_t$. By the law of total expectation,

$$\mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[(v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s^* - (v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s_t\right]$$
$$= \mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[(v^*(\mathbf{x}) - p(\mathbf{x}_t))\,\Delta \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right]\right]$$
$$= \mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[\left(\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) | \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right] - p(\mathbf{x}_t)\right)\Delta\right].$$

The variable Δ is only nonzero when $s^* \neq s_t$. Let E denote the event where $s^* \neq s_t$ and let $p_E = \mathbb{P}_{\mathbf{x}_t\sim\mathcal{P}}[E]$. Then

$$\mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[(v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s^* - (v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s_t\right]$$
$$\leq \mathop{\mathbb{E}}_{\mathbf{x}_t}\left[\left|\mathop{\mathbb{E}}_{\mathbf{x}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right] - p(\mathbf{x}_t)\right| \,\Big|\, E\right] \cdot p_E$$
$$\leq \mathop{\mathbb{E}}_{\mathbf{x}_t}\left[\left|\mathop{\mathbb{E}}_{\mathbf{x}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right] - p(\mathbf{x}_t)\right| \,\Big|\, E\right].$$

By definition, when event E happens, we know that

$$A(h_{\mathbf{w}}(\mathbf{x}_t)) < p(\mathbf{x}_t) \leq \mathop{\mathbb{E}}_{\mathbf{x}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))] \text{ or}$$
$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))] < p(\mathbf{x}_t) \leq A(h_{\mathbf{w}}(\mathbf{x}_t)),$$

where in either case we have that

$$\left|\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))] - p(\mathbf{x}_t)\right|$$
$$\leq \left|\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))] - A(h_{\mathbf{w}}(\mathbf{x}_t))\right|.$$

Given $\delta' \in (0, 1)$, let $\epsilon' = \frac{\ell}{t'}\left(d \ln \frac{2et'}{d} + \ln \frac{4\ell}{\delta'}\right)$. By Lemma 3.3, for any value of $h_{\mathbf{w}}(\mathbf{x}) \in \mathcal{Y}$, with probability at least $1 - \delta'$,

$$\left|\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))] - A(h_{\mathbf{w}}(\mathbf{x}_t))\right| \leq 2\epsilon' H.$$

Setting $\delta' = \delta/|\mathcal{Y}| = \delta/2^\ell$, by a union bound over elements in $\mathcal{Y}$, we have that with probability at least $1 - \delta$,

$$\mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[(v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s^* - (v^*(\mathbf{x}_t) - p(\mathbf{x}_t))\, s_t\right]$$
$$\leq \mathop{\mathbb{E}}_{\mathbf{x}_t\sim\mathcal{P}}\left[\left|\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right] - p(\mathbf{x}_t)\right| \,\Big|\, E\right]$$
$$\leq \mathop{\mathbb{E}}_{\mathbf{x}_t}\left[\left|\mathop{\mathbb{E}}_{\mathbf{x}}\left[v^*(\mathbf{x}) \mid \mathbf{x} \in h_{\mathbf{w}}^{-1}(h_{\mathbf{w}}(\mathbf{x}_t))\right] - A(h_{\mathbf{w}}(\mathbf{x}_t))\right| \,\Big|\, E\right]$$
$$\leq \frac{2H\ell}{t'}\left(d \ln \frac{2et'}{d} + \ln \frac{\ell \cdot 2^{\ell+2}}{\delta}\right),$$

which completes the proof. □

This instantaneous regret bound then implies a regret bound for Algorithm 1, the proof of which is in Appendix B.

Theorem 3.5. *With probability at least $1 - \delta$, the regret of Algorithm 1 is $R_T = \mathcal{O}(\sqrt{Td\ell \log(T\ell/\delta)})$.*

In Algorithm 1, we assumed that we knew the horizon T. This assumption can be lifted via a doubling trick; see Appendix B.

4. General Masking Functions

We next consider a more general setting where in each round, an item $\mathbf{x}_t$ is drawn from an unknown distribution $\mathcal{P}$ and a published price p_t is drawn from some fixed unknown distribution. We study the case where the masking function is an arbitrary mapping $h : \mathcal{X} \to [n]$. In this setting, is there a no-regret strategy that the buyer can use?

We answer this question in the affirmative, building on the classical `Exp4.VC` algorithm (Beygelzimer et al., 2011). Out of the box, `Exp4.VC` has per-round runtime that is exponential in n, but we exploit the structure of our problem setting to obtain a polynomial per-round runtime. We prove that this algorithm has a regret bound of $\mathcal{O}(\sqrt{T(n \log T/n + \log 1/\delta)})$ with probability at least $1 - \delta$.

4.1. `Exp4.VC` algorithm

Based on the optimal strategy from Proposition 2.3, we define an infinite set of policies that take as input $(p_t, h(\mathbf{x}_t))$ and return decisions in $\{0, 1\}$ indicating whether or not the buyer should buy the item. Each policy is defined by a vector $\mathbf{v} \in [0, H]^n$ as follows: $\pi_{\mathbf{v}}(p_t, h(\mathbf{x}_t)) = \mathbb{I}(\mathbf{v}[h(\mathbf{x}_t)] \geq p_t)$. The optimal strategy from Proposition 2.3 corresponds to the strategy $\pi_{\mathbf{v}}$ with $\mathbf{v} = (\mathbb{E}[v^*(\mathbf{x}) \mid h(\mathbf{x}) = 1], \ldots, \mathbb{E}[v^*(\mathbf{x}) \mid h(\mathbf{x}) = n])$. We use the notation $\Pi = \{\pi_{\mathbf{v}} \mid \mathbf{v} \in [0, H]^n\}$ to denote the set of all such policies.

A key observation is that our problem can be framed as a contextual bandit problem with an oblivious adversary and an infinite set of contexts. At each round $t = 1, \ldots, T$, the buyer observes stochastic context $(p_t, h(\mathbf{x}_t))$ and makes

Algorithm 2 `Exp4.VC` with an unknown distribution

Input: $T \geq 0$, $\delta \in (0,1)$.
Set $\tau = \sqrt{Tn \log \frac{eT}{n} + \log \frac{2}{\delta}}$.
for $t = 1, 2, \ldots, \tau$ **do** {Initialization phase}
  Receive $h(\mathbf{x}_t)$, price p_t where $\mathbf{x}_t \overset{iid}{\sim} \mathcal{P}$.
  Make decision $b_t \sim Bern(0.5)$ at random.
end for
for $i = 1, 2, \ldots, n$ **do** {Extract a finite subset of policies}
  Let $i_1, i_2, \ldots, i_{m_i} \in [\tau]$ be the set of indices where $h(\mathbf{x}_{i_j}) = i$ for all $j \in \{1, 2, \ldots, m_i\}$
  Define $\mathcal{V}_i = \{0, p_{i_1}, p_{i_2}, \ldots, p_{i_{m_i}}\}$
end for
Define $\mathcal{V} = \times_{i=1}^{n} \mathcal{V}_i$
Set $\gamma = \sqrt{\frac{\log |\mathcal{V}|}{2(T-\tau)}}$ and $w_{\tau+1}^{\mathbf{v}} = 1$ for all $\mathbf{v} \in \mathcal{V}$.
for $t = \tau + 1, \ldots, T$ **do** {`Exp.4` subroutine}
  Receive $h(\mathbf{x}_t)$, price p_t where $\mathbf{x}_t \overset{iid}{\sim} \mathcal{P}$.
  Get advice vectors $\boldsymbol{\xi}_t^{\mathbf{v}} \in \{0,1\}^2$ for all $\mathbf{v} \in \mathcal{V}$ where $\boldsymbol{\xi}_t^{\mathbf{v}} = (1 - \pi_{\mathbf{v}}(p_t, h(\mathbf{x}_t)), \pi_{\mathbf{v}}(p_t, h(\mathbf{x}_t)))$.
  Set $W_t = \sum_{\mathbf{v} \in \mathcal{V}} w_t^{\mathbf{v}}$ and define $\bar{\boldsymbol{\xi}}_t \in (0,1)^2$ as

$$\bar{\boldsymbol{\xi}}_t[0] = (1 - 2\gamma) \sum_{\mathbf{v} \in \mathcal{V}} \frac{w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[0]}{W_t} + \gamma$$

$$\bar{\boldsymbol{\xi}}_t[1] = (1 - 2\gamma) \sum_{\mathbf{v} \in \mathcal{V}} \frac{w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[1]}{W_t} + \gamma.$$

  Draw decision $b_t \sim Bern(\bar{\boldsymbol{\xi}}_t[1])$ and receive reward $u_t = (v^*(\mathbf{x}_t) - p_t) b_t$.
  Set $\hat{\mathbf{r}}_t = ({}^{b_t u_t}/_{\bar{\boldsymbol{\xi}}_t[1]}, 0)^\top$.
  for $\mathbf{v} \in \mathcal{V}$ **do**
    Set

$$C_t^{\mathbf{v}} = \frac{\gamma}{2} \left(\boldsymbol{\xi}_t^{\mathbf{v}} \cdot \hat{\mathbf{r}}_t + \sum_{b=0}^{1} \frac{\boldsymbol{\xi}_t^{\mathbf{v}}[b]}{\bar{\boldsymbol{\xi}}_t^{\mathbf{v}}[b]} \sqrt{\frac{\log |\mathcal{V}|/\delta}{2(T-\tau)}} \right)$$

    Set $w_{t+1}^{\mathbf{v}} = w_t^{\mathbf{v}} \exp C_t^{\mathbf{v}}$
  end for
end for

a purchase decision. This is a contextual bandit problem with two arms: the first arm corresponds to the decision "no purchase" and the second arm corresponds to the decision "purchase." The reward of the first arm is always zero, while the reward of pulling the second arm depends on the item $\mathbf{x}_t$ and the price p_t.

We prove that the class of policies Π has a VC dimension of n, which allows us to adapt `Exp4.VC` from Beygelzimer et al. (2011), a generic contextual bandits algorithm for policy classes with finite VC dimension. Algorithm 2 begins with an initialization phase of length τ. In this phase, the buyer chooses their action uniformly at random and collects tuples $\{(h(\mathbf{x}_t), p_t)\}_{t=1}^{\tau}$. The algorithm then uses these tuples to identify a finite (though exponentially large), representative subset of policies in Π. In particular, using the collected tuples, the algorithm partitions Π into a *finite* set of equivalence classes where two policies π, π' are equivalent if they agree on the set of τ collected tuples. Then the buyer constructs a finite set of policies Π' by selecting one policy from each equivalence class. Algorithm 2 does this by first defining for each $i \in [n]$ a set $\mathcal{V}_i \subset \mathbb{R}$ which is the set of all prices from the first τ rounds for items $\mathbf{x}_t$ with $h(\mathbf{x}_t) = i$. The finite set Π' of policies is then defined as $\Pi' = \{\pi_{\mathbf{v}} : \mathbf{v} \in \times_{i=1}^{n} \mathcal{V}_i\}$. We prove that Π' contains a policy from each equivalence class in Lemma 4.2.

In the remaining rounds, Algorithm 2 follows the `Exp4.P` strategy (Beygelzimer et al., 2011) which runs multiplicative weight updates on each of the selected policies $\pi_{\mathbf{v}}$ with $\mathbf{v} \in \times_{i=1}^{n} \mathcal{V}_i$. Out-of-the-box, `Exp4.VC` would therefore have a per-round runtime that is exponential in n since $\times_{i=1}^{n} \mathcal{V}_i$ is exponentially large. However, with a careful analysis, we show that in our setting these multiplicative weight updates can be computed in polynomial time.

4.2. Regret

The key first step is to show that although the set of all policies we need to consider Π is infinite, it has a *finite* VC dimension. The full proof is in Appendix C.

Lemma 4.1. *The VC dimension of* Π *is* n.

Proof sketch. First, we show that the functions in Π cannot be used to label $n+1$ contexts in all possible ways. Given $n+1$ contexts $(h(\mathbf{x}_1), p_1), \ldots, (h(\mathbf{x}_{n+1}), p_{n+1})$, by the pigeonhole principle there must exist at least two items $\mathbf{x}_i$ and $\mathbf{x}_j$ that have the same index: $h(\mathbf{x}_i) = h(\mathbf{x}_j)$. Therefore, for any policy $\pi_{\mathbf{v}}$, the decisions for these two items are determined by the same threshold $\mathbf{v}[h(\mathbf{x}_i)] = \mathbf{v}[h(\mathbf{x}_j)]$. Without loss of generality, assume that $p_i < p_j$. There is no policy $\pi_{\mathbf{v}}$ where the decision is to purchase item j but not purchase item i because this would imply that $p_j \leq \mathbf{v}[h(\mathbf{x}_j)] = \mathbf{v}[h(\mathbf{x}_i)] < p_i$. However, with fewer than $n+1$ items, since all items can use a different threshold, their decisions do not interfere with each other. □

Next, in order to invoke the regret bound of `Exp4.VC`, we verify that Π' is a representative set of policies. Formally, suppose we partition Π into a set of equivalence classes where policies π and π' are equivalent if they agree on the set of τ tuples collected in the initalization phase. We prove that Π' contains a policy from each equivalence class.

Lemma 4.2. *Let $\mathcal{V}$ be defined as in Algorithm 2. Then*

$$\{(\pi_{\mathbf{v}}(h(\mathbf{x}_1), p_1), \ldots, \pi_{\mathbf{v}}(h(\mathbf{x}_\tau), p_\tau)) : \mathbf{v} \in [0, H]^n\}$$
$$= \{(\pi_{\mathbf{v}}(h(\mathbf{x}_1), p_1), \ldots, \pi_{\mathbf{v}}(h(\mathbf{x}_\tau), p_\tau)) : \mathbf{v} \in \mathcal{V}\}.$$

The proof of this lemma can be found in Appendix C.

Lemmas 4.1 and 4.2 imply the following regret bound:[3]

Theorem 4.3. *With probability $1-\delta$, Algorithm 2 achieves a regret rate that is $R_T = \mathcal{O}(\sqrt{T(n \log T/n + \log 1/\delta)})$.*

Proof. This theorem follows directly from Lemmas 4.1 and 4.2 and Theorem 5 of Beygelzimer et al. (2011). □

4.3. Computational Complexity

The key challenge in applying `Exp4.VC` out-of-the-box is that it computes multiplicative weight updates over every policy $\pi_{\mathbf{v}}$ with $\mathbf{v} \in \mathcal{V}$—an exponential number of policies. We show that by exploiting our problem structure, we can perform these multiplicative weight updates in each round in polynomial time. In particular, we show that we can efficiently compute the purchase probabilities $\bar{\boldsymbol{\xi}}_t[0]$ and $\bar{\boldsymbol{\xi}}_t[1]$ without computing the multiplicative weights $w_t^{\mathbf{v}}$ for each $\mathbf{v} \in \mathcal{V}$ explicitly, and therefore Algorithm 2 can be run with polynomial per-round runtime. Intuitively, rather than sum over every vector in $\mathcal{V} = \times_{i=1}^n \mathcal{V}_i$ as in the definitions of $\bar{\boldsymbol{\xi}}_t[0]$ and $\bar{\boldsymbol{\xi}}_t[1]$, we show how to sum over individual elements in $\cup_{i=1}^n \mathcal{V}_i$, of which there are $\tau + n = \tilde{\mathcal{O}}(\sqrt{Tn} + n)$. We provide the complete proof in Appendix C.

Theorem 4.4. *The purchase probabilities $\bar{\boldsymbol{\xi}}_t[0]$ and $\bar{\boldsymbol{\xi}}_t[1]$ in Algorithm 2 can be computed in $\mathcal{O}(n+\tau) = \mathcal{O}(n + \sqrt{Tn \log(T/n) + \log(1/\delta)})$ time.*

Proof sketch. Our proof begins with the observation that for each index $i \in [n]$, the thresholds $0 \le p_{i_1} \le \cdots \le p_{i_{m_i}}$ in $\mathcal{V}_i$ divide the price range $[0, H]$ into $m_i + 1$ non-overlapping "buckets": $[0, p_{i_1}), [p_{i_2}, p_{i_3}), \ldots, [p_{i_{m_i}}, H]$. Using the notation $m = \max_i m_i$, the total number of buckets is $\mathcal{O}(mn)$. Each context $(h(\mathbf{x}_t), p_t)$ corresponds to exactly one bucket: the bucket $[p_{i_j}, p_{i_{j+1}})$ containing p_t where $h(\mathbf{x}_t) = i$. Moreover, the decision of each policy in each bucket is constant since the policies all use the same thresholds, namely, the boundaries of these buckets. Given a vector $\mathbf{v} \in \mathcal{V}$ and a bucket k, let $a_k^{\mathbf{v}} \in \{0, 1\}$ be the policy's recommendation of "buy" or "do not buy" for any item that falls in that bucket. This allows us to rewrite the policy decision $\boldsymbol{\xi}_t^{\mathbf{v}}$ as an alternative sum: $\boldsymbol{\xi}_t^{\mathbf{v}} = \left(\sum_{k:a_k^{\mathbf{v}}=0} \mathbb{I}(\text{item in } k), \sum_{k:a_k^{\mathbf{v}}=1} \mathbb{I}(\text{item in } k)\right)^\top$. Intuitively, $\mathbb{I}(\text{item in } k)$ is only nonzero for the bucket that this item belongs to, and the policy's decision for the item is the same as the policy's decision for that bucket. Using this fine-grained argument, we then show that all the purchase probabilities $\bar{\boldsymbol{\xi}}_t[0]$ and $\bar{\boldsymbol{\xi}}_t[1]$ can be computed in polynomial time without explicitly computing the exponentially many weights $w_t^{\mathbf{v}}$. □

[3] By running n copies of `Exp4.VC` in parallel for each context, we would obtain a regret bound of $\mathcal{O}(\sqrt{Tn \log(Tn/\delta)})$, but by using a more careful analysis in this section, we improve the dependence on n.

Remark 4.5. Under *adversarial* prices, we can run n independent copies of an algorithm for Lipschitz contextual bandits (for example, the algorithm from Section 8.3 of the textbook by Slivkins et al. (2019)) to obtain an expected regret bound of $\tilde{O}(T^{2/3} n^{1/3})$.

5. Conclusion

We presented learning algorithms for buyers who participate in auctions with limited item information. This model captures a broad set of practical applications, including advertising auctions. Our algorithms are no-regret with respect to an oracle buyer who has perfect knowledge of the distribution over items and the masking function that the seller uses to obfuscate the item information. We proposed no-regret learning algorithms in a variety of settings, including when the distribution over items is either known or unknown to the buyer, and when the prices are either stochastic or adversarial.

To the best of our knowledge, this is the first result on no-regret learning strategies for buyers with partial item information. Many interesting questions remain open for future research. First, we have assumed that the valuation is bounded, and it would be an interesting direction for future research to extend our results to unbounded distributions. Second, we focused on posted-prices in this work. It would be interesting to consider extensions to second-price auctions with partial item information. With stochastic prices, the problem seems potentially feasible, but adversarial prices might pose challenges because the adversary could block the learner from estimating the second-highest bid. Moreover, can the platform release partial information in a way that optimally trades off between revenue and privacy? When the distribution over items is unknown, our algorithm works with general masking functions. Can better purchasing strategies be developed by exploiting the properties of a specific set of masking functions?

Acknowledgements

We wish to acknowledge support from the Vannevar Bush Faculty Fellowship program under grant number N00014-21-1-2941. WG acknowledges support from a Google PhD fellowship.

References

Anthony, M. and Bartlett, P. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

Arefeva, A. and Meng, D. Revealing information in auctions: The optimal auction versus the second-price auction. *Economics Letters*, 204:109895, 2021.

Arnosti, N., Beck, M., and Milgrom, P. Adverse selection and auction design for internet display advertising. *American Economic Review*, 106(10):2852–66, 2016.

Bagnoli, M. and Bergstrom, T. Log-concave probability and its applications. In *Rationality and Equilibrium*, pp. 217–241. Springer, 2006.

Balseiro, S. R. and Gur, Y. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019.

Bergemann, D. and Pesendorfer, M. Information structures in optimal auctions. *Journal of Economic Theory*, 137(1): 580–609, 2007.

Bergemann, D., Heumann, T., and Morris, S. Selling impressions: Efficiency vs. competition. 2021.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26. JMLR Workshop and Conference Proceedings, 2011.

De Corniere, A. and De Nijs, R. Online advertising and privacy. *The RAND Journal of Economics*, 47(1):48–72, 2016.

Dikkala, N. and Tardos, É. Can credit increase revenue? In *International Conference on Web and Internet Economics*, pp. 121–133. Springer, 2013.

Edelman, B., Ostrovsky, M., and Schwarz, M. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1):242–259, 2007.

Epasto, A., Muñoz Medina, A., Avery, S., Bai, Y., Busa-Fekete, R., Carey, C., Gao, Y., Guthrie, D., Ghosh, S., Ioannidis, J., et al. Clustering for private interest-based advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2802–2810, 2021.

Eső, P. and Szentes, B. Optimal information disclosure in auctions and the handicap auction. *The Review of Economic Studies*, 74(3):705–731, 2007.

Feng, Z., Podimata, C., and Syrgkanis, V. Learning to bid without knowing your value. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 505–522, 2018.

Ganuza, J.-J. Ignorance promotes competition: an auction model with endogenous private valuations. *RAND Journal of Economics*, pp. 583–598, 2004.

Gershkov, A. Optimal auctions and information disclosure. *Review of Economic Design*, 13(4):335–344, 2009.

Guha, S., Cheng, B., and Francis, P. Privad: Practical privacy in online advertising. In *USENIX Conference on Networked Systems Design and Implementation*, pp. 169–182, 2011.

Juels, A. Targeted advertising... and privacy too. In *Cryptographers' Track at the RSA Conference*, pp. 408–424. Springer, 2001.

Li, D. and Jewitt, I. Cheap talk advertising in auctions: horizontally vs vertically differentiated products. 2017.

Li, H. and Shi, X. Discriminatory information disclosure. *American Economic Review*, 107(11):3363–85, 2017.

Lovász, L. and Vempala, S. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 57–68. IEEE, 2006.

Milgrom, P. *Putting Auction Theory to Work*. Cambridge University Press, 2004.

Milgrom, P. Simplified mechanisms with an application to sponsored-search auctions. *Games and Economic Behavior*, 70(1):62–70, 2010.

Post, D., Coppinger, S., and Sheble, G. Application of auctions as a pricing mechanism for the interchange of electric power. *IEEE Transactions on Power Systems*, 10 (3):1580–1584, 1995.

Rafieian, O. and Yoganarasimhan, H. Targeting and privacy in mobile advertising. *Marketing Science*, 40(2):193–218, 2021.

Roesler, A.-K. and Szentes, B. Buyer-optimal learning and monopoly pricing. *American Economic Review*, 107(7): 2072–80, 2017.

Saumard, A. and Wellner, J. A. Log-concavity and strong log-concavity: a review. *Statistics surveys*, 8:45, 2014.

Slivkins, A. et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Weed, J., Perchet, V., and Rigollet, P. Online learning in repeated auctions. In *Conference on Learning Theory*, pp. 1562–1583. PMLR, 2016.

A. Proofs for Section 2

Proposition 2.3. *The strategy s^* that maximizes $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[(v^*(\mathbf{x}) - p(\mathbf{x}))s^*(h(\mathbf{x}), p(\mathbf{x}), h, \mathcal{P}) \mid h(\mathbf{x})]$ is*

$$s^*(h(\mathbf{x}), p(\mathbf{x}), h, \mathcal{P}) = \mathbb{I}\left(\mathop{\mathbb{E}}_{x\sim\mathcal{P}}[v^*(\mathbf{x}) \mid h(\mathbf{x})] > p(\mathbf{x})\right).$$

Proof. For a decision $b \in \{0, 1\}$, let R_b denote the expected utility of buying or not buying an item given the published information and price:

$$\begin{aligned}
R_b &\stackrel{\text{def}}{=} \mathbb{E}[(v^*(\mathbf{x}) - p(\mathbf{x})) \cdot b \mid h(\mathbf{x}), p(\mathbf{x})] \\
&= \mathbb{E}[(v^*(\mathbf{x}) - p(\mathbf{x})) \cdot b \mid v^*(\mathbf{x}) \geq p(\mathbf{x}), h(\mathbf{x}), p(\mathbf{x})] \cdot \mathbb{P}[v^*(\mathbf{x}) \geq p(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})] \\
&\quad - \mathbb{E}[(p(\mathbf{x}) - v^*(\mathbf{x})) \cdot b \mid v^*(\mathbf{x}) < p(\mathbf{x}), h(\mathbf{x}), p(\mathbf{x})] \cdot \mathbb{P}[v^*(\mathbf{x}) < p(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})].
\end{aligned}$$

Then, by definition, the optimal strategy $s^*(h(\mathbf{x}), p(\mathbf{x}), p, \mathcal{P}) = \mathbb{I}(R_1 > R_0) = \mathbb{I}(R_1 > 0)$.

Further, letting $x^+ = \max\{0, x\}$, by the law of total expectation,

$$\begin{aligned}
&\mathbb{E}[(v^*(\mathbf{x}) - p(\mathbf{x}))^+ \mid h(\mathbf{x}), p(\mathbf{x})] \\
&= \mathbb{E}[(v^*(\mathbf{x}) - p(\mathbf{x})) \mid v^*(\mathbf{x}) \geq p(\mathbf{x}), h(\mathbf{x}), p(\mathbf{x})] \cdot \mathbb{P}[v^*(\mathbf{x}) \geq p(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})] \\
&\quad + \mathbb{E}[0 \mid v^*(\mathbf{x}) < p(\mathbf{x}), h(\mathbf{x}), p(\mathbf{x})] \cdot \mathbb{P}(v^*(\mathbf{x}) < p(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})) \\
&= \mathbb{E}[(v^*(\mathbf{x}) - p(\mathbf{x})) \mid v^*(\mathbf{x}) \geq p(\mathbf{x}), h(\mathbf{x}), p(\mathbf{x})] \cdot \mathbb{P}[v^*(\mathbf{x}) \geq p(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})].
\end{aligned}$$

Similarly, letting $x^- = -\min\{0, x\}$,

$$\begin{aligned}
&\mathbb{E}[(v^*(\mathbf{x}) - p(\mathbf{x}))^- \mid h(\mathbf{x}), p(\mathbf{x})] \\
&= \mathbb{E}[(p(\mathbf{x}) - v^*(\mathbf{x})) \mid v^*(\mathbf{x}) < p(\mathbf{x}), h(\mathbf{x}), p(\mathbf{x})] \cdot \mathbb{P}(v^*(\mathbf{x}) < p(\mathbf{x}) \mid h(\mathbf{x}), p(\mathbf{x})).
\end{aligned}$$

Therefore, the optimal strategy is:

$$\begin{aligned}
s^*(h(\mathbf{x}), p(\mathbf{x}), h, p, \mathcal{P}) &= \mathbb{I}\left(\mathbb{E}[(v^*(x) - p(\mathbf{x}))^+ \mid h(\mathbf{x}), p(\mathbf{x})] - \mathbb{E}[(v^*(x) - p(\mathbf{x}))^- \mid h(\mathbf{x}), p(\mathbf{x})] > 0\right) \\
&= \mathbb{I}\left(\mathbb{E}[(v^*(x) - p(\mathbf{x})) \mid h(\mathbf{x}), p(\mathbf{x})] > 0\right) \\
&= \mathbb{I}\left(\mathbb{E}[v^*(x) \mid h(\mathbf{x}), p(\mathbf{x})] > p(\mathbf{x})\right).
\end{aligned}$$

The lemma statement then follows from Equation (1). □

B. Proofs for Section 3

Definition B.1 (Lovász & Vempala (2006))**.** Define the centroid of f_π as $c_f = \int \mathbf{x} df_\pi$. Define the variance of f_π as $\mathrm{var}(f_\pi) = \int \|\mathbf{x} - c_f\|^2 df_\pi$. Also denote $\mathcal{L}(\theta)$ as the level set $\{\mathbf{x} : v^*(\mathbf{x})\pi(\mathbf{x}) \geq \theta\}$.

Theorem 3.5. *With probability at least $1 - \delta$, the regret of Algorithm 1 is $R_T = \mathcal{O}(\sqrt{Td\ell \log(T\ell/\delta)})$.*

Proof. First, in the exploration phase, the total regret is upper bounded by Ht'.

Now consider the regret in the exploitation phase. Notice that, we would have obtained $\{\mathbf{x}_i, h_{\mathbf{w}}(\mathbf{x}_i)\}_{i=1}^{t'}$ number of i.i.d. samples from the exploration phase.

Let $\epsilon = \frac{\ell}{t'}\left(d \ln \frac{2et'}{d} + \ln \frac{\ell \cdot 2^{\ell+2}}{\delta'}\right)$. By Lemma 3.4 and a union bound over all rounds, with probability at least $1 - \delta' T$, the total expected regret incurred by Algorithm 1 is $R_T \leq Ht' + 2\epsilon H(T - t')$.

Let $t' = \sqrt{Td\ell \log\left(\frac{2\ell}{\delta'}\right)}$, we have

$$R_T \leq H\sqrt{Td\ell \log\left(\frac{2\ell}{\delta'}\right)} + 2H\ell d\sqrt{\frac{T}{d\ell \log\left(\frac{2\ell}{\delta'}\right)}} \log\left(\frac{2\ell}{d}\sqrt{Td\ell \log\left(\frac{2\ell}{\delta'}\right)}\right) + 2H\ell\sqrt{\frac{T}{d\ell \log\left(\frac{2\ell}{\delta'}\right)}} \log\left(\frac{\ell 2^{\ell+2}}{\delta'}\right).$$

Note that $\log\left(\frac{\ell 2^{\ell+2}}{\delta'}\right) = \log\left(\frac{\ell}{\delta'}\right) + \ell\log(4) \leq \log\left(\frac{\ell}{\delta'}\right) + d\log(4)$. Setting $\delta' = \delta/T$, we have that with probability at least $1-\delta$,

$$R_T = \tilde{\mathcal{O}}\left(\sqrt{Td\ell\log\left(\frac{2\ell T}{\delta}\right)}\right). \tag{3}$$

This completes the proof. □

B.1. Algorithm for an unknown horizon T

In Theorem 3.5, we assumed that we knew the time horizon T, which allowed us to set the correct length for the exploration phase. This assumption can be lifted by using the doubling trick which runs the algorithm in independent intervals that are doubling in length, as summarized by Algorithm 3. The regret bound remains the same up to constant factors.

Algorithm 3 Explore-then-Commit with Unknown Horizon

1: **Input:** starting epoch length T_0.
2: **for** $i = 1, 2, \ldots$ **do**
3: $\quad T_i \leftarrow 2^i T_0$.
4: $\quad$ Run Algorithm 1 with $T = T_i$.
5: **end for**

Corollary B.2. *Suppose that $v^*(\mathbf{x})\pi(\mathbf{x})$ is logconcave and well-rounded. Then with probability at least $1-\delta$, the regret of Algorithm 3 is $R_T = \tilde{\mathcal{O}}(\sqrt{Td\ell\log(T\ell/\delta)})$.*

Proof. Denote the total expected accumulated regret as R_T, and the expected accumulated regret in each interval with length T_i as R_{T_i}. Denote the number of intervals as $m \leq \log_2 \frac{2T}{T_0}$. Then, by Theorem 3.5 we have that for some universal constant C and with probability at least $1-\delta' m$:

$$\begin{aligned}
R_T &\leq \sum_{i=1}^{m} R_{T_i} \\
&\leq \sum_{i=1}^{m} C\sqrt{T_i d\ell\log\left(\frac{2\ell T_i}{\delta'}\right)} \\
&\leq C\sqrt{d\ell}\sum_{i=1}^{m}\sqrt{T_i\left(\log\left(\frac{2\ell}{\delta'}\right) + \log T_i\right)} \\
&\leq C\sqrt{d\ell}\left(\sqrt{\log\left(\frac{2\ell}{\delta'}\right)}\cdot\sum_{i=1}^{m}\sqrt{2^i T_0} + \sum_{i=1}^{m}\sqrt{2^i T_0\log(2^i T_0)}\right) \\
&= \mathcal{O}\left(\sqrt{d\ell T\log\left(\frac{2\ell}{\delta'}\right)}\right).
\end{aligned}$$

Let $\delta' = \delta/m$ and note that $m \leq \log_2 \frac{2T}{T_0}$, applying a union bound over all intervals, we have that with probability at least $1-\delta$,

$$R_T = \tilde{\mathcal{O}}\left(\sqrt{d\ell T\log\left(\frac{2\ell}{\delta}\right)}\right).$$

This completes the proof. □

C. Proofs for Section 4

Lemma 4.1. *The VC dimension of Π is n.*

Proof. We argue that any function contained in Π cannot be used to label $n+1$ input points in all possible ways. For simplicity denote $h(\mathbf{x}) = y \in [n]$. Consider a set of input points $\{y_i, p_i\}_{i=1}^{n+1}$. By the pigeonhole principle there must exist at least two elements $(y_i, p_i), (y_j, p_j)$ such that $y_i = y_j$ and $p_i \neq p_j$. Therefore by definition, we have that for any $\mathbf{v} \in \mathbb{R}^n$,

$$\pi_{\mathbf{v}}(y_i, p_i) = \mathbb{I}(\mathbf{v}[y_i] > p_i) = \mathbb{I}(\mathbf{v}[y_j] > p_i),$$
$$\pi_{\mathbf{v}}(y_j, p_j) = \mathbb{I}(\mathbf{v}[y_j] > p_j) = \mathbb{I}(\mathbf{v}[y_i] > p_j).$$

Without loss of generality, assume that $p_i < p_j$. Then the pair of labels $\pi_{\mathbf{v}}(y_i, p_i) = 1$ and $\pi_{\mathbf{v}}(y_j, p_j) = 0$ can never be achieved for any $\mathbf{v} \in \mathbb{R}^n$. Thus $VCdim(\Pi) < n+1$.

Next, consider a set of n input points $\{(i, 0.5)\}_{i=1}^{n}$. Each point in this set is labeled using a different index i, so all possible combinations of labels can be achieved using vectors $\mathbf{v} \in \{0,1\}^n$. Thus we conclude that $VCdim(\Pi) = n$.

□

Lemma 4.2. *Let $\{(h(\mathbf{x}_1), p_1), \ldots, (h(\mathbf{x}_\tau), p_\tau)\}$ be a subset of $[n] \times [0, H]$. For each $i \in [n]$, let $i_1, i_2, \ldots, i_{m_i} \in [\tau]$ be the set of indices where $h(\mathbf{x}_{i_j}) = i$ for all $j \in \{1, 2, \ldots, m_i\}$. Define $\mathcal{V}_i = \{0, p_{i_1}, p_{i_2}, \ldots, p_{i_{m_i}}\}$ and $\mathcal{V} = \times_{i=1}^{n} \mathcal{V}_i$. Then*

$$\left\{ \begin{pmatrix} \pi_{\mathbf{v}}(h(\mathbf{x}_1), p_1) \\ \vdots \\ \pi_{\mathbf{v}}(h(\mathbf{x}_\tau), p_\tau) \end{pmatrix} \right\}_{\mathbf{v} \in [0,H]^n} = \left\{ \begin{pmatrix} \pi_{\mathbf{v}}(h(\mathbf{x}_1), p_1) \\ \vdots \\ \pi_{\mathbf{v}}(h(\mathbf{x}_\tau), p_\tau) \end{pmatrix} \right\}_{\mathbf{v} \in \mathcal{V}}.$$

Proof. We will show that for every $\mathbf{v} \in [0, H]^n$, there exists a vector $\mathbf{v}_0 \in \times_{i=1}^{n} \mathcal{V}_i$ such that $\pi_{\mathbf{v}}(h(\mathbf{x}_j), p_j) = \pi_{\mathbf{v}_0}(h(\mathbf{x}_j), p_j)$ for every $j \in [\tau]$. To this end, fix an index $i \in [n]$ and without loss of generality, let $i_1, i_2, \ldots, i_{m_i}$ be sorted such that $0 := p_{i_0} < p_{i_1} < p_{i_2} < \cdots < p_{i_{m_i}}$. Let $i' \in \{0, 1, 2, \ldots, i_{m_i}\}$ be the largest index such that $\mathbf{v}[i] \geq p_{i'}$. Define $\mathbf{v}_0[i] = p_{i'}$. For every index $i_j \leq i'$, we know that $\mathbf{v}[i] \geq p_{i'} = \mathbf{v}_0[i] \geq p_{i_j}$, so $\pi_{\mathbf{v}}(h(\mathbf{x}_{i_j}), p_{i_j}) = \mathbb{I}(\mathbf{v}[i] \geq p_{i_j}) = 1 = \mathbb{I}(\mathbf{v}_0[i] \geq p_{i_j}) = \pi_{\mathbf{v}_0}(h(\mathbf{x}_{i_j}), p_{i_j})$. Meanwhile, for every index $i_j > i'$, we know that $p_{i'} = \mathbf{v}_0[i] \leq \mathbf{v}[i] < p_{i_j}$. Therefore, $\pi_{\mathbf{v}}(h(\mathbf{x}_{i_j}), p_{i_j}) = 0 = \pi_{\mathbf{v}_0}(h(\mathbf{x}_{i_j}), p_{i_j})$. In either case, we have that $\pi_{\mathbf{v}}(h(\mathbf{x}_{i_j}), p_{i_j}) = \pi_{\mathbf{v}_0}(h(\mathbf{x}_{i_j}), p_{i_j})$. Since this is true for every index $i \in [n]$, the lemma statement holds. □

Theorem 4.4. *The purchase probabilities $\bar{\boldsymbol{\xi}}_t[0]$ and $\bar{\boldsymbol{\xi}}_t[1]$ in Algorithm 2 can be computed in $\mathcal{O}(n+\tau) = \mathcal{O}(n + \sqrt{Tn\log(T/n) + \log(1/\delta)})$ time.*

Proof. Label the elements of $\mathcal{V}_i$ as $0 := p_{i,0} < p_{i,1} < \cdots < p_{i,m_i}$. Let $m = \max m_i$. For the ease of notation, define the variables $p_{i,m_i+1} = p_{i,m_i+2} = \cdots p_{i,m} = H$. For each $i \in [n]$, $j \in \{1, \ldots, m\}$, and $t \in \{\tau+1, \tau_2, \ldots, T\}$, we define the variable

$$d_{i,j}(t) = \mathbb{I}(h(\mathbf{x}_t) = i \text{ and } p_t \in (p_{i,j-1}, p_{i,j}]).$$

We also set $d_{i,0}(t) = \mathbb{I}(h(\mathbf{x}_t) = i \text{ and } p_t = 0)$. Next, for each $\mathbf{v} \in \times_{i=1}^{n} \mathcal{V}_i$, $i \in [n]$, and $j \in \{0, \ldots, m\}$, we define the variable $a_{i,j}^{\mathbf{v}} = \mathbb{I}(\mathbf{v}[i] \geq p_{i,j})$.

Claim C.1. *For $b \in \{0, 1\}$,*

$$\boldsymbol{\xi}_t^{\mathbf{v}}[b] = \sum_{(i,j): a_{i,j}^{\mathbf{v}} = b} d_{i,j}(t). \tag{4}$$

Proof of Claim C.1. First, let $i_t = h(\mathbf{x}_t)$. If $p_t = 0$, define $j_t = 0$ and otherwise, define j_t such that $p_t \in (p_{i_t, j_t - 1}, p_{i_t, j_t}]$. Therefore, $d_{i,j}(t) = 1$ if $(i,j) = (i_t, j_t)$ and $d_{i,j}(t) = 0$ otherwise. This means that

$$\sum_{(i,j): a_{i,j}^{\mathbf{v}} = b} d_{i,j}(t) = \mathbb{I}(a_{i_t, j_t}^{\mathbf{v}} = b). \tag{5}$$

We split the proof into two cases: $b = 0$ and $b = 1$.

Case 1: $b = 0$**.** We know that $\boldsymbol{\xi}_t^{\mathbf{v}}[0] = 1 - \pi_{\mathbf{v}}(p_t, h(\mathbf{x}_t)) = \mathbb{I}(\mathbf{v}[h(\mathbf{x}_t)] < p_t) = \mathbb{I}(\mathbf{v}[i_t] < p_t)$.

If $\boldsymbol{\xi}_t^{\mathbf{v}}[0] = 0$, then $\mathbf{v}[i_t] \geq p_t$. Since $\mathbf{v} \in \mathcal{V}$, we know that $\mathbf{v}[i_t] = p_{i_t,j}$ for some $j \in [m]$. Moreover, since $p_t \in (p_{i_t,j_t-1}, p_{i_t,j_t}]$, the fact that $\mathbf{v}[i_t] \geq p_t$ means that $\mathbf{v}[i_t] \geq p_{i_t,j_t}$. Therefore, $a_{i_t,j_t}^{\mathbf{v}} = 1$. By Equation (5), this means that $\sum_{(i,j):a_{i,j}^{\mathbf{v}}=0} d_{i,j}(t) = 0$, so Equation (4) holds.

Meanwhile, if $\boldsymbol{\xi}_t^{\mathbf{v}}[0] = 1$, then $\mathbf{v}[i_t] < p_t \leq p_{i_t,j_t}$. Therefore, $a_{i_t,j_t}^{\mathbf{v}} = 0$. By Equation (5), this means that $\sum_{(i,j):a_{i,j}^{\mathbf{v}}=0} d_{i,j}(t) = 1$, so Equation (4) holds.

Case 2: $b = 1$**.** We know that $\boldsymbol{\xi}_t^{\mathbf{v}}[1] = \mathbb{I}(\mathbf{v}[i_t] \geq p_t)$.

If $\boldsymbol{\xi}_t^{\mathbf{v}}[1] = 0$, then $\mathbf{v}[i_t] < p_t$. By the same logic as the previous case, this means that $a_{i_t,j_t}^{\mathbf{v}} = 0$. By Equation (5), this means that $\sum_{(i,j):a_{i,j}^{\mathbf{v}}=1} d_{i,j}(t) = 0$, so Equation (4) holds.

Meanwhile, if $\boldsymbol{\xi}_t^{\mathbf{v}}[1] = 1$, then $\mathbf{v}[i_t] \geq p_t$. By the same logic as the previous case, $a_{i_t,j_t}^{\mathbf{v}} = 1$. By Equation (5), this means that $\sum_{(i,j):a_{i,j}^{\mathbf{v}}=1} d_{i,j}(t) = 1$, so Equation (4) holds. □

This means that

$$\begin{aligned}
\boldsymbol{\xi}_t^{\mathbf{v}} \cdot \hat{\mathbf{r}}_t &= \sum_{b=0}^{1} \sum_{(i,j):a_{i,j}^{\mathbf{v}}=b} d_{i,j}(t) \hat{\mathbf{r}}_t[b] \\
&= \sum_{i=1}^{n} \sum_{j=0}^{m} d_{i,j}(t) \left(\hat{\mathbf{r}}_t[0] \mathbb{I}(a_{i,j}^{\mathbf{v}} = 0) + \hat{\mathbf{r}}_t[1] \mathbb{I}(a_{i,j}^{\mathbf{v}} = 1) \right) \\
&= \sum_{i=1}^{n} \sum_{j=0}^{m} d_{i,j}(t) \hat{\mathbf{r}}_t[a_{i,j}^{\mathbf{v}}].
\end{aligned}$$

Similarly,

$$\sum_{b=0}^{1} \frac{\boldsymbol{\xi}_t^{\mathbf{v}}[b]}{\bar{\boldsymbol{\xi}}_t^{\mathbf{v}}[b]} = \sum_{b=0}^{1} \sum_{(i,j):a_{i,j}^{\mathbf{v}}=b} \frac{d_{i,j}(t)}{\bar{\boldsymbol{\xi}}_t^{\mathbf{v}}[b]} = \sum_{i=1}^{n} \sum_{j=0}^{m} \frac{d_{i,j}(t)}{\bar{\boldsymbol{\xi}}_t^{\mathbf{v}}[a_{i,j}^{\mathbf{v}}]}.$$

Therefore,

$$w_{t+1}^{\mathbf{v}} = w_t^{\mathbf{v}} \exp\left(\frac{\gamma}{2} \left(\boldsymbol{\xi}_t^{\mathbf{v}} \cdot \hat{\mathbf{r}}_t + \sum_{b=0}^{1} \frac{\boldsymbol{\xi}_t^{\mathbf{v}}[b]}{\bar{\boldsymbol{\xi}}_t^{\mathbf{v}}[b]} \sqrt{\frac{\log |\mathcal{V}|/\delta}{2(T-\tau)}} \right) \right) = w_t^{\mathbf{v}} \exp\left(\sum_{i=1}^{n} \sum_{j=0}^{m} d_{i,j}(t) f_{a_{i,j}^{\mathbf{v}}}(t) \right),$$

where

$$f_{a_{i,j}^{\mathbf{v}}}(t) = \frac{\gamma}{2} \left(\hat{\mathbf{r}}_t[a_{i,j}^{\mathbf{v}}] + \frac{1}{\bar{\boldsymbol{\xi}}_t^{\mathbf{v}}[a_{i,j}^{\mathbf{v}}]} \sqrt{\frac{\log |\mathcal{V}|/\delta}{2(T-\tau)}} \right).$$

We can therefore write

$$\begin{aligned} w_{t+1}^{\mathbf{v}} &= \prod_{\tau=1}^{t} \exp\left(\sum_{i=1}^{n}\sum_{j=0}^{m} d_{i,j}(\tau) f_{a_{i,j}^{\mathbf{v}}}(\tau)\right) \\ &= \exp\left(\sum_{\tau=1}^{t}\sum_{i=1}^{n}\sum_{j=0}^{m} d_{i,j}(\tau) f_{a_{i,j}^{\mathbf{v}}}(\tau)\right) \\ &= \exp\left(\sum_{i=1}^{n}\sum_{j=0}^{m}\sum_{\tau=1}^{t} d_{i,j}(\tau) f_{a_{i,j}^{\mathbf{v}}}(\tau)\right) \\ &= \prod_{i=1}^{n}\prod_{j=0}^{m} \exp\left(\sum_{\tau=1}^{t} d_{i,j}(\tau) f_{a_{i,j}^{\mathbf{v}}}(\tau)\right). \end{aligned}$$

Letting $g_{i,j}(t,b) = \exp\left(\sum_{\tau=1}^{t} d_{i,j}(\tau) f_b(\tau)\right)$, we have that

$$w_{t+1}^{\mathbf{v}} = \prod_{i=1}^{n}\prod_{j=0}^{m} g_{i,j}(t, a_{i,j}^{\mathbf{v}}).$$

Let $\mathbf{b}_1, \ldots, \mathbf{b}_m$ be the set of m increasing bit vectors $\mathbf{b}_1 = (1,0,0,\ldots,0)$, $\mathbf{b}_2 = (1,1,0,\ldots,0)$, $\mathbf{b}_3 = (1,1,1,\ldots,0)$, $\ldots$, $\mathbf{b}_m = (1,1,1,\ldots,1)$. For each $i \in [n]$ and $\mathbf{b}_j$, define

$$g_i(t, \mathbf{b}_j) = g_{i,1}(t, b_j[1]) g_{i,2}(t, b_j[2]) \cdots g_{i,m}(t, b_j[m]).$$

We now prove the following claim.

Claim C.2. *The normalizing constant W_t can be computed in polynomial time as*

$$W_t = \prod_{i=1}^{n}\sum_{j=1}^{m_i} g_i(t, \mathbf{b}_j).$$

Proof of Claim C.2. We first write

$$\begin{aligned} W_t &= \sum_{\mathbf{v}\in\mathcal{V}} w_t^{\mathbf{v}} \\ &= \sum_{\mathbf{v}\in\mathcal{V}} \prod_{i=1}^{n}\prod_{j=0}^{m} g_{i,j}(t, a_{i,j}^{\mathbf{v}}) \\ &= \sum_{\mathbf{v}\in\mathcal{V}} \prod_{i=1}^{n}\prod_{j=0}^{m} g_{i,j}(t, \mathbb{I}(\mathbf{v}[i] \geq p_{i,j})) \\ &= \sum_{j_1=0}^{m_1} \cdots \sum_{j_n=0}^{m_n} \prod_{i=1}^{n}\prod_{j=0}^{m} g_{i,j}(t, \mathbb{I}(p_{i,j_i} \geq p_{i,j})). \end{aligned}$$

This last equality holds because $\mathcal{V} = \times_{i=1}^{n} \mathcal{V}_i$ and $\mathcal{V}_i = \{p_{i,0}, \ldots, p_{i,m_i}\}$. Moreover,

$$W_t = \sum_{j_1=0}^{m_1} \cdots \sum_{j_n=0}^{m_n} \prod_{i=1}^{n}\prod_{j=0}^{j_i} g_{i,j}(t,1) \prod_{j=j_i+1}^{m} g_{i,j}(t,0)$$

because $p_{i,j_i} \geq p_{i,j}$ for all $j \leq j_i$ and $p_{i,j_i} < p_{i,j}$ for all $j > j_i$.

By definition of the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m$,

$$W_t = \sum_{j_1=0}^{m_1} \cdots \sum_{j_n=0}^{m_n} \prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) = \prod_{i=1}^{n} \sum_{j=1}^{m_i} g_i(t, \mathbf{b}_j),$$

as claimed. □

We next prove that $\bar{\boldsymbol{\xi}}_t$ can be computed in polynomial time. To do so, let $h(\mathbf{x}_t) = i_t$. If $p_t = 0$, define $j_t = 0$ and otherwise, define j_t such that $p_t \in (p_{i_t, j_t - 1}, p_{i_t, j_t}]$. Next, define $\bar{m}_i$ as follows:

$$\bar{m}_i = \begin{cases} j_t - 1 & \text{if } i = i_t \\ m_i & \text{otherwise.} \end{cases}$$

Similarly, define $\underline{m}_i$ as follows:

$$\underline{m}_i = \begin{cases} j_t & \text{if } i = i_t \\ 0 & \text{otherwise.} \end{cases}$$

Then $\bar{\boldsymbol{\xi}}_t$ has the following form:

Claim C.3. *The probabilities $\bar{\boldsymbol{\xi}}_t$ can be computed in polynomial time as:*

$$\bar{\boldsymbol{\xi}}_t[0] = \frac{1}{W_t} \prod_{i=1}^{n} \sum_{j_i=0}^{\bar{m}_i} g_i(t, \mathbf{b}_{j_i})$$

and

$$\bar{\boldsymbol{\xi}}_t[1] = \frac{1}{W_t} \prod_{i=1}^{n} \sum_{j_i=\underline{m}_i}^{m_i} g_i(t, \mathbf{b}_{j_i}).$$

Proof of Claim C.3. Recall that $\boldsymbol{\xi}_t^{\mathbf{v}}[0] = \mathbb{I}(\mathbf{v}[i_t] < p_t)$. Therefore,

$$\begin{aligned} \sum_{\mathbf{v} \in \mathcal{V}} w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[0] &= \sum_{\mathbf{v} \in \mathcal{V}} \prod_{i=1}^{n} \prod_{j=0}^{m} g_{i,j}(t, a_{i,j}^{\mathbf{v}}) \boldsymbol{\xi}_t^{\mathbf{v}}[0] \\ &= \sum_{\mathbf{v} \in \mathcal{V}} \prod_{i=1}^{n} \prod_{j=0}^{m} g_{i,j}(t, a_{i,j}^{\mathbf{v}}) \mathbb{I}(\mathbf{v}[i_t] < p_t) \\ &= \sum_{j_1=0}^{m_1} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \mathbb{I}(p_{i_t, j_{i_t}} < p_t). \end{aligned}$$

This last equality holds because $\mathcal{V} = \times_{i=1}^{n} \mathcal{V}_i$ and $\mathcal{V}_i = \{p_{i,0}, \ldots, p_{i,m_i}\}$. Without loss of generality, suppose that $i_t = 1$, so $\mathbb{I}(p_{i_t, j_{i_t}} < p_t) = \mathbb{I}(p_{1,j_1} < p_t)$. Then

$$\begin{aligned} \sum_{\mathbf{v} \in \mathcal{V}} w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[0] &= \sum_{j_1=0}^{m_1} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \mathbb{I}(p_{1,j_1} < p_t) \\ &= \sum_{j_1=0}^{m_1} \mathbb{I}(p_{1,j_1} < p_t) \left(\sum_{j_2=0}^{m_2} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \right). \end{aligned}$$

Since $p_t = 0$ if $j_t = 0$ and otherwise $p_t \in (p_{1,j_t-1}, p_{1,j_t}]$ we have that $\mathbb{I}(p_{1,j_1} < p_t) = 1$ if and only if $j_1 \leq j_t - 1$. Therefore,

$$\sum_{\mathbf{v} \in \mathcal{V}} w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[0] = \sum_{j_1=0}^{j_t-1} \left(\sum_{j_2=0}^{m_2} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \right) = \sum_{j_1=0}^{\bar{m}_1} \cdots \sum_{j_n=0}^{\bar{m}_n} \prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) = \prod_{i=1}^{n} \sum_{j_i=0}^{\bar{m}_i} g_i(t, \mathbf{b}_{j_i}).$$

Similarly, since $\boldsymbol{\xi}_t^{\mathbf{v}}[1] = \mathbb{I}(\mathbf{v}[i_t] \geq p_t)$, we have

$$\sum_{\mathbf{v}\in\mathcal{V}} w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[1] = \sum_{j_1=0}^{m_1} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \mathbb{I}(p_{i_t, j_{i_t}} \geq p_t).$$

Without loss of generality, suppose that $i_t = 1$, so $\mathbb{I}(p_{i_t, j_{i_t}} \geq p_t) = \mathbb{I}(p_{1,j_1} \geq p_t)$. Then

$$\sum_{\mathbf{v}\in\mathcal{V}} w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[1] = \sum_{j_1=0}^{m_1} \mathbb{I}(p_{1,j_1} \geq p_t) \left(\sum_{j_2=0}^{m_2} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \right).$$

Since $p_t = 0$ if $j_t = 0$ and otherwise $p_t \in (p_{1,j_t-1}, p_{1,j_t}]$ we have that $\mathbb{I}(p_{1,j_1} \geq p_t) = 1$ if and only if $j_1 \geq j_t$. Therefore,

$$\begin{aligned}
\sum_{\mathbf{v}\in\mathcal{V}} w_t^{\mathbf{v}} \boldsymbol{\xi}_t^{\mathbf{v}}[0] &= \sum_{j_1=j_t}^{m_1} \left(\sum_{j_2=0}^{m_2} \cdots \sum_{j_n=0}^{m_n} \left(\prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \right) \right) \\
&= \sum_{j_1=\underline{m}_1}^{m_1} \cdots \sum_{j_n=\underline{m}_n}^{m_n} \prod_{i=1}^{n} g_i(t, \mathbf{b}_{j_i}) \\
&= \prod_{i=1}^{n} \sum_{j_i=\underline{m}_i}^{m_i} g_i(t, \mathbf{b}_{j_i}).
\end{aligned}$$

□

Lastly, note that in the initialization phase of Algorithm 2, every decision b_t is computed with $\mathcal{O}(1)$ time. In the `Exp4` subroutine, by Claim C.2 and Claim C.3, every iteration is also computed in polynomial time. Therefore the theorem statement holds. □