Agnostic learning with unknown utilities

Kush Bhatia[†]   Peter L. Bartlett[†,‡]   Anca D. Dragan[†]   Jacob Steinhardt[‡]

Department of Electrical Engineering and Computer Sciences, UC Berkeley[†]
Department of Statistics, UC Berkeley[‡]

July 8, 2021

Abstract

Traditional learning approaches for classification implicitly assume that each mistake has the same cost. In many real-world problems though, the utility of a decision depends on the underlying context x and decision y; for instance, misclassifying a stop sign is worse than misclassifying a road-side postbox. However, directly incorporating these utilities into the learning objective is often infeasible since these can be quite complex and difficult for humans to specify.

We formally study this as *agnostic learning with unknown utilities*: given a dataset $S = \{x_1, \ldots, x_n\}$ where each data point $x_i \sim \mathcal{D}_x$ from some unknown distribution $\mathcal{D}_x$, the objective of the learner is to output a function f in some class of decision functions $\mathcal{F}$ with small excess risk. This risk measures the performance of the output predictor f with respect to the best predictor in the class $\mathcal{F}$ on the *unknown* underlying utility $u^* : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$. This utility u^* is not assumed to have any specific structure and is allowed to be any bounded function. This raises an interesting question whether learning is even possible in our setup, given that obtaining a generalizable estimate of utility u^* might not be possible from finitely many samples. Surprisingly, we show that estimating the utilities of only the sampled points S suffices to learn a decision function which generalizes well.

With this insight, we study mechanisms for eliciting information from human experts which allow a learner to estimate the utilities u^* on the set S. While humans find it difficult to directly provide utility values reliably, it is often easier for them to provide comparison feedback based on these utilities. We show that, unlike in the realizable setup, the vanilla comparison queries where humans compare a pair of decisions for a single input x are insufficient. We introduce a family of elicitation mechanisms by generalizing comparisons, called the k-comparison oracle, which enables the learner to ask for comparisons across k different inputs x at once. We show that the excess risk in our agnostic learning framework decreases at a rate of $O\left(\frac{1}{k}\right)$ with such queries. This result brings out an interesting accuracy-elicitation trade-off – as the order k of the oracle increases, the comparative queries become harder to elicit from humans but allow for more accurate learning.

1   Introduction

Our focus is on learning predictive models for decision-making tasks. Current paradigms for classification tasks use datasets consisting of scenarios[1] x along with the decisions y taken by human experts to learn a decision function[2] $f : \mathcal{X} \mapsto \mathcal{Y}$. For instance, in economics such decisions correspond to whether buyers bought an item at a suggested price [Afr67; BV06], in robotics such feedback comprises expert demonstrations in imitation learning [AN04; Arg+09], and in machine learning literature such supervision consists of labels selected by human annotators [Bis06; DHS12].

---

[1]We use the term scenario/context/feature for the vector x interchangeably.
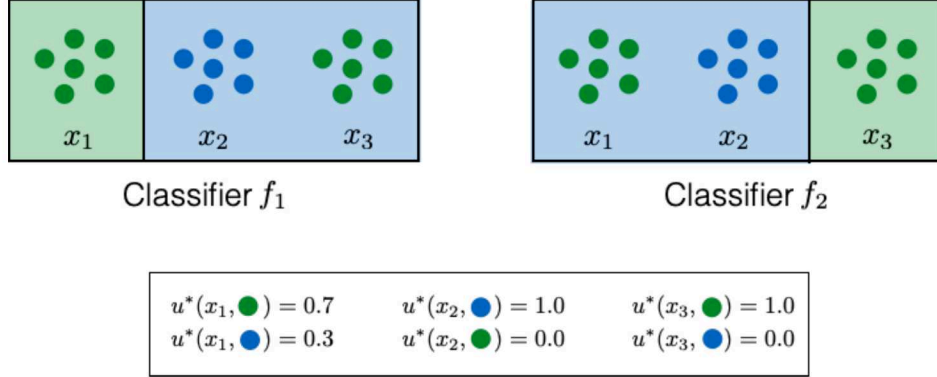[2]We consider finite decision spaces $\mathcal{Y}$.

Figure 1. Consider a binary decision-task with decisions G(reen) and B(lue). The instance space comprises of three equiprobable clusters of datapoints x_1, x_2 and x_3, and have associated utilities u^* for decisions B and G. The colour of the datapoints represents the decision with higher utility. The function class $\mathcal{F}$ consists of linear predictors. In the traditional learning setups where the dataset consists of pairs (x, y), no learner will have enough information to select between f_1 and f_2 since the $0 - 1$ error for both is $1/3$. In contrast, using a 2-comparison oracle, a learner can ask a query of the form "Which of $u^*(x_1, \mathrm{G}) + u^*(x_3, \mathrm{B})$ or $u^*(x_1, \mathrm{B}) + u^*(x_3, \mathrm{G})$ is bigger?". This allows them to infer that correctly predicting x_3 gives a higher overall utility and output the optimal decision function f_2.

When we optimize models to predict correctly on these datasets, we often implicitly assume that all mistakes are equally costly, and that each scenario x in the data is just as important. In reality though, this is rarely the case. For instance, the standard $0 - 1$ loss for classification tasks assigns a unit of loss for each mistake, but misclassifying a stop sign is significantly more dangerous than misclassifying a road-side postbox. In Figure 1, we expand on this insight and illustrate how learning from such revealed decisions can often lead to suboptimal decision functions.

What is missing from this classical framework is that for most decision-making tasks there exists an underlying function $u^* : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ which evaluates the utility of a decision y depending on the surrounding context x. Depending on the decision task, such utility functions can encode buyer preferences in economics, rewards for robotic skills, or misprediction costs for classification. However, these utility functions are a priori unknown to the learner since the dataset consists only of context-decision pairs (x, y). Furthermore, asking human experts to write down these complex utility functions can be quite challenging and prone to serious errors [Amo+16].

One commonly studied approach, referred to as learning from revealed preferences in economics [BV06; Bal+14] and inverse reinforcement learning (IRL) in the machine learning literature [NR00; Zie+08], assumes that the utility function u^* belongs to some pre-specified class and uses the fact that decision y was the optimal decision for scenario x to learn estimates of these utilities. This setup is called the well-specified or realizable setup. However, this posited utility class can be misspecified in that the underlying utility u^* might not belong to this class. The correctness of such learning approaches crucially relies on the well specified assumption and offers no guarantees on how their performance degrades in the presence of class misspecifications.

We overcome this uncertainty in specifying the utility function u^* by proposing an *agnostic learning* framework which places no assumptions on the class of utility functions. Instead, we consider decision functions belonging to some class $\mathcal{F} = \{f \mid f : \mathcal{X} \mapsto \mathcal{Y}\}$ and study the objective of obtaining the "best" decision rule in $\mathcal{F}$ with respect to the unknown utility u^*. Formally, given the decision class $\mathcal{F}$ and samples from a distribution $\mathcal{D}_x$ over the feature space $\mathcal{X}$, the objective of

the learner is to output a model $\hat{f} \in \mathcal{F}$ with small excess risk or regret

$$\mathsf{err}(\hat{f}, \mathcal{F}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim \mathcal{D}_x}[u^*(x, f(x))] - \mathbb{E}_{x \sim \mathcal{D}_x}[u^*(x, \hat{f}(x))] . \tag{1}$$

Our proposed notion of excess risk measures the performance of an estimator $\hat{f}$ by comparing its decisions with those of the best predictive model in the class $\mathcal{F}$ under the utility u^*. Contrast this with the classical agnostic learning framework [Hau92] where the evaluation metric for classification measures what proportion of datapoints $\hat{f}$ predicts correctly

$$\mathsf{err}_{\mathsf{cl}}(\hat{f}, \mathcal{F}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{x \sim \mathcal{D}_x}[\mathbb{I}[f(x) \neq y_x]] - \mathbb{E}_{x \sim \mathcal{D}_x}[\mathbb{I}[\hat{f}(x) \neq y_x]] , \tag{2}$$

where $y_x = \mathrm{argmax}_{y \in \mathcal{Y}} u^*(x, y)$ represents the expert decision (revealed decision) for scenario x. Our above framework generalizes the proper agnostic learning framework – we restrict our attention to proper learners which output models $\hat{f} \in \mathcal{F}$ and the decision class $\mathcal{F}$ is agnostic towards the unknown underlying utility u^*. Indeed, our agnostic framework allows for misspecification in the decision class $\mathcal{F}$ and allows for situations where no predictive model $f \in \mathcal{F}$ matches the expert predictions y_x for all instances x.

As highlighted by Figure 1, such a misspecification in the function class $\mathcal{F}$ implies that no decision function $f \in \mathcal{F}$ will be able to perfectly fit these optimal decisions y_x for all points $x \in S$. In order to solve the agnostic learning problem, it is necessary for the learner to understand the how costly these different mistakes are relative to each other. From the learners perspective, observing only the optimal decisions y_x for each instance x, such as revealed preferences or expert demonstrations, are clearly insufficient to obtain any information about these costs. One way to overcome this information-theoretic limit of revealed decisions is to directly elicit the utilities from humans – for scenarios x and decision y, ask an expert "What is the utility $u^*(x, y)$ for taking the decision y given situation x?". However, since the underlying utility u^* can be quite complex, humans are inept at answering them reliably [Mil56; SBC05]. For instance, it can be challenging for humans to correctly specify the costs of mispredicting, say, a stop sign as a red signal relative to that of predicting it as a post-box.

On the other hand, it is often easier for humans to provide comparative evaluations based on these utilities [Thu27; FH10] and allow the learner to obtain relative feedback. Using these, the learner can query an expert with comparison or preference queries asking "For instance x, which of the two utilities $u^*(x, y_1)$ or $u^*(x, y_2)$ is larger?". Such vanilla comparisons can allow the learner to infer relative utilities for decisions y_1 and y_2 for a given context x; the learner can conclude that mispredicting stop sign as post-box is worse than mispredicting it as a red signal. However, such feedback still does not provide any information about the mistake costs across different examples – given a choice, should the learner correctly predict a stop-sign or correctly predict a post-box?

While vanilla comparisons are insufficient for the agnostic setup, let us consider the other extreme: suppose that we have access to an oracle which can provide us with comparisons of overall utilities for functions $f_1, f_2 \in \mathcal{F}$. That is, the oracle can answer question of the form "Which of the two overall utilities $\mathbb{E}_x[u^*(x, f_1(x))]$ or $\mathbb{E}_x[u^*(x, f_2(x))]$ is larger?". Given access to such an oracle, we will be able to find the optimal classifier in the class $\mathcal{F}$. We call this the ∞-comparison oracle since such preferences requires a human to reason about the utilities over the entire feature space $\mathcal{X}$ at once. Even for a small image classification task with a million images, this would require a human to compare the utility of a million simultaneous predictions! While this approach does

allows for optimal estimation, the trade-off is that it puts the complete burden of learning on the human's side. It is worth highlighting that the comparisons between lotteries used to establish the von Neumann-Morgenstern utility theorem [MV53] can be shown to be a special case of such an ∞-comparison oracle.

While comparison queries only allow comparison within a single instance, the ∞-comparison oracle takes the other extreme and requires a comparison along all instances. However, we need not restrict our self to either of these extremes; our key insight is that there is a natural spectrum of such comparisons, which we call k-comparisons which interpolate between the single or 1-comparison and the ∞-comparison oracle. Such comparison queries allow a learner to pick k instances $\{x_1, \ldots, x_k\}$ and two sets of corresponding decision, $\{y_1, \ldots, y_k\}$ and $\{y'_1, \ldots, y'_k\}$, and ask "Which of the cumulative utilities $\sum_i u^*(x_i, y_i)$ or $\sum_i u^*(x_i, y'_i)$ is bigger?". For instance, for the example in Figure 1, giving the learner access to a 2-comparison oracle allows the algorithm to output the optimal decision function.

These higher-order comparison oracles form a natural hierarchy of elicitation mechanisms for the learner with a k'-oracle being strictly more informative than a k-oracle for $k' > k$. They allow for a natural trade-off between accuracy and elicitation in the learning with unknown utilities framework. As we increase the order k of the oracle, the learner can obtain finer information about the utilities u^* and output functions with lower excess risk. However, this increase in information comes at the expense of asking for a harder elicitation from the human expert.

Our Contributions. We propose a novel framework, which we call *agnostic learning with unknown utilities*, for studying decision problems wherein the learner is evaluated with respect to an unknown utility function. Within this framework, we show that standard approaches which work well in the realizable setup, such as revealed preferences as well as vanilla comparisons, can perform quite poorly in the face of misspecification and can have excess risk $\Omega(1)$. To overcome this, we propose a family of elicitation mechanisms, the k-comparisons, which allows the learner access to finer information from an human expert with increasing values of the order k. Our main results, detailed in Section 3, provide a tight characterization of the excess risk as a function of the order k of the comparison oracle available to the learner. These result brings out an interesting accuracy-elicitation trade-off – as the order k of the oracle increases, the comparative queries allow for more accurate learning in our setup but become harder to elicit from humans.

We would like to highlight that increasing the order k of the comparisons could lead to potentially biased and noisy responses from the human expert. As a consequence, there might be an additional trade-off involving the *quality of the information* obtained by increasing the order. While we do not focus on this aspect of elicitation, it is an interesting direction for future work.

Paper Organization. The remainder of the paper is organized as follows: Section 2 introduces our agnostic learning with unknown utilities problem setup and the k-comparison elicitation mechanism, and Section 3 gives an overview of our main results and algorithmic contributions. In Section 4, we study excess risk bounds for the binary decision problem in our framework and propose our algorithm, Comptron, to learn from higher-order comparisons and in Section 5, we study adaptive estimators which are optimal for each instance of our problem.

2   Problem formulation

In this section, we formally state our learning with unknown utilities problem and introduce the k-comparison oracle. Let $\mathcal{X} \subseteq \mathbb{R}^d$ represent the space of feature vectors, $\mathcal{Y}$ denote the corresponding decision space and $\mathcal{F}$ denote a class of decision making functions, given as $\mathcal{F} = \{f \mid f : \mathcal{X} \mapsto \mathcal{Y}\}$. Our framework considers an underlying utility function $u^* : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ which assigns a non-negative real value for making a decision $y \in \mathcal{Y}$ given a situation $x \in \mathcal{X}$. Further, let us denote the set

$$\mathcal{U} = \{u \mid u : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]\} \tag{3}$$

of all possible such utility functions. For any distribution $\mathcal{D}_x$ over the feature space $\mathcal{X}$, we define the expected utility of a decision function $f \in \mathcal{F}$ as $U(f; u^*) := \mathbb{E}_{x \sim \mathcal{D}_x}[u^*(x, f(x))]$. Observe that such an expected utility model assumes that the utilities are additive across the different instances x and is a commonly studied model both in the machine learning, statistics and economics literature. We denote the excess risk of a function f with respect to the function class $\mathcal{F}$ by

$$\mathsf{err}(f, \mathcal{F}; u^*) := \max_{f' \in \mathcal{F}} U(f'; u^*) - U(f; u^*). \tag{4}$$

Further, we denote the optimal decision for any instance x with respect to the underlying utility u^* by $y_x := \mathrm{argmax}_{y \in \mathcal{Y}} u^*(x, y)$.

Similar to the classical agnostic learning setup [Hau92], we assume that the learner does not know the underlying distribution $\mathcal{D}_x$ of the instances. However, our setup differs from it in that we do not assume that the underlying utility function u^* is known to the learner. Instead, we provide the learner access to an oracle which allows the learner to elicit responses to higher-order preferences queries.

Comparison Oracle   Since the utility function u^* is unknown to the learner, our framework allows the learner access to an oracle which provides comparative feedback based on the utilities u^*. We consider a family of such oracles $\mathcal{O}_k$, each indexed by its order k which determines the number of different instances the learner is allowed to specify in the comparison query. For an oracle $\mathcal{O}_k$, a learner is allowed to select a set of k situations $\mathbf{x} \in \mathcal{X}^k$ and two pairs of corresponding decisions $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}^k$. The oracle then compares, in a possibly noisy manner, the cumulative utilities of the pair $(\mathbf{x}, \mathbf{y}_1)$ and $(\mathbf{x}, \mathbf{y}_2)$ and responds with the feedback on which one is larger. As the order k of the oracle increases, the queries become more complex – an expert is required to evaluate a larger number of instances at once. This family of comparison oracles captures a natural hierarchy of elicitation mechanisms where with each increasing value of k, a learner has access to more information about the utility function u^*.

Formally, we represent a k-query by a tuple $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where the input $\mathbf{x} = (x_1, \ldots, x_k)$ comprises k feature vectors and the corresponding decision vectors $\mathbf{y}_1 = (y_1, \ldots, y_k)$ and $\mathbf{y}_2 = (y'_1, \ldots, y'_k)$.[3] Given such a query q, the oracle $\mathcal{O}_k$ provides the learner a binary response

$$\mathcal{O}_k(q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)) = \begin{cases} \mathbb{I}\left[u^*(\mathbf{x}, \mathbf{y}_1) \geq u^*(\mathbf{x}, \mathbf{y}_2)\right] & \text{with prob. } 1 - \eta_q \\ 1 - \mathbb{I}\left[u^*(\mathbf{x}, \mathbf{y}_1) \geq u^*(\mathbf{x}, \mathbf{y}_2)\right] & \text{otherwise} \end{cases}, \tag{5}$$

---

[3]We overload our notation and represent the cumulative utilities of the k inputs $(\mathbf{x}, \mathbf{y})$ by $u^*(\mathbf{x}, \mathbf{y}) = \sum_i u^*(x_i, y_i)$.

where the parameter $0 \leq \eta_q < \frac{1}{2}$ represents the noise level corresponding to query q. Thus, the oracle[4] $\mathcal{O}_k$ provides noisy comparisons of the cumulative utilities $u^*(\mathbf{x}, \mathbf{y}_1)$ and $u^*(\mathbf{x}, \mathbf{y}_2)$ with varying noise level η_q. Observe that we allow the noise levels η_q to be different for each query q.

Problem Statement We are interested in the *agnostic learning with unknown utilities* problem where a learner is provided n samples $S = \{x_1, \ldots, x_n\}$ with each $x_i \sim \mathcal{D}_x$ and access to the k-comparison oracle described above, and is required to output a decision function $\hat{f} \in \mathcal{F}$ such that error $\mathsf{err}(\hat{f}, \mathcal{F})$ is small. The caveat is to do so with a minimum number of calls, which we term the query complexity n_q of learning, to the comparison oracle $\mathcal{O}_k$. Quantitatively, we would like to characterize the excess risk from equation (4) in terms of the number of sampled instances n, the order k of the comparison oracle and properties of the decision function class $\mathcal{F}$, and the associated oracle query complexity n_q to obtain this bound.

Obtaining such bounds on the excess risk $\mathsf{err}(f, \mathcal{F}; u^*)$ in terms of the order k allow us to quantify the trade-offs in learning better decision functions at the expense of requiring more complex information from the human expert. Going forward, we focus on the binary decision making problem where the label space $\mathcal{Y} = \{0, 1\}$ for clarity of exposition. Whenever our results can be extended to arbitrary decision sets, we provide a small remark about this extension.

3 Main results

With the formal problem setup in place, we discuss our main results for learning in this framework of unknown utilities. At a high level, our objective is to understand how the excess risk $\mathsf{err}(f, \mathcal{F}; u^*)$ defined in equation (4) behaves as a function of the oracle order k – specifically, at what rates does learning in our proposed framework get easier as we allow learner to elicit more complex information from the oracle?

For our main results, on the upper bound side, we design estimators for learning from the k-comparison oracle, and on the lower bound side, we study information-theoretic limits of learning with such higher-order comparisons. While we state our results for the binary decision problem where the label space $\mathcal{Y} = \{0, 1\}$ for clarity, most of our results can be generalized to arbitrary outcome space $\mathcal{Y}$.

3.1 Excess risk with k-comparison oracle (Section 4)

We study a class of *plug-in* estimators which are based on the following two-step procedure:

   i. Obtain estimate $\hat{u}$ of the true utility u^* on the sampled datapoints.

   ii. Output utility maximizing function $\hat{f}_{k,n}$ with respect to the estimated utility $\hat{u}$.

For learning the parameters $\hat{u}$, we introduce the Comptron (Algorithm 1) and Rob-Comptron (Algorithm 2) algorithms for the noiseless and noisy comparison oracles respectively. We show that when these estimates $\hat{u}$ are combined with the two-step plug-in estimator, the excess risk of the function $\hat{f}_{k,n}$ scales as $O(\frac{1}{k})$ and an additive complexity term capturing uniform convergence of the decision class $\mathcal{F}$ with respect to the true utility u^*.

---

[4]Note that while the oracle depends on the underlying utility function u^*, our notation suppresses this dependence for clarity. We use the notation $\mathcal{O}_k(q; u^*)$ whenever we want to make this dependence explicit.

Theorem 1 (Informal, noiseless comparisons). *Given n samples, the excess risk for the function $\hat{f}_{k,n} \in \mathcal{F}$ output by the plug-in estimator using estimates $\hat{u}$ from Comptron satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq \mathsf{Complexity}_n(\mathcal{F}; u^*) + O\left(\frac{1}{k}\right) \cdot \left(\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]\right) ,$$

where the ERM function $f_{\mathsf{ERM}} \in \arg\max_{f \in \mathcal{F}} \sum_{i=1}^{n} u^*(x_i, f(x_i))$. Furthermore, Comptron makes only $O(n \log k)$ queries to the oracle $\mathcal{O}_k$.

We make a few remarks on this result. First, observe that the complexity term depends on the true utility function u^* and not on the estimates $\hat{u}$. This ensures that the complexity term does not depend on the utility class $\mathcal{U}$ but rather only on the specific utility u^* – indeed, the class $\mathcal{U}$ consists of all bounded function and uniform convergence might not even be possible with finite sample for a large class of distributions $\mathcal{D}_x$. Second, the additional error of $O(\frac{1}{k})$ accounts for the fact that the utilities u^* are unknown. One can learn better decision functions by increasing the order k of the comparison oracle but this comes at the cost of the human expert answering a more complex set of queries. Furthermore, this error is multiplied by the $0 - 1$ prediction error of the optimal on-sample classifier $f_{\mathsf{ERM}} = \arg\max_{f \in \mathcal{F}} \sum_i u^*(x_i, f(x_i))$. This implies that in the well-specified setup, where there exists an $f \in \mathcal{F}$ such that $f(x_i) = y_i$ on the sampled datapoints, the second term becomes 0 and the learner pays no additional error for not knowing the utilities u^*. Third, observe that our proposed algorithms, Comptron and Rob-Comptron, are query efficient; both require only $O(n \log k)$ calls to the k-comparison oracle to produce "good" estimates $\hat{u}$.

The proof of the above theorem proceeds in two steps. First, we adapt the classical proof for upper bounding the risk of ERM procedures to show that the gap $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F})$ decomposes into the complexity term and estimation error $\|\hat{u} - u^*\|_{S,\infty}$, evaluated on the dataset S. Next, we show that this estimation error scales as $O\left(\frac{1}{k}\right)$ for the Comptron and Rob-Comptron procedures.

Next, we address the optimality of the above plug-in procedure by studying the information-theoretic limits of learning with a k-comparison oracle. Specifically, in Theorem 3 we establish that the rate of $\frac{1}{k}$ is indeed minimax optimal – for any $k > 1$ and any predictor $\hat{f}$ in some class $\mathcal{F}$, we can construct utility functions u^* such that excess risk $\mathsf{err}(\hat{f}, \mathcal{F}; u^*) = \Omega\left(\frac{1}{k}\right)$. These lower bounds imply that traditional comparison based learning, corresponding to $k = 1$, is insufficient for learning good decision rules in our framework.

3.2 Instance-optimal learning (Section 5).

While the previous results show that the error rate of $O(\frac{1}{k})$ is optimal on worst-case instances, some instances of our learning with unknown utilities problem might be easier than these worst-case ones and one would expect the excess risk to be smaller for them. In this section, we study estimators whose error adapts to hardness of the specific problem instance.

To begin with, in Proposition 2 we establish that the plug-in estimator with Comptron estimates $\hat{u}$ is not optimal for all instances – it does not adapt to these easier instances. Inspired from the robust optimization literature, we introduce a randomized estimator p_{rob} and show that it is instance-optimal. Informally, we establish in Theorem 5 that for any instance $(\mathcal{D}_x, u^*, \mathcal{F})$ of the problem, the excess risk for p_{rob} is characterized by a local modulus of continuity; this modulus captures how quickly the optimal decision function in class $\mathcal{F}$ can change in a small neighborhood around u^* for the distribution $\mathcal{D}_x$. In Theorem 4, we derive a lower bound on the *local minimax*

excess risk and show that the local modulus is indeed the correct instance-dependent complexity measure for this problem.

However, note that such adaptivity to the hardness of the instance comes at the cost of query efficiency. Our estimator p_{rob} makes an exponential number $O(n^k)$ of calls to the oracle $\mathcal{O}_k$.

4 Binary decision-making with k-comparisons

In this section, we obtain upper and lower bounds on the excess risk for the binary prediction problem with unknown utilities where the learner can elicit utility information using a k-comparison oracle. In Section 4.1, we introduce algorithms which learn decision-making rules from higher-order preference queries and obtain upper bounds on the excess risk for such estimators. Then, in Section 4.2, we turn to the information-theoretic limits of learning from k-queries and obtain lower bounds on the minimax risk of any estimator.

Recall from Section 2, our setup gives the learner access to a dataset $S = \{x_1, \ldots, x_n\}$ comprising n points, each sampled i.i.d. from an underlying distribution $\mathcal{D}_x$ and to a comparison oracle $\mathcal{O}_k$. Before proceeding to define the estimator, we introduce some notation. For any function $f \in \mathcal{F}$, let us denote the empirical cumulative utility with respect to utility function u^* and the corresponding empirical utility maximizer as

$$\hat{U}_n(f; u^*) = \frac{1}{n} \sum_i u^*(x_i, f(x_i)) \quad \text{and} \quad f_{\mathsf{ERM}} \in \operatorname*{argmax}_{f \in \mathcal{F}} \hat{U}_n(f; u^*) \,, \tag{6}$$

where the subscript n encodes the dependence on the number of samples. If the underlying utility u^* were in fact known to the learner, it would have output the classifier f_{ERM}, which, from the classical learning theory literature, is known to have favorable generalization properties [Sha+10]. For the case of unknown utilities, we extend this ERM procedure to a natural two-stage plug-in estimator which outputs the minimizer with respect to an estimate $\hat{u}_k$ of these utilities.

4.1 Excess-risk upper bounds for plug-in estimator

Building on the ERM estimator f_{ERM} described in equation (6), we design a two stage *plug-in* estimator $\hat{f}_{k,n}$, where the subscript k represents the order of the comparison oracle used to obtain the estimate.

In the first stage, we form estimates $\hat{u}_k$ of the true utility function u^* on the sampled datapoints S using the k-comparison oracle. The predictor $\hat{f}_{k,n} \in \mathcal{F}$ is then given by the empirical utility maximizer with respect to $\hat{u}_k$, that is,

$$\hat{f}_{k,n} \in \operatorname*{argmax}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \hat{u}_k(x_i, f(x_i)). \tag{7}$$

Before detailing out the procedures for producing utility estimates $\hat{u}_k$, we present our first main result which shows that the excess risk $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*)$ can be upper bounded as a sum of two terms: (i) a complexity term corresponding to the rate of uniform convergence of the cumulative utility $U(f; u^*)$ over the decision class $\mathcal{F}$ and (ii) an estimation error term which denotes how well the estimates $\hat{u}_k$ approximate u^* on the sampled datapoints. Our result measures this estimation error

in terms of a data-dependent norm

$$\|u\|_{S,\infty} := \sup_{i \in [n]} \sup_{y \in \mathcal{Y}} |u(x_i, y)|. \tag{8}$$

Recall from equation (6) that the function f_{ERM} is the minimizer of the empirical utility $\hat{U}_n(f; u^*)$. While the following results hold for general decision spaces $\mathcal{Y}$, we later specialize this in Proposition 1 for the binary prediction setup.

Theorem 2 (Excess-risk upper bound). *Given datapoints $S = \{x_1, \ldots, x_n\}$ such that each $x_i \sim \mathcal{D}_x$, and an estimate $\hat{u}_k$ of the true utility function u^*, the plug-in estimate $\hat{f}_{k,n}$ from equation (7) satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}_n(f; u^*)| \right) + 2\|u^* - \hat{u}_k\|_{S,\infty} \cdot \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)] \right). \tag{9}$$

A few comments on Theorem 2 are in order. First, notice that the upper bound on the risk $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*)$ is a deterministic bound comprising two terms. The uniform convergence term captures how fast the empirical utility $\hat{U}_n(f; u^*)$ converge to the population utility $U(f; u^*)$ uniformly over the decision class $\mathcal{F}$. Using standard bounds [BM02], one can show that this term is upper bounded by the empirical Rademacher complexity of the class $\mathcal{F}$ on the datapoints S, that is,

$$\sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}_n(f; u^*)| \right) \leq \mathbb{E}_{\varepsilon} \left[\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i u^*(x_i, f(x_i)) \right| \right] := \widehat{\mathfrak{R}}_n(\mathcal{F} \circ u^*) \tag{10}$$

where each ε_i is an i.i.d. Rademacher random variable taking values $\{-1, +1\}$ equiprobably. Such complexity measures are commonly studied in the learning theory literature and one can obtain sample complexity rates for a wide range of decision classes including parametric decision classes and non-parametric kernel classes amongst others.

The second term in equation (9) is given by a product of two terms. The first part $\|u^* - \hat{u}_k\|_{S,\infty}$ captures the *on-sample approximation error* of the estimates $\hat{u}_k$. Notice that, in general, the problem of estimating u^* uniformly over the space $\mathcal{X}$ is infeasible since the class $\mathcal{U}$ contains the set of all bounded functions on $\mathcal{X} \times \mathcal{Y}$. However, the fact that we are required to estimate the utilities u^* only on the sampled datapoints S makes learning feasible in our framework. The second part, $\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)] \leq 1$ the mismatch between the predictions of f_{ERM}, obtained with complete knowledge of u^*, and of $\hat{f}_{k,n}$, obtained from estimates $\hat{u}_k$. Notice that whenever the function class $\mathcal{F}$ is correctly specified on S, that is, there exists a function $f \in \mathcal{F}$ such that $f(x_i) = y_i$), then the predictions of $\hat{f}_{k,n}$ and f_{ERM} will coincide. This follows since the labels y_i can be inferred using a 1-comparison. In such a well-specified setup, this second term vanishes and we recover the upper bound in terms of the uniform convergence term. Surprisingly, this exhibits that not knowing the utility u^* affects learnability only when the function class $\mathcal{F}$ is misspecified.

Proof. We begin by decomposing the excess error $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*)$ and then handle each term in the decomposition separately. Recall that the function f_{ERM} is the maximizer of the empirical

9

utility $\hat{U}_n(f; u^*)$. Then, for any decision function $f \in \mathcal{F}$, consider the error

$$\mathsf{err}(\hat{f}_{k,n}, f; u^*) = U(f; u^*) - \hat{U}_n(f; u^*) + \hat{U}_n(f; u^*) - \hat{U}_n(f_{\mathsf{ERM}}; u^*) + \hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(\hat{f}_{k,n}; u^*)$$
$$+ \hat{U}_n(\hat{f}_{k,n}; u^*) - U(\hat{f}_{k,n}; u^*)$$
$$\overset{(i)}{\leq} 2 \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}_n(f; u^*)| \right) + \underbrace{\hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(\hat{f}_{k,n}; u^*)}_{\text{Term (I)}}, \qquad (11)$$

where the inequality (i) follows by noting that f_{ERM} is the maximizer of $\hat{U}_n(f; u^*)$. We now focus our attention on Term (I) in the above expression.

$$\hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(\hat{f}_{k,n}; u^*) = \hat{U}_n(f_{\mathsf{ERM}}; u^*) - \hat{U}_n(f_{\mathsf{ERM}}; \hat{u}) + \hat{U}_n(f_{\mathsf{ERM}}; \hat{u}) - \hat{U}_n(\hat{f}_{k,n}; \hat{u})$$
$$+ \hat{U}_n(\hat{f}_{k,n}; \hat{u}) - \hat{U}_n(\hat{f}_{k,n}; u^*)$$
$$\overset{(i)}{\leq} \frac{2}{n} \sum_{i=1}^n \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)] \cdot \sup_{y \in \mathcal{Y}} |u^*(x_i, y) - \hat{u}(x_i, y)|$$
$$\leq 2\|u^* - \hat{u}\|_{S,\infty} \cdot \left(\frac{1}{n} \sum_{i=1}^n \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq \hat{f}_{k,n}(x_i)] \right) ,$$

where (i) follows by noting that $\hat{f}_{k,n}$ maximizes the utility $\hat{U}_n(f; \hat{u})$. Plugging the bound above in equation (11) completes the proof. $\qquad \square$

We now specialize the result of Theorem 2 to the binary prediction setup where the label space $\mathcal{Y} = \{0, 1\}$. Recall that for each datapoint x_i, we denote the true label by $y_i = \mathrm{argmax}_y u^*(x_i, y)$. We now introduce the notion of utility gaps $u_{\mathsf{gap}}(x_i)$ which measures the excess utility a learner gains by predicting a datapoint x_i correctly relative to an incorrect prediction. Formally, the gap $u_{\mathsf{gap}}(x_i)$ for datapoint x_i with respect to some utility function $u \in \mathcal{U}$ is given as

$$u_{\mathsf{gap}}(x_i) := u(x_i, y_i) - u(x_i, \bar{y}_i) , \qquad (12)$$

where we denote the incorrect label by $\bar{y} = 1 - y$. With this notation, the following proposition obtains an upper bound on the excess error of plug-in estimator $\hat{f}_{k,n}$ for the binary prediction problem in terms of the estimation error in these gaps $u_{\mathsf{gap}}(x_i)$.

Proposition 1 (Upper bounds for binary prediction). *Consider the binary decision making problem with label space $\mathcal{Y} = \{0, 1\}$. Given n datapoints $\{x_1, \ldots, x_n\}$ such that each datapoint $x_i \sim \mathcal{D}_x$, and an estimate $\hat{u}_k$ of the utility function u^*, the plug-in estimator $\hat{f}_{k,n}$ from equation (7) satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)| \right) + 2 \max_i [u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)] \cdot \left(\frac{1}{n} \sum_{i=1}^n \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i] \right) . \qquad (13)$$

The proof of the above proposition follows similar to Theorem 2 and is deferred to Appendix B. This specializes the result of Theorem 2 and shows that for the binary prediction problem, estimating the utility gaps u_{gap} well for each datapoint suffices

The upper bound on excess risk given by Proposition 1 shows that the function $\hat{f}_{k,n}$ derived from estimates $\hat{u}_k$ will have small error as long as the estimates $\hat{u}_{\mathsf{gap}}(x_i)$ approximate the true utility

gaps $u^*_{\mathsf{gap}}(x_i)$ for each datapoint x_i. Therefore, in the following sections, we focus on procedures for obtaining the utility estimates $\hat{u}_{\mathsf{gap}}$ using the k-comparison oracle. we separate the presentation based on whether the oracle $\mathcal{O}_k$ provides noiseless comparisons ($\eta_q = 0$ for all q) or whether the oracle evaluations are noisy.

4.1.1 Estimating u^*_{gap} with noiseless oracle

In this section, we propose our algorithm for estimating the gaps u^*_{gap} when the k-comparison oracle is noiseless. Recall from equation (5), for a query $q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ comprising k feature vectors $\mathbf{x} = (x_1, \ldots, x_k)$, and two decision vectors $\mathbf{y}_1 = (y_1, \ldots, y_k)$ and $\mathbf{y}_2 = (y'_1, \ldots, y'_k)$, such a noiseless oracle deterministically outputs

$$\mathcal{O}_k(q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)) = \mathbb{I}\left[u^*(\mathbf{x}, \mathbf{y}_1) \geq u^*(\mathbf{x}, \mathbf{y}_2)\right] \ ,$$

where recall that $u^*(\mathbf{x}, \mathbf{y}) = \sum_{i \in [k]} u^*(x_i, y_i)$ is the sum of the utilities under u^* for the tuple $(\mathbf{x}, \mathbf{y})$. In the binary prediction setup, such queries allow a learner to specify a set of k instances $\mathbf{x}$ and a subset $S_q \subset \mathbf{x}$ and ask the oracle "whether correctly predicting instances in S_q has higher utility or the instances in the complement $\mathbf{x} \setminus S_q$?".

Recall that Proposition 1 shows that excess risk for the plug-in estimator can be bounded by the worst-error $|u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)|$ over the set of sampled datapoints S. To obtain such estimates, we introduce *Comptron* in Algorithm 1 which is a coordinate-wise variant of the classical perceptron algorithm [Ros58]. At a high level, Comptron is an iterative procedure which estimates the utility gaps $u^*_{\mathsf{gap}}(x_i)$ for each x_i relative to the largest gap

$$u^*_{\mathsf{max}} := \max_{i \in [n]} u^*_{\mathsf{gap}}(x_i) \leq 1. \tag{14}$$

At each iteration t, the queries $q_{i,t}$ are selcted such that $\hat{u}^{t-1}_{\mathsf{gap}}(\mathbf{x}, \mathbf{y}_1) > \hat{u}^{t-1}_{\mathsf{gap}}(\mathbf{x}, \mathbf{y}_2)$ under the current estimates $\hat{u}^{t-1}_{\mathsf{gap}}$. If the oracle's response is $r_{i,t} = 1$, the estimates are consistent with the response and it keeps the current estimate. On the other hand, if the response $r_{i,t} = 0$, the algorithm decreases its current estimate of the ith datapoint in order to be consistent with this query. Comptron repeats the above procedure for $T = \log_2 k - 1$ timesteps and finally outputs the estimates $\hat{u}^T_{\mathsf{gap}}$.

It is worth highlighting here that Comptron initializes all the estimates as the largest gap, that is, $\hat{u}^0_{\mathsf{gap}}(x_i) = u^*_{\mathsf{max}}$. Such an initialization is purely symbolic in nature and the algorithm *does not* require knowledge of this value. This is because the comparison queries $q_{i,t}$ allows the algorithm to compare the estimates $\hat{u}_{\mathsf{gap}}$ with u^*_{max} and the algorithm maintains its estimates $\hat{u}^t_{\mathsf{gap}}$ as a multiplicative factor of u^*_{max} for iterations t. Further, we can use symbolic estimates to output the plug-in estimator since it is invariant to scaling the utility gaps by a positive constant,

$$\operatorname*{argmax}_{f \in \mathcal{F}} \sum_{i=1}^n \hat{u}(x_i, f(x_i)) \equiv \operatorname*{argmax}_{f \in \mathcal{F}} \sum_{i=1}^n \hat{u}_{\mathsf{gap}}(x_i) \cdot \mathbb{I}[f(x_i) = y_i]$$

$$\equiv \operatorname*{argmax}_{f \in \mathcal{F}} \sum_{i=1}^n \frac{\hat{u}_{\mathsf{gap}}(x_i)}{u^*_{\mathsf{max}}} \cdot \mathbb{I}[f(x_i) = y_i] \ .$$

The following lemma provides an upper bound on the estimation error of Comptron and shows that the output estimates $\hat{u}_{\mathsf{gap}}(x_i)$ are within a factor $O(\frac{u^*_{\mathsf{max}}}{k})$ of the true gaps $u^*_{\mathsf{gap}}(x_i)$.

11

Algorithm 1: Comptron: Comparison based Coordinate-Perceptron for estimating u^*_{gap}

Input: Datapoints $S = \{x_1, \ldots, x_n\}$, k-comparison oracle $\mathcal{O}_k$

Initialize: Set $T = \log_2 k - 1$

Obtain $y_i = \operatorname{argmax}_y u^*(x_i, y)$ for each i using 1-comparison.

Obtain index i_{max} using 2-comparisons such that $i_{\mathsf{max}} = \operatorname{argmax}_i u^*_{\mathsf{gap}}(x_i)$.

Set initial estimates $\hat{u}^0_{\mathsf{gap}} = [\hat{u}^0_{\mathsf{gap}}(x_1), \ldots, \hat{u}^0_{\mathsf{gap}}(x_n)] = u^*_{\mathsf{max}} := u^*_{\mathsf{gap}}(x_{i_{\mathsf{max}}})$.

(Note that exact value of u^*_{max} is not required since comparison queries are relative)

for $t = 1, \ldots, T$ **do**

    for $i = 1, \ldots, n$ **do**

        Denote by $\lambda = \frac{k}{2u^*_{\mathsf{max}}}\left(\hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^t}\right)$ and query $q_{i,t} = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where

$$\mathbf{x} = (\underbrace{x_i, \ldots, x_i}_{\frac{k}{2}\text{ times}}, \underbrace{x_{i_{\mathsf{max}}}, \ldots, x_{i_{\mathsf{max}}}}_{\lambda\text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_i, \ldots, y_i}_{\frac{k}{2}\text{ times}}, \underbrace{1 - y_{i_{\mathsf{max}}}, \ldots, 1 - y_{i_{\mathsf{max}}}}_{\lambda\text{ times}}), \quad \mathbf{y}_2 = 1 - \mathbf{y}_1.$$

        Query oracle $\mathcal{O}_k$ with $q_{i,t}$ and receive response $r_{i,t}$.

        Update $\hat{u}^t_{\mathsf{gap}}(x_i) = \hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \mathbb{I}[r_{i,t} = 0] \cdot \frac{u^*_{\mathsf{max}}}{2^t}$.

Output: Gap estimates $\hat{u}^T_{\mathsf{gap}}$

Lemma 1 (Estimation error of Algorithm 1). *Given access to datapoints $S = \{x_1, \ldots, x_n\}$ and k-comparison oracle $\mathcal{O}_k$, Comptron (Algorithm 1) uses $O(n \log k)$ queries to the oracle and produces estimates $\hat{u}_{\mathsf{gap}}$ such that*

$$\max_{i \in [n]} \left| \hat{u}_{\mathsf{gap}}(x_i) - u^*_{\mathsf{gap}}(x_i) \right| \leq \frac{2u^*_{\mathsf{max}}}{k} . \tag{15}$$

We defer the proof of the lemma to Appendix B. The proof proceed via an inductive argument where we show that the confidence interval around $u^*_{\mathsf{gap}}(x_i)$ shrinks by a factor of $\frac{1}{2}$ in each iteration for every datapoint x_i. Given the above estimation error guarantee for Comptron, the following corollary combines these with the excess risk bounds of Proposition 1 to obtain an upper bound on the excess risk of $\hat{f}_{k,n}$.

Corollary 1. *Consider the binary decision making problem with label space $\mathcal{Y} = \{0, 1\}$. Given n datapoints $\{x_1, \ldots, x_n\}$ such that each $x_i \sim \mathcal{D}_x$, the plug-in estimate $\hat{f}_{k,n}$ from equation (7), when instantiated with the output of Comptron (Algorithm 1), satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}}\left(|U(f; u^*) - \hat{U}(f; u^*)|\right) + \frac{2u^*_{\mathsf{max}}}{k} \cdot \left(\frac{1}{n}\sum_{i=1}^n \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]\right) .$$

We defer the proof of the corollary to Appendix B. Corollary 1 exhibits the advantage of using higher-order comparisons for the learning with unknown utilities problem – as the order k increases, the error of the plug-in estimate decreases additively as $O\left(\frac{1}{k}\right)$. It is worth noting here that while the higher-order comparisons allow the learner to better estimate the underlying utilities, the problem gets harder from the side of the human expert. Indeed, with higher values of k, the expert is required to compare utilities across k different possible situations which can make the elicitation a harder task.

Algorithm 2: Rob-Comptron: Robust Comptron for estimating u^*_{gap} with noisy oracle

---

Input: Datapoints $S = \{x_1, \ldots, x_n\}$, k-comparison oracle $\mathcal{O}_k$, noise level η, confidence δ

Initialize: $T = \log_2 k - 1$, $J = \frac{8}{(1-2\eta)^2} \log\left(\frac{nT}{\delta}\right)$

Obtain $y_i = \arg\max_y u^*(x_i, y)$ for each i using 1-comparison.

Obtain index i_{max} using 2-comparisons such that $i_{\mathsf{max}} = \arg\max_i u^*_{\mathsf{gap}}(x_i)$.

Set initial estimates $\hat{u}^0_{\mathsf{gap}} = [\hat{u}^0_{\mathsf{gap}}(x_1), \ldots, \hat{u}^0_{\mathsf{gap}}(x_n)] = u^*_{\mathsf{max}}$ symbolically

for $t = 1, \ldots, T$ **do**

    for $i = 1, \ldots, n$ **do**

        Denote by $\lambda = \frac{k}{2u^*_{\mathsf{max}}} \left(\hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^t} \right)$

        Set query $q_{i,t} = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where

$$\mathbf{x} = (\underbrace{x_i, \ldots, x_i}_{\frac{k}{2}\text{ times}}, \underbrace{x_{i_{\mathsf{max}}}, \ldots, x_{i_{\mathsf{max}}}}_{\lambda\text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_i, \ldots, y_i}_{\frac{k}{2}\text{ times}}, \underbrace{1 - y_{i_{\mathsf{max}}}, \ldots, 1 - y_{i_{\mathsf{max}}}}_{\lambda\text{ times}}), \quad \mathbf{y}_2 = 1 - \mathbf{y}_1.$$

        for $j = 1, \ldots, J$ **do**

            Query oracle $\mathcal{O}_k$ with $q_{i,t}$ and receive response $r_{i,j,t}$.

        Update $\hat{u}^t_{\mathsf{gap}}(x_i) = \hat{u}^{t-1}_{\mathsf{gap}}(x_i) - \mathbb{I}[\frac{1}{J}\sum_j r_{i,j,t} < \frac{1}{2}] \cdot \frac{u^*_{\mathsf{max}}}{2^t}$.

Output: Gap estimates $\hat{u}^T_{\mathsf{gap}}$

While the results in this section exhibit how the excess risk $\mathsf{err}(\hat{f}_{k,n}; \mathcal{F})$ varies as a function of k, they rely on the oracle responses being noiseless. In the next section, we consider the setup where the oracle responses can be noisy and propose a robust version of the Comptron algorithm for learning in this scenario.

4.1.2 Estimating u^*_{gap} with noisy oracle

In contrast to the deterministic noiseless oracle of the previous section, here, we consider learning with unkown utilities when the oracle $\mathcal{O}_k$ can output noisy responses to each query. Recall from equation (5), for any query q, the noisy k-comparison oracle the correct response with probability $1 - \eta_q$ and flips the response with probability η_q for some value of $\eta_q < \frac{1}{2}$. While we allow this error probability to vary across different queries, we assume that this error is bounded uniformly across all queries by some constant $\eta < \frac{1}{2}$.

Assumption 1. *For the noisy k-comparison oracle described in equation (5), we have that $\eta_q \leq \eta < \frac{1}{2}$ for all queries q.*

From an algorithmic perspective, it is well known that the perceptron algorithm itself is not noise-stable and can oscillate if there are datapoints x which have noisy labels. In order to overcome this limitation, several noise-robust perceptron variants have been proposed in the literature; see [KW07] for an extensive review.

We build on this line of work and present Rob-Comptron (Algorithm 2), a noise-robust variant of the deterministic Comptron algorithm. The main difference is the presence of an additional inner-loop with index j which repeatedly queries $q_{i,t}$ for $J = \tilde{O}\left(\frac{1}{(1-2\eta)^2}\right)$ times. In each iteration, the

update is again a coordinate-wise perceptron update which matches the prediction of the current estimate with the average of the oracle responses. Such an averaging has been previously used in the context of learning halfspaces from noisy data both in a passive [Byl94] and active [YZ17] framework.

The following lemma, whose proof we defer to Appendix B, provides an upper bound on the estimation error of the gap estimates produced by Rob-Comptron.

Lemma 2 (Estimation error of Algorithm 2). *Given access to datapoints $S = \{x_1, \ldots, x_n\}$ and noisy k-comparison oracle $\mathcal{O}_k$ satisfying Assumption 1 with parameter η, Rob-Comptron (Algorithm 2) uses $O\left(\frac{n}{(1-2\eta)^2} \cdot \log k \cdot \log \frac{n \log k}{\delta}\right)$ queries and produces estimates $\hat{u}_{\mathsf{gap}}$ such that*

$$\max_{i \in [n]} \left|\hat{u}_{\mathsf{gap}}(x_i) - u^*_{\mathsf{gap}}(x_i)\right| \leq \frac{2u^*_{\mathsf{max}}}{k} \ , \tag{16}$$

with probability at least $1 - \delta$.

In comparison to Comptron which requires $O(n \log k)$ queries to the comparison oracle, the robust variant Rob-Comptron requires a fraction $\frac{1}{(1-2\eta)^2}$ more queries to achieve a similar estimation error. Such an increase in query complexity is typical of learning with such noisy oracles in the binary classification setup [BBZ07; BL13; DKT09; YZ17].

Similar to Corollary 1 in the previous section, we can combine the above high-probability bound on the estimation error to obtain a bound on the excess risk which scales as $\frac{1}{k}$ with the order k of the comparison oracle.

Corollary 2. *Consider the binary decision making problem with label space $\mathcal{Y} = \{0, 1\}$. Given n datapoints $\{x_1, \ldots, x_n\}$ such that each $x_i \sim \mathcal{D}_x$, the plug-in estimate $\hat{f}_{k,n}$ from equation (7), when instantiated with the output of Comptron (Algorithm 1), satisfies*

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)|\right) + \frac{2u^*_{\mathsf{max}}}{k} \cdot \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]\right) \ .$$

with probability at least $1 - \delta$.

We omit the proof of this corollary since it essentially follows the same steps as that for Corollary 1. This corollary establishes that by increasing the query complexity by a factor of $O\left(1/(1-2\eta)^2\right)$, one can recover the same additive $\frac{1}{k}$ excess risk bound of the deterministic setup. Combined, Corollaries 1 and 2 establish the trade-offs in the reduction of the excess risk while eliciting more complex information about the underlying utility u^* through the k-comparison oracle.

4.2 Information-theoretic lower bounds

In the previous section, we studied the learning with unknown utility problem from an algorithmic perspective and showed that the plug-in estimator with Comptron estimates $\hat{u}$ achieve an excess risk bound which scales as $O(\frac{1}{k})$ with the order k of the comparison. In this section, we ask whether such a scaling of the error term is optimal and study this lower bound question from an information-theoretic perspective.

Recall from Theorem 2 that the excess risk decomposes into two terms: (i) a uniform convergence term for the decision class $\mathcal{F}$ with respect to utility function u^* and (ii) an estimation error

term corresponding to how well $\hat{u}_k$ approximates u^* on the sampled datapoints. When the underlying utility function u^* is known, classical results from the learning theory literature the uniform convergence complexity term is in general unavoidable [see SB14, Theorem 6.8]. With this, we take the infinite-data limit, where the learner is assumed to have access to the distribution $\mathcal{D}_x$, and study whether the excess error of $O(\frac{1}{k})$ is necessary.

Our notion of minimax risk is based on the subset of utility functions which cannot be distinguished by any learner with access to a k-comparison oracle. Formally, given any oracle $\mathcal{O}_k(\cdot\,;u^*)$, where we have made the dependence on the utility u^* explicit, we denote by $\mathcal{U}_{k,u^*}$ the subset of utility functions in the class $\mathcal{U}$ which are consistent with the responses of $\mathcal{O}_k(\cdot\,;u^*)$. With this, we define the information-theoretic minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ with respect to the function class $\mathcal{F}$ and distribution $\mathcal{D}_x$ as

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x) := \sup_{\mathcal{O}_k(\cdot\,;u^*)} \inf_{p \in \Delta_{\mathcal{F}}} \sup_{u \in \mathcal{U}_{k,u^*}} \mathbb{E}_{f \sim p}\left[\mathsf{err}(f, \mathcal{F}; u)\right] , \tag{17}$$

where the infimum is taken over all procedures which take as input the distribution $\mathcal{D}_x$ over the instances and access to a k-comparison oracle, and output a possibly randomized estimate $p \in \Delta_{\mathcal{F}}$. The above notion of minimax risk can be viewed as a three-stage game between the learner and the environment. The sequence of supremum and infimum depicts the order in which information is revealed in this game. The environment first selects a k-query oracle $\mathcal{O}(\cdot\,;u^*)$ with underlying utility u^*. The learner is then provided access to the underlying distribution $\mathcal{D}_x$, function class $\mathcal{F}$ and the oracle $\mathcal{O}(\cdot\,;u^*)$ based on which it outputs a possibly randomized decision function given by $p \in \Delta_{\mathcal{F}}$. The environment is then allowed to select the worst-case utility u such that it is consistent with the k-oracle $\mathcal{O}(\cdot\,;u^*)$ and the learner is evaluated in expectation over this chosen utility. We call this the minimax risk of learning with respect to class $\mathcal{F}$ and distribution $\mathcal{D}_x$.

Our next main result shows that there exist instances of the binary prediction problem $(\mathcal{F}, \mathcal{D}_x)$ such that the minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ is lower bounded by $\frac{1}{k}$ for any $k \geq 2$ up to some universal constants. Observe that this matches the corresponding upper bounds obtained in Corollaries 1 and 2 exhibiting that the proposed plug-in estimator in equation (7) with Comptron (Rob-Comptron for noisy oracle) utilities is indeed minimax optimal for the binary prediction setup.

Theorem 3. *There exists a universal constant $c > 0$ such that for any $k \geq 2$, there exist a binary prediction problem instance $(\mathcal{F}, \mathcal{D}_x)$ such that*

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x) \geq \frac{c}{k} .$$

A few comments on Theorem 3 are in order. First, the above result shows a family of lower bounds for our learning with unknown utilities framework – one for each value of the order k. Specifically, it shows that for every $k \geq 2$, there exists a worst-case instance such that any algorithm will incur an error of $\Omega(\frac{1}{k})$. Compare this with the upper bounds on excess risk from the previous section. In the limit of infinite data, Corollaries 1 and 2 exhibit that the excess risk $\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) = O(\frac{1}{k})$ for the plug-in estimator $\hat{f}_{k,n}$. This establishes that the plug-in estimator with Comptron and Rob-Comptron utility estimates is indeed minimax optimal.

Proof. In order to establish a lower bound on the minimax risk $\mathfrak{M}_k$, we will construct two utility functions $u_1, u_2 \in \mathcal{U}$ such that the k-comparison oracle has identical responses for both these utility functions. For the purpose of our construction, we will consider noiseless oracle; the problem only

becomes harder for the learner if the oracle responses are noisy. Given these two utility functions, we next show that their maximizers f_1 and f_2 are different for some function class $\mathcal{F}$. We then combine these two insights to obtain the final minimax bound.

For our lower bound construction, we will focus on a setup where the features are one dimensional with $\mathcal{X} = \mathbb{R}$ and the linear decision function class

$$\mathcal{F}_{\mathsf{lin}} = \{f_a \mid f_a(x) = \mathrm{sign}(ax), \; a \in [-1, 1]\} \, .$$

Recall that for any point x, we represent by $u_{\mathsf{gap}}(x) = u(x, y_x) - u(x, \bar{y}_x)$ the utility gain corresponding to the function u. Before constructing the explicit example, we present a technical lemma which highlights a limitation of a k-comparison oracle – it establishes that a k-oracle will not be able to distinguish utility functions for which the utility gaps are in the range $(1 - \frac{1}{k}, 1)$.

Lemma 3. *Consider any utility functions $u_1, u_2 \in \mathcal{U}$. Let datapoints x have utility gain $u_{\mathsf{gap}}^i(x)$ for $i = \{1, 2\}$. For any two points x_1, x_2 such that*

$$u_{\mathsf{gap}}^1(x_1) = u_{\mathsf{gap}}^2(x_1) = u_{\mathsf{gap}}(x_1) \quad and \quad \left(1 - \frac{1}{k}\right) \cdot u_{\mathsf{gap}}(x_1) \le u_{\mathsf{gap}}^i(x_2) \le u_{\mathsf{gap}}(x_1) \, ,$$

the oracle responses for any query $q = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ comprising points x_1 and x_2 are identical for $u^* = u_1$ or $u^* = u_2$.

We defer the proof of the above lemma to Appendix B. Taking this as given, we proceed with our lower bound construction.

Utility functions u_1 and u_2. Our construction considers two datapoints $x_+ = +1$ and $x_- = -1$ and two utility functions u and $\tilde{u}$ satisfying

$$u_1(x_+, 1) > u_1(x_+, 0) \quad and \quad u_1(x_-, 1) > u_1(x_-, 0) \, ,$$
$$u_2(x_+, 1) > u_2(x_+, 0) \quad and \quad u_2(x_-, 1) > u_2(x_-, 0).$$

Observe that under these utilities, any function $f_a \in \mathcal{F}_{\mathsf{lin}}$ can make a correct decision for either point x_+ or point x_- but not for both simultaneously. Given these datapoints, the two utility functions are given by

$$u_1(x_+, 1) = 1, \quad u_1(x_-, 1) = 1 - \gamma_1 \; \text{ where } \gamma_1 = \frac{1}{2(3k + 1)}$$

$$u_2(x_+, 1) = 1, \quad u_2(x_-, 1) = 1 - \gamma_2 \; \text{ where } \gamma_2 = \frac{2}{(3k + 1)},$$

and $u_i(x, 0) = 0$ for both $i = \{1, 2\}$. Observe that both γ_1, γ_2 have been set to satisfy the conditions of Lemma 3, that is,

$$\left(1 - \frac{1}{k}\right) \cdot u_{\mathsf{gap}}(x_+) \le u_{\mathsf{gap}}^i(x_-) \le u_{\mathsf{gap}}(x_+) \text{ for } i = \{1, 2\}.$$

Distribution $\mathcal{D}_x$. For any $k > 2$, consider the distribution $\mathcal{D}_x$ over the points $\{x_+, x_-\}$ such that

$$\Pr(x = x_+) = \frac{3k}{6k+1} \quad \text{and} \quad \Pr(x = x_-) = \frac{3k+1}{6k+1}.$$

By Lemma 3, we have that using the k-comparison oracle, no learner can distinguish between the utility functions u_1 and u_2 on the distribution $\mathcal{D}_x$. Further, recall that any classifier $f_a \in \mathcal{F}_{\mathsf{lin}}$ can either predict x_+ or x_- correctly. We now obtain a bound on the excess risk $\mathsf{err}(f_a, \mathcal{F}; u)$ for both these cases separately.

<u>Case 1</u>: $f_a(x_+) = 1$. In this case, the utility gap is maximized by setting the utility $u = u_1$ in the minimax risk. The corresponding excess risk is given by

$$\mathsf{err}(f_a, \mathcal{F}; u_1) = \frac{(3k+1)(1-\gamma_1)}{6k+1} - \frac{3k}{6k+1} = \frac{1}{2(6k+1)}. \tag{18}$$

<u>Case 2</u>: $f_a(x_-) = 1$. In this case, the utility gap is maximized by setting the utility $u = u_2$ and the excess risk is given by

$$\mathsf{err}(f_a, \mathcal{F}; u_2) = \frac{3k}{6k+1} - \frac{(3k+1)(1-\gamma_2)}{6k+1} = \frac{1}{(6k+1)}. \tag{19}$$

Noting that any predictor $\hat{f}$ will output a function corresponding to one of the two cases above and combining equations (18) and (19) establishes the desired claim. $\qquad\square$

While the information theoretic results of this section showed that the plug-in estimator is minimax optimal, the next section focuses on whether this estimator is able to *adapt* to easier problem instances – specifically, whether our estimation procedures Comptron and Rob-Comptron are optimal for every problem instance? We answer this in the negative and introduce a new estimator which is instance optimal. However, such an adaptivity to easier instances comes at the cost of an exponential query complexity.

5 Instance-optimal guarantees for binary prediction

In the previous section, we proposed query-efficient algorithms, Comptron and Rob-Comptron, for learning a function $\hat{f}_{k,n}$ with small excess risk using only $\tilde{O}(n \log k)$ queries to the k-comparison oracle. Further, the upper bounds in Corollaries 1 and 2 along with the lower bound of Theorem 3 establish that our proposed algorithms are indeed minimax optimal over the class of utility functions $\mathcal{U}$. Given this, it is natural to ask whether our proposed algorithms are instance wise-optimal, that is, do they achieve the best possible excess-risk bounds for *all* $u^* \in \mathcal{U}$?

To simplify our presentation, we study this question at the population level,[5] where we assume that the learner has access to the underlying distribution $\mathcal{D}_x$. This allows us to focus on the excess risk as a function of the order k of the comparison oracle and ignore the uniform convergence term. We also restrict our attention to the deterministic noiseless oracle since one can reduce the noisy oracle to the noiseless oracle by using the averaging technique presented in Section 4.1.

The following proposition shows that the plug-in estimator with Comptron utilities are *not instance-optimal*, that is, it does not adapt to the hardness of the learning with unknown utilities

---

[5]Our analysis could be extended to the finite sample setup using the bound obtained in Theorem 2.

problem instance. Specifically, it constructs a problem instance $(\mathcal{F}, \mathcal{D}_x)$ with a noiseless oracle and shows that the estimate[6] $\hat{f}_k$ from equation (7) with Comptron utility estimates has an excess risk of $\frac{1}{k}$ while there exists an estimator, which uses all k-queries and is able to achieve zero excess risk.

Recall that for any utility $u^* \in \mathcal{U}$, we denote by $\mathcal{U}_{k,u^*}$ the subset of utility functions in the class $\mathcal{U}$ which are indistinguishable from u^* under the k-comparison oracle $\mathcal{O}(\cdot\,; u^*)$.

Proposition 2 (Plug-in with Comptron estimates is not instance-optimal). *For every $k > 2$, there exists an binary prediction instance $(\mathcal{F}, \mathcal{D}_x)$ along with an oracle $\mathcal{O}_k$ such that*

  a) *The error of the plug-in estimate $\hat{f}_k$ from equation (7) with estimated utilities $\hat{u}_k$ from Comptron (Algorithm 1) is non-zero, that is,*

$$\mathsf{err}(\hat{f}_k, \mathcal{F}; u^*) = \frac{1}{k}.$$

  b) *There exists an optimal predictor $\tilde{f}$ with zero excess-risk, that is,*

$$\sup_{u \in \mathcal{U}_{k,u^*}} \mathsf{err}(\tilde{f}, \mathcal{F}; u) = 0.$$

We make a few remarks about the proposition. While the first part of the proposition shows that the excess risk $\mathsf{err}(\hat{f}_k, \mathcal{F}; u^*) = \frac{1}{k}$, the second part makes a stronger claim about the performance of $\tilde{f}$ on all utilities $u \in \mathcal{U}_{k,u^*}$. This shows that the predictor $\tilde{f}$ performs well when evaluated on an entire neighborhood around the true utility u^*. We defer the proof of the proposition to Appendix C.

Having established that our estimators from the previous section are not adaptive, we introduce a notion of *local minimax risk* and study estimators which are instance-optimal. We begin by precisely defining this notion of instance-wise minimax optimality. Recall from Section 4.2, our notion of minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ was a worst-case notion – the minimax risk was defined as a supremum over all oracles $\mathcal{O}_k(\cdot\,; u^*)$. We extend this global minimax notion to a local minimax one. In particular, for any $u^* \in \mathcal{U}$, we define the local minimax risk around u^* as

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x; u^*) := \inf_{\hat{f}} \sup_{u \in \mathcal{U}_{|u^*}} \left[\mathsf{err}(\hat{f}, \mathcal{F}; u) \right], \tag{20}$$

where the infimum is again over the set of all estimators which output a function $\hat{f} \in \mathcal{F}$ given access to distribution $\mathcal{D}_x$ and k-comparison oracle $\mathcal{O}_k$. Observe that this local notion of minimax risk concerns the performance of an algorithm $\hat{f}$ around a specific instance u^* as compared to the worst-case instance.

For any utility function $u \in \mathcal{U}$, we define its population maximizer $f_u \in \operatorname{argmax}_{f \in \mathcal{F}} U(f; u)$. With this notation, our next theorem provides a lower bound on this local minimax risk in terms of a local modulus of continuity with respect to the set $\mathcal{U}_{k,u^*}$.

Theorem 4 (Local minimax lower bound). *For any distribution $\mathcal{D}_x$ over feature space $\mathcal{X}$, utility function $u^* \in \mathcal{U}$, function class $\mathcal{F}$ and order k of the comparison oracle, the local minimax risk*

$$\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x; u^*) \geq \frac{1}{2} \cdot \sup_{u_1, u_2 \in \mathcal{U}_{k,u^*}} \left(U(f_{u_1}; u_1) - U(f_{\frac{u_1+u_2}{2}}; u_1) \right). \tag{21}$$

---

[6]Since we are working at the population level, we have dropped the subscript n from $\hat{f}_{k,n}$

Proof. Consider any two utility functions $u_1, u_2 \in \mathcal{U}_{k,u^*}$ and let $\bar{u} = \frac{u_1 + u_2}{2}$. We can then lower bound the minimax risk as

$$
\begin{aligned}
\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x; u^*) &\geq \inf_{f \in \mathcal{F}} \left(\frac{1}{2} \, \mathsf{err}(f, \mathcal{F}; u_1) + \frac{1}{2} \, \mathsf{err}(f, \mathcal{F}; u_2) \right) \\
&= \frac{1}{2} \, \mathsf{err}(f_{\bar{u}}, \mathcal{F}; u_1) + \frac{1}{2} \, \mathsf{err}(f_{\bar{u}}, \mathcal{F}; u_2) \\
&\geq \frac{1}{2} \left(U(f_{u_1}; u_1) - U(f_{\bar{u}}; u_1) \right),
\end{aligned}
$$

where the last equality follows by noting that $\mathsf{err}(f_{\bar{u}}, \mathcal{F}; u_2) \geq 0$. Since the above holds for any choice of u_1, u_2, the desired bound follows by taking a supremum over these values. $\qquad\square$

A few comments on Theorem 4 are in order. The theorem establishes that the local minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$ is lower bounded by a local modulus of continuity,

$$
\sup_{u_1, u_2 \in \mathcal{U}_{k,u^*}} \left(U(f_{u_1}; u_1) - U(f_{\frac{u_1 + u_2}{2}}; u_1) \right), \tag{22}
$$

which captures the worst-case variation in the performance of utility maximizers of utility in a neighborhood of u^*. For any two utilities $u_1, u_2 \in \mathcal{U}_{k,u^*}$, it measures the performance drop in the utility of a learner uses the maximizer $f_{\frac{u_1 + u_2}{2}}$ in place of f_{u_1} when the underlying utility is u_1.

Given this lower bound on the local minimax risk $\mathfrak{M}_k(\mathcal{F}, \mathcal{D}_x)$, it is natural to ask whether this local modulus of continuity exactly captures the instance-specific hardness of the problem. To this end, our next result answers this in the affirmative. In particular, it shows that for any u^*, the randomized minimax robust estimator $p_{\mathsf{rob}} \in \Delta_{\mathcal{F}}$, given by

$$
p_{\mathsf{rob}} \in \underset{p \in \Delta_{\mathcal{F}}}{\arg\min} \, \sup_{u \in \mathcal{U}_{k,u^*}} \, \mathbb{E}_{f \sim p}[\mathsf{err}(f, \mathcal{F}; u)], \tag{23}
$$

(nearly-)obtains the same excess-risk bound as that given by the lower bound in Theorem 4.

Theorem 5 (Upper bounds for p_{rob}). *For any distribution $\mathcal{D}_x$ over feature space $\mathcal{X}$, utility function $u^* \in \mathcal{U}$ and function class $\mathcal{F}$, the expected excess risk of the randomized estimator given by the distribution $p_{\mathsf{rob}} \in \Delta_{\mathcal{F}}$ is*

$$
\begin{aligned}
\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}}, \mathcal{F}; u^*)] &= \sup_{p_u} \left(\mathbb{E}_{u' \sim p_u} \left[U(f_{u'}; u') - U(f_{p_u}; u') \right] \right) \\
&\leq \sup_{u_1, u_2 \in \mathcal{U}_{k,u^*}} \left(U(f_{u_1}; u_1) - U(f_{u_2}; u_1) \right),
\end{aligned} \tag{24}
$$

where the distribution $p_u \in \Delta_{\mathcal{U}_{k,u^*}}$ is over the space of utility functions consistent with u^*.

We defer the proof of Theorem 5 to Appendix C. Compared with the lower bound of Theorem 4, the bound in (24) shows that the local minimax risk can indeed be upper bounded by a similar local modulus of continuity. Observe that the while the lower bound evaluates the performance loss of the maximizer $f_{\frac{u_1 + u_2}{2}}$, the upper bound is evaluated on f_{u_2}. While the minimax estimator p_{rob} in equation (23) is defined at the population level, we can naturally extend it to the finite sample regime as

$$
\hat{p}_{\mathsf{rob},n} \in \underset{p \in \Delta_{\mathcal{F}}}{\arg\min} \, \sup_{u \in \hat{\mathcal{U}}_{k,u^*}} \, \mathbb{E}_{f \sim p}[\hat{U}(f_u; u) - \hat{U}(f; u)] \tag{25}
$$

19

where the class of utilities $\hat{\mathcal{U}}_{k,u^*}$ represents the set of all n-dimensional vectors in $[0,1]^n$ which are consistent with responses to all k-queries on the set of sampled datapoints S. Using a similar analysis as in Theorem 2, one can then upper bound the excess risk of this estimator in terms of the local modulus on the dataset S and an additional uniform convergence term.

In comparison to the Comptron procedure which uses $O(n \log k)$ queries to the comparison oracle for estimating utilities, the estimator $\hat{p}_{\mathsf{rob},n}$ uses $O(n^k)$ queries to construct the set $\hat{\mathcal{U}}_{k,u^*}$. Thus, while this estimator adapts to the problem hardness, such an adaptation comes at the cost of an exponential increase in query complexity. Achieving instance-optimality by using fewer queries is an interesting question for future research.

Acknowledgments

Appendices

A   Other related work

This paper sits at the intersection of multiple fields of study: agnostic learning , learning with nuisance parameters, and utility learning from preferences . Here, we review the papers that are most relevant to our contributions.

Agnostic learning.   The framework of probably approximately correct (PAC) learning was introduced in their seminal work by Valiant [Val84]. This framework formalized the problem of learning from sampled data in a realizable setup. This was formally extended to the agnostic setup, with no assumptions on the data generating distribution, by Haussler [Hau92]. Connections of learnability with uniform convergence were first established by Vapnik [Vap92], and more recently it was established in [SB14] that for the general learning problem, such a uniform convergence is not necessary to establish learnability. Similar to the classical agnostic supervised learning, the learner does not know the distribution $\mathcal{D}_x$ but only has access to it via samples. The key difference is that the classical setup assumes that the utility function u^* is known to the learner while our framework does not.

Learning with nuisance parameters.   Closely related to our setup is the problem of learning with a nuisance component [FS19] which comprises as special case the problems of heterogeneous treatment effect estimation [Che+17], offline policy learning [AW17], and learning with missing data [Gra11] amongst others. In this setup, objective is to learn a predictor with small excess risk and this risk depends on a underlying nuisance parameter which is unknown to the learner a priori. The unknown utility u^* of our setup can be seen as a nuisance component in their framework. However, the two problems differ in the form of information available to the learner – they allow the learner to directly elicit (possibly noisy) values of utility u^*. They additionally require that utility u^* belongs to some pre-specified function class and their bounds depend on the rate at which this utility function is learnable over this class.

   Another line of work, called double/debiased machine learning in the statistics and econometrics literature [Che+18a; Che+18b; CNR18], addresses semiparametric inference [Rob88; Kos07] where the function class $\mathcal{F}$ is assumed to be a parametric family along with a non-parametric nuisance component. In addition to the differences mentioned above, this class of methods focuses on exact parameter recovery and conditions under which $\sqrt{n}$-consistent and asymptotically normal estimators can be obtained.

Utility estimation with preferences.   The seminal work of von Neumann and Morgenstern [MV53] established that any rational agent whose preferences satisfy certain axioms will have a utility function. Furthermore, the proof of this expected utility theorem showed these utilities could be elicited from the agent using preferences over randomized lotteries. As discussed in Section 1, such preferences over lotteries can be seen as a special case of the ∞-comparison oracle. There have been several recent works studying the consequences of incomplete preferences [Ok02; GK13] which show the existence of a class of utility functions which are consistent with these incomplete preferences. Our k-comparison oracles can be seen as a quantitative approach to studying such incomplete preferences; for each value of $k \geq 1$, the human expert can only compare lotteries up to a granularity

of $\frac{1}{k}$. Our work goes a step forwards and studies the consequences of such incomplete preferences for decision-making tasks.

B   Deferred proofs from Section 4

B.1   Proof of Proposition 1

The first part of the proof essentially follows the same as that for Theorem 2. The proof differs in how we upper bound Term (I) from equation (11).

$$
\hat{U}(f_{\mathsf{ERM}}; u^*) - \hat{U}(\hat{f}_{k,n}; u^*) \leq \frac{1}{n} \sum_{i=1}^{n} (\mathbb{I}[f_{\mathsf{ERM}}(x_i) = y_i] - \mathbb{I}[\hat{f}_{k,n}(x_i) = y_i])(u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i))
$$

$$
= \frac{1}{n} \sum_{i=1}^{n} (\mathbb{I}[\hat{f}_{k,n}(x_i) \neq y_i] - \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_1])(\hat{u}_{\mathsf{gap}}(x_i) - u^*_{\mathsf{gap}}(x_i)) \quad (26)
$$

$$
\overset{(i)}{\leq} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i](u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i))
$$

$$
\leq \max_i [u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)] \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i], \quad (27)
$$

where inequality (i) follows from the fact that $\hat{u}$ is a lower estimate of u^*. This establishes the desired claim. □

B.2   Proof of Lemma 1

We begin by noting that for any given datapoint x_i, the deterministic comparison oracle $\mathcal{O}_k$ when queried with $q_{i,t}$ outputs

$$
\mathcal{O}_k(q_{i,j}) = \mathbb{I}\left[\frac{k}{2} u^*_{\mathsf{gap}}(x_i) \geq \lambda u^*_{\mathsf{max}}\right],
$$

for values[7] of $\lambda \in [\frac{k}{2}]$. This effectively allows one to compare the utility gap u^*_i with u^*_{max} at a multiplicative granularity of $\frac{2}{k}$. With this observation, let us establish that for any time $t \in [T]$, for any datapoint $x_i \in S$, we have

$$
\hat{u}^t_{\mathsf{gap}}(x_i) - \frac{u^*_{\mathsf{max}}}{2^t} \leq u^*_{\mathsf{gap}}(x_i) \leq \hat{u}^t_{\mathsf{gap}}(x_i). \quad (28)
$$

The proof will proceed via an inductive argument.

<u>Base Case</u>. For initial time $t = 0$, by the boundedness of the utility functions, we have for all x_i,

$$
\hat{u}^0_{\mathsf{gap}}(x_i) - u^*_{\mathsf{max}} = 0 \leq u^*_{\mathsf{gap}}(x_i) \leq u^*_{\mathsf{max}} = \hat{u}^0_{\mathsf{gap}}(x_i).
$$

---

[7]We denote by $[d]$ the set of integers $\{1, \ldots, d\}$.

<u>Induction Step.</u> Assume that for some $t = s$, equation (28) holds for all $x_i \in S$. We will now show that it holds for $t = s + 1$. Note that by the induction hypothesis, the value of λ at time $s + 1$ can be equivalently written as

$$\lambda = \frac{k}{2u^*_{\max}} \cdot \left(\frac{\hat{u}^s_{\mathsf{gap}}(x_i) - \frac{u^*_{\max}}{2^s} + \hat{u}^s_{\mathsf{gap}}(x_i)}{2} \right) ,$$

that is, as a scaled mid-point of the confidence interval at time s. the query $q_{i,t}$ then compares the gap $u^*_{\mathsf{gap}}(x_i)$ with the mid-point of the confidence interval.

Case 1. If the response $r_{i,t} = 1$ which implies that $\hat{u}^s_{\mathsf{gap}}(x_i) \geq \frac{2\lambda}{k}$, the upper estimate remains the same and the lower estimate is (implicitly) moved to the mid-point $\frac{2\lambda}{k}$ since we know from the oracle's response that $u^*_{\mathsf{gap}}(x_i)$ is greater than the mid-point. Thus, after each update, the confidence interval shrinks by a factor of $\frac{1}{2}$ and reduces to $\frac{1}{2^{s+1}}$ at the end of time $t = s + 1$.

Case 2. On the other hand if $r_{i,t} = 0$, the estimate $\hat{u}^{s+1}_{\mathsf{gap}}(x_i)$ is updated to be the midpoint $\frac{u^*_{\max}}{2^{s+1}}$ while the lower estimate remains the same because of the oracle's response.

Combining both the cases above, we see that at time $t = s + 1$, the confidence interval for $u^*_{\mathsf{gap}}(x_i)$ is exactly $\frac{1}{2^{s+1}}$ for both the cases. Thus, we must have that

$$\hat{u}^{s+1}_{\mathsf{gap}}(x_i) - \frac{u^*_{\max}}{2^{s+1}} \leq u^*_{\mathsf{gap}}(x_i) \leq \hat{u}^{s+1}_{\mathsf{gap}}(x_i).$$

This establishes the first part of the claim. The bound on the query complexity follows from the fact that for each datapoint x_i, we use $\log_2 k - 1$ queries to the oracle in the procedure. This establishes the desired claim. $\qquad\square$

B.3   Proof of Corollary 1

The excess risk of the plug-in estimator can be upper-bounded from Proposition 1 as

$$\mathsf{err}(\hat{f}_{k,n}, \mathcal{F}; u^*) \leq 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)| \right) + \max_i [u^*_{\mathsf{gap}}(x_i) - \hat{u}_{\mathsf{gap}}(x_i)] \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i]$$

$$\overset{\text{Lemma 1}}{\leq} 2 \cdot \sup_{f \in \mathcal{F}} \left(|U(f; u^*) - \hat{U}(f; u^*)| \right) + \frac{2u^*_{\max}}{k} \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[f_{\mathsf{ERM}}(x_i) \neq y_i] , \qquad (29)$$

where the last inequality follows by noting that Comptron produces an estimate $\hat{u}_{\mathsf{gap}}$ of the utility gap $u^*_{\mathsf{gap}}(x_i)$ with an additive error of $\frac{2u^*_{\max}}{k}$. $\qquad\square$

B.4   Proof of Lemma 2

To establish the above claim, we show that the updates to the gap estimates $\hat{u}^t_{\mathsf{gap}}$ performed by Rob-Comptron mirror those performed by the deterministic Comptron with high probability. For any datapoint x_i and any time $t \in [T]$, denote by $r^*_{i,t} = \mathbb{E}[r_{i,j,t}]$ the expected value of the response for query $q_{i,j,t}$. By Assumption 1, we have that $\mathbb{I}[r^*_{i,t} < \frac{1}{2}]$ provides the true label for the query

$q_{i,j,t}$. By an application of the Hoeffding's inequality, we have,

$$\Pr\left(\mathbb{I}\left[\frac{1}{J}\sum_j r_{i,j,t} < \frac{1}{2}\right] \neq \mathbb{I}\left[r^*_{i,t} < \frac{1}{2}\right]\right) \leq \exp\left(\frac{-J(1-2\eta)^2}{4}\right).$$

Taking a union bound over all datapoints x_i and time $t \in [T]$, and substituting the value of $J = \frac{8}{(1-2\eta)^2}\log(\frac{nT}{\delta})$, we have,

$$\Pr\left(\exists i, t \text{ s.t. } \mathbb{I}\left[\frac{1}{J}\sum_j r_{i,j,t} < \frac{1}{2}\right] \neq \mathbb{I}\left[r^*_{i,t} < \frac{1}{2}\right]\right) \leq \delta. \tag{30}$$

From the above equation, we have that with probability at least $1 - \delta$, every update performed by Rob-Comptron uses the correct label. Combining the above with the proof of Lemma 1 establishes the required claim. □

B.5 Proof of Lemma 3

Observe that from the conditions of the lemma statement, we have

$$\left(1 - \frac{1}{k}\right) \cdot u_{\mathsf{gap}}(x_1) \leq u^1_{\mathsf{gap}}(x_2), u^2_{\mathsf{gap}}(x_2) \leq u_{\mathsf{gap}}(x_1).$$

Assume without loss of generality that $u^1_{\mathsf{gap}}(x_2) > u^2_{\mathsf{gap}}(x_2)$. Observe that any k-query comprising only points x_1 and x_2 must have the form

$$\mathbf{x} = (\underbrace{x_1, \ldots, x_1}_{j_1 \text{ times}}, \underbrace{x_2, \ldots, x_2}_{j_2 \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_1, \ldots, y_1}_{j_1 \text{ times}}, \underbrace{\bar{y}_2, \ldots, \bar{y}_2}_{j_2 \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{\bar{y}_1, \ldots, \bar{y}_1}_{j_1 \text{ times}}, \underbrace{y_2, \ldots, y_2}_{j_2 \text{ times}})$$

with $j_1 + j_2 = k$. For any query q to be different under the oracles $\mathcal{O}_k(\cdot\,; u_1)$ and $\mathcal{O}_k(\cdot\,; u_2)$, we should have

$$\mathbb{I}[j_1 u^1_{\mathsf{gap}}(x_1) > j_2 u^1_{\mathsf{gap}}(x_2)] = 1 \quad \text{and} \quad \mathbb{I}[j_1 u^2_{\mathsf{gap}}(x_1) > j_2 u^2_{\mathsf{gap}}(x_2)] = 0$$

since $u^1_{\mathsf{gap}}(x_2) > u^2_{\mathsf{gap}}(x_2)$. In order for the above equation to be satisfied, we requires that the ratio $\frac{j_1}{j_2} \geq 1 - \frac{1}{k}$. However, under the constraints $j_1 + j_2 = k$, this is not possible. Hence, it is not possible to distinguish between the utilities u_1 and u_2 using a k comparison oracle. □

C Deferred proofs from Section 5

C.1 Proof of Proposition 2

Our example construction will focus on the real-valued feature space $\mathcal{X} = \mathbb{R}$, binary decision space $\mathcal{Y} = \{0, 1\}$, and the class of linear decision functions

$$\mathcal{F}_{\mathsf{lin}} = \{f_a \mid f_a(x) = \mathrm{sign}(ax), \ a \in [-1, 1]\}.$$

Distribution $\mathcal{D}_x$. Our example will focus on three points $x_1 = 1, x_2 = 2, x_3 = -1$ with their population probabilities given by

$$\Pr(x = x_1) = p, \quad Pr(x = x_2) = p, \quad \text{and} \quad \Pr(x = x_3) = 1 - 2p \,,$$

for some value $p > 0$ which we define later. Note that our final choice of p will depend on the order k of the comparison oracle.

Utility function u^*. Given the above three points, we set the utility $u^*(x_i, 0) = 0$ for all data-points x_i. The utilities for label $y = 1$ are given by

$$u^*(x_1, 1) = 1, \quad u^*(x_2, 1) = \frac{4}{k}, \quad \text{and} \quad u^*(x_3, 1) = \frac{2}{k^2}.$$

With these utilities, observe that the true label $y_i = 1$ for all the datapoints. Further, any predictor $f \in \mathcal{F}_{\mathsf{lin}}$ can either correctly predict the points $\{x_1, x_2\}$ or the point x_3 but not all three simultaneously.

Performance of predictors. For this setup described above, we now proceed to describe the optimal function f^*, the plug-in estimate $\hat{f}_k$ and an alternate predictor $\tilde{f}$ which outperforms the plug-in estimate. Observe that any estimator will pick either f_{-1} or f_{+1} depending on the value of p.

Optimal Classifier. The difference in the expected utility between the classifiers f_{+1} and f_{-1} is given by

$$U(f_{+1}; u^*) - U(f_{-1}; u^*) = p \cdot \left(1 + \frac{4}{k}\right) - (1 - 2p) \cdot \left(\frac{2}{k^2}\right)$$
$$= \frac{p}{k^2} \cdot (k + 2)^2 - \frac{2}{k^2}$$
$$= \frac{1}{k^2} \left(p(k + 2)^2 - 2\right).$$

Given the calculation above, the optimal classifier f^* is given by

$$f^* = \begin{cases} f_{+1} & \text{for } p \geq \frac{2}{(k+2)^2} \\ f_{-1} & \text{otherwise} \end{cases} . \tag{31}$$

Plug-in estimate $\hat{f}_k$. We now study the prediction $\hat{f}_k$ obtained by using the prediction $\hat{u}$ from Comptron (Algorithm 1). Recall that since Comptron produces upper estimates for u^*_{gap} (which is equivalent to u^* since $u^*(x, 0) = 0$) within an error of $\frac{2}{k}$, the output estimates will be

$$\hat{u}(x_1, 1) = 1, \quad \hat{u}(x_2, 1) = \frac{4}{k}, \quad \text{and} \quad \hat{u}(x_3, 1) = \frac{2}{k}.$$

Observe that while Comptron is able to correctly learn the utilities for x_1 and x_2, it overestiamtes the utility for the point x_3. Let us look at the difference of estimated utilities

$$U(f_{+1}; \hat{u}) - U(f_{-1}; \hat{u}) = p \cdot \left(1 + \frac{4}{k}\right) - (1 - 2p) \cdot \left(\frac{2}{k}\right)$$
$$= \frac{p}{k} \cdot (k + 8) - \frac{2}{k}$$
$$= \frac{1}{k} \cdot (p(k + 8) - 2).$$

Given the above calculations, we see that the function $\hat{f}_k$ is given by

$$\hat{f}_k = \begin{cases} f_{+1} & \text{for } p \geq \frac{2}{k+8} \\ f_{-1} & \text{otherwise} \end{cases}. \tag{32}$$

Alternate estimator $\tilde{f}$. While Comptron compares the utilities of both x_2 and x_3 with respect to x_1 (equivalently $x_{i_{\max}}$), consider the alternate procedure which differs in the estimation of utility gap $u^*_{\mathsf{gap}}(x_3)$. Instead of using the proposed queries $q_{3,t}$ of Comptron, we modify those as $\tilde{q}_{3,t} = (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ where

$$\mathbf{x} = (\underbrace{x_3, \ldots, x_3}_{\frac{k}{2} \text{ times}}, \underbrace{x_2, \ldots, x_2}_{\lambda \text{ times}}), \quad \mathbf{y}_1 = (\underbrace{y_3, \ldots, y_3}_{\frac{k}{2} \text{ times}}, \underbrace{1 - y_2, \ldots, 1 - y_2}_{\lambda \text{ times}}), \quad \mathbf{y}_2 = 1 - \mathbf{y}_1.$$

Following the same proof as of Lemma 1, we can show that one can obtain an upper estimate $\tilde{u}(x_3, 1) = \frac{8}{k^2}$. This follows from the fact that we can deduce that $u^*(x_3) \in [0, \frac{2u^*(x_2)}{k}]$ from the above queries and combining this with the fact that $u^*(x_2) \leq \frac{4}{k}$. Evaluating the difference between the utilities with respect to $\tilde{u}$, we get

$$U(f_{+1}; \tilde{u}) - U(f_{-1}; \tilde{u}) = p \cdot \left(1 + \frac{4}{k}\right) - (1 - 2p) \cdot \left(\frac{8}{k^2}\right)$$

$$= \frac{p}{k^2} \cdot ((k+2)^2 + 12) - \frac{8}{k^2}$$

$$= \frac{1}{k^2} \left(p((k+2)^2 + 12) - 8\right).$$

Using such estimates $\tilde{u}$ with the plug-in estimator in equation (7), we have that the function

$$\tilde{f} = \begin{cases} f_{+1} & \text{for } p \geq \frac{8}{(k+2)^2+12} \\ f_{-1} & \text{otherwise} \end{cases}. \tag{33}$$

Thus, the three estimators f^*, $\hat{f}_k$ and $\tilde{f}$ differ in the threshold for p for switching between the functions f_{+1} and f_{-1}. Setting a value of $p = \frac{1}{k+8}$, we see that for $k > 10$

$$\frac{2}{(k+2)^2} < \frac{8}{(k+2)^2 + 12} < \underbrace{\frac{1}{k+8}}_{p} < \frac{2}{k+8}.$$

Thus, for this setting of p, while the predictor $f^* = \tilde{f} = f_{+1}$, the estimator $\hat{f}_k = f_{-1}$ and hence it incurs an excess risk $\mathsf{err}(\hat{f}_k, \mathcal{F}; u^*) = \frac{1}{k}$. This establishes the first part of the claim.

For the second part, observe that the estimator $\tilde{f}$ outputs f_+ for the particular setting of p for all $\tilde{u}_3 \in [0, \frac{8}{k^2}]$. This set precisely captures the set of all utilities which are consistent with the k oracle $\mathcal{O}(\cdot\,; k)$. Since the optimal decision function $f^* = f_+$, this establishes the second part of the claim. $\qquad\square$

C.2 Proof of Theorem 5

Let us represent by $\Delta_{\mathcal{F}}$ the space of probability distributions over the function $\mathcal{F}$. The error of the estimator p_{rob} can then be upper bounded as

$$
\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}}, \mathcal{F}; u^*)] = \inf_{p \in \Delta_{\mathcal{F}}} \sup_{u' \in \mathcal{U}_{|u^*}} \mathbb{E}_f[\mathsf{err}(f, \mathcal{F}; u')]
$$

$$
= \inf_{p \in \Delta_{\mathcal{F}}} \sup_{u' \in \mathcal{U}_{|u^*}} \sup_{f' \in \mathcal{F}} \mathbb{E}_f[U(f'; u') - U(f; u')]
$$

$$
\stackrel{(i)}{=} \sup_{p \in \Delta_{\mathcal{F} \times \mathcal{U}_{|u^*}}} \inf_{f \in \mathcal{F}} \mathbb{E}_{(f', u')} \left[U(f'; u') - U(f; u') \right],
$$

where the equality (i) follows from an application of Sion's minimax theorem and the space $\Delta_{\mathcal{F} \times \mathcal{U}_{|u^*}}$ denotes the space of all distributions over the joint space $\mathcal{F} \times \mathcal{U}_{|u^*}$. Let us decompose the distribution $p = p_u \cdot p_{f|u}$ where p_u represents the marginal distribution over the space $\mathcal{U}_{|u^*}$ and $p_{f|u}$ denotes the conditional distirbution of sampling a function $f \in \mathcal{F}$ given utility function u. Denote by

$$
f_u := \operatorname*{argmax}_{f \in \mathcal{F}} U(f; u) \quad \text{and} \quad f_p := \operatorname*{argmax}_{f \in \mathcal{F}} \mathbb{E}_{u \sim p}[U(f; u)]
$$

as the maximizers for the corresponding (expected) utility functions. Then, the excess risk

$$
\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}}, \mathcal{F}; u^*)] = \sup_{p \in \Delta_{\mathcal{F} \times \mathcal{U}_{|u^*}}} \inf_{f \in \mathcal{F}} \mathbb{E}_{(f', u')} \left[U(f'; u') - U(f; u') \right]
$$

$$
= \sup_{p_u} \sup_{p_{f|u}} \inf_{f \in \mathcal{F}} \left(\mathbb{E}_{u' \sim p_u} \mathbb{E}_{f' \sim p_{f|u'}} \left[U(f'; u') \right] - \mathbb{E}_{u' \sim p_u} \left[U(f; u') \right] \right)
$$

$$
\stackrel{(i)}{=} \sup_{p_u} \sup_{p_{f|u}} \left(\mathbb{E}_{u' \sim p_u} \mathbb{E}_{f' \sim p_{f|u'}} \left[U(f'; u') \right] - \mathbb{E}_{u' \sim p_u} \left[U(f_{p_u}; u') \right] \right)
$$

$$
\stackrel{(ii)}{=} \sup_{p_u} \left(\mathbb{E}_{u' \sim p_u} \left[U(f_{u'}; u') - U(f_{p_u}; u') \right] \right),
$$

where the inequality (i) follows from the fact that f_{p_u} maximizes the expected utility with respect to p_u and (ii) follows by noting that the maximizing distribution $p_{f|u'} = \mathbb{I}[f = f_{u'}]$. Noting that f_{p_u} is the maximizer corresponding to the distribution p_u, we have,

$$
\mathbb{E}[\mathsf{err}(p_{\mathsf{rob}}, \mathcal{F}; u^*)] = \sup_{p_u} \left(\mathbb{E}_{u' \sim p_u} \left[U(f_{u'}; u') - \mathbb{E}_{\tilde{u} \sim p_u}[U(f_{\tilde{u}}; u')] + \mathbb{E}_{\tilde{u} \sim p_u}[U(f_{\tilde{u}}; u')] - U(f_{p_u}; u') \right] \right)
$$

$$
\leq \sup_{p_u} \left(\mathbb{E}_{u' \sim p_u} \left[U(f_{u'}; u') - \mathbb{E}_{\tilde{u} \sim p_u}[U(f_{\tilde{u}}; u')] \right] \right)
$$

$$
\stackrel{(i)}{\leq} \sup_{u_1, u_2 \in \mathcal{U}_{|u^*}} \left(U(f_{u_1}; u_1) - U(f_{u_2}; u_1) \right),
$$

where the inequality (i) follows by upper bounding the expected deviation with a worst-case deviation. This establishes the required claim. □

References

[Afr67]   Sydney N Afriat. "The construction of utility functions from expenditure data". In: *International economic review* 8.1 (1967) (Cited on page 1).

[Amo+16]   Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565* (2016) (Cited on page 2).

[AN04]   Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004 (Cited on page 1).

[Arg+09]   Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. "A survey of robot learning from demonstration". In: *Robotics and autonomous systems* 57.5 (2009) (Cited on page 1).

[AW17]   Susan Athey and Stefan Wager. "Efficient policy learning". In: *arXiv preprint arXiv:1702.02896* (2017) (Cited on page 21).

[Bal+14]   Maria-Florina Balcan, Amit Daniely, Ruta Mehta, Ruth Urner, and Vijay V Vazirani. "Learning economic parameters from revealed preferences". In: *International Conference on Web and Internet Economics*. 2014 (Cited on page 2).

[BBZ07]   Maria-Florina Balcan, Andrei Broder, and Tong Zhang. "Margin based active learning". In: *International Conference on Computational Learning Theory*. Springer. 2007, pp. 35–50 (Cited on page 14).

[Bis06]   Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006 (Cited on page 1).

[BL13]   Maria-Florina Balcan and Phil Long. "Active and passive learning of linear separators under log-concave distributions". In: *Conference on Learning Theory*. 2013, pp. 288–316 (Cited on page 14).

[BM02]   Peter L Bartlett and Shahar Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482 (Cited on page 9).

[BV06]   Eyal Beigman and Rakesh Vohra. "Learning from revealed preference". In: *Proceedings of the 7th ACM Conference on Electronic Commerce*. 2006 (Cited on pages 1, 2).

[Byl94]   Tom Bylander. "Learning linear threshold functions in the presence of classification noise". In: *Proceedings of the seventh annual conference on Computational learning theory*. 1994 (Cited on page 14).

[Che+17]   Victor Chernozhukov, Matt Goldman, Vira Semenova, and Matt Taddy. "Orthogonal machine learning for demand estimation: High dimensional causal inference in dynamic panels". In: *arXiv preprint arXiv:1712.09988* (2017) (Cited on page 21).

[Che+18a]   Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. *Double/debiased machine learning for treatment and structural parameters*. 2018 (Cited on page 21).

[Che+18b]   Victor Chernozhukov, Denis Nekipelov, Vira Semenova, and Vasilis Syrgkanis. "Plug-in regularized estimation of high-dimensional parameters in nonlinear semiparametric models". In: *arXiv preprint arXiv:1806.04823* (2018) (Cited on page 21).

[CNR18]   Victor Chernozhukov, Whitney K Newey, and James Robins. *Double/de-biased machine learning using regularized Riesz representers*. Tech. rep. cemmap working paper, 2018 (Cited on page 21).

[DHS12]   Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012 (Cited on page 1).

[DKT09]   Sanjoy Dasgupta, Adam Tauman Kalai, and Adam Tauman. "Analysis of Perceptron-Based Active Learning." In: *Journal of Machine Learning Research* 10.2 (2009) (Cited on page 14).

[FH10]   Johannes Fürnkranz and Eyke Hüllermeier. "Preference learning and ranking by pairwise comparison". In: *Preference learning*. 2010 (Cited on page 3).

[FS19]   Dylan J Foster and Vasilis Syrgkanis. "Orthogonal statistical learning". In: *arXiv preprint arXiv:1901.09036* (2019) (Cited on page 21).

[GK13]   Tsogbadral Galaabaatar and Edi Karni. "Subjective expected utility with incomplete preferences". In: *Econometrica* 81.1 (2013) (Cited on page 21).

[Gra11]   Bryan S Graham. "Efficiency bounds for missing data models with semiparametric restrictions". In: *Econometrica* 79.2 (2011) (Cited on page 21).

[Hau92]   David Haussler. "Decision theoretic generalizations of the PAC model for neural net and other learning applications". In: *Information and computation* 100.1 (1992) (Cited on pages 3, 5, 21).

[Kos07]   Michael R Kosorok. *Introduction to empirical processes and semiparametric inference*. 2007 (Cited on page 21).

[KW07]   Roni Khardon and Gabriel Wachman. "Noise tolerant variants of the perceptron algorithm". In: *Journal of Machine Learning Research* 8 (2007) (Cited on page 13).

[Mil56]   George A Miller. "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2 (1956) (Cited on page 3).

[MV53]   Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953 (Cited on pages 4, 21).

[NR00]   Andrew Y Ng and Stuart J Russell. "Algorithms for inverse reinforcement learning." In: *International Conference on Machine Learning*. Vol. 1. 2000, p. 2 (Cited on page 2).

[Ok02]   Efe A Ok. "Utility representation of an incomplete preference relation". In: *Journal of Economic Theory* 104.2 (2002) (Cited on page 21).

[Rob88]   Peter M Robinson. "Root-N-consistent semiparametric regression". In: *Econometrica: Journal of the Econometric Society* (1988) (Cited on page 21).

[Ros58]   Frank Rosenblatt. "The Perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958) (Cited on page 11).

[SB14]   Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014 (Cited on pages 15, 21).

[SBC05]   Neil Stewart, Gordon DA Brown, and Nick Chater. "Absolute identification by relative judgment." In: *Psychological review* 112.4 (2005) (Cited on page 3).

[Sha+10]   Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. "Learnability, stability and uniform convergence". In: *The Journal of Machine Learning Research* 11 (2010) (Cited on page 8).

[Thu27]   Louis L Thurstone. "The method of paired comparisons for social values." In: *The Journal of Abnormal and Social Psychology* 21.4 (1927) (Cited on page 3).

[Val84]   Leslie G Valiant. "A theory of the learnable". In: *Communications of the ACM* 27.11 (1984) (Cited on page 21).

[Vap92]   Vladimir Vapnik. "Principles of risk minimization for learning theory". In: *Advances in neural information processing systems*. 1992, pp. 831–838 (Cited on page 21).

[YZ17]   Songbai Yan and Chicheng Zhang. "Revisiting perceptron: Efficient and label-optimal learning of halfspaces". In: *Advances in Neural Information Processing Systems*. 2017 (Cited on page 14).

[Zie+08]   Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. "Maximum entropy inverse reinforcement learning." In: *Aaai*. Vol. 8. 2008 (Cited on page 2).